



Acta de Correcciones al Documento de Trabajo de Grado

Santiago de Cali, 22 de julio de 2024

Autor: FABIAN ANDRÉS CAICEDO CUELLAR

Título del Trabajo de Grado: “Framework para la integración de herramientas de Inteligencia Artificial en los productos de software para el área de Seguridad y Salud en el Trabajo desarrollados por la Corporación Talentum”

Director: CEBALLOS ARGOTE OSCAR ORLANDO Ph. D.

Como indica el artículo 2.13 de las Directrices para Trabajo de Grado de Maestría, he verificado que el estudiante indicado arriba ha implementado todas las correcciones que los Jurados del Proyecto de Trabajo de Grado definieron que se efectuaran, como consta en el Acta de Evaluación correspondiente.

Firma del Director del Trabajo de Grado

Santiago de Cali, 17 de mayo de 2024.

Señores

Pontificia Universidad Javeriana Cali

Ph.D. Luisa Rincón

Directora Maestría en Ingeniería de Software

Cali.

Cordial Saludo.

Me permito presentar a su consideración el proyecto de grado titulado “Framework para la integración de herramientas de Inteligencia Artificial en los productos de software para el área de Seguridad y Salud en el Trabajo desarrollados por la Corporación Talentum” con el fin de cumplir con los requisitos exigidos por la Universidad y para que sea sometido a revisión del jurado y cumpla su aprobación, para conseguir posteriormente el título de Magister en Ingeniería de Software.

Atentamente,



Fabian Andres Caicedo Cuellar, Esp.

Código: 8984641

Santiago de Cali, 17 de mayo de 2024.

Señores

Pontificia Universidad Javeriana Cali

Ph.D. Luisa Rincón

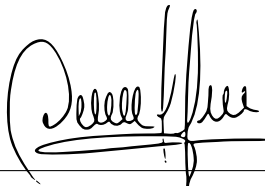
Directora Maestría en Ingeniería de Software

Cali.

Cordial Saludo.

Por medio de la presente hago constar que en mi calidad de director de trabajo de grado he revisado el proyecto titulado “Framework para la integración de herramientas de Inteligencia Artificial en los productos de software para el área de Seguridad y Salud en el Trabajo desarrollados por la Corporación Talentum” realizado por el estudiante de Magister en Ingeniería de Software Fabian Andres Caicedo Cuellar, Esp. (cod: 8984641), el cual se encuentra terminado y considero que cumple con los requisitos para ser sustentado.

Atentamente,

A handwritten signature in black ink, appearing to read 'Oscar Orlando Ceballos Argote', written over a horizontal line.

Ceballos Argote Oscar Orlando, Ph.D.



Oscar Orlando Ceballos Argote

Ph.D. and Master in Computer Science - Systems Engineer

(+57) 311 3311149

oscar.o.ceballos@gmail.com

linkedin.com/in/oscar-o-ceballos/

github.com/oscarceballos

Systems Engineer, Master and Ph.D. in Computer Science with 10+ year of experience teaching at important Colombia's universities. Currently, I work as a Data Engineer at Deloitte Canada. I understand database design for SQL and NoSQL models. I have a background in ELT/ETL, Data Warehousing, and Data Mining process. I have experience with some Big Data open tools for batch and streaming data processing. Most important, I have outstanding communication, leadership, critical thinking, and problem-solving capabilities.

TEACHING EXPERIENCE

Pontificia Universidad Javeriana Cali

Part-Time Teacher

Remote

Periods: 2023-B and 2022-B

- Program: Computer and Systems Engineering
Signature: Big Data Processing

Universidad de Nariño

Part-Time Teacher

Pasto, Nariño, Colombia

Periods: from 2006-B to 2008-B, 2020-B, and 2021-B

- Program: Master's Degree in Computer and Systems Engineering
Signature: Advanced Database System
- Program: Systems Engineering
Signatures: Graphic Software, Hardware I, Hardware III, and Software Optimization

Universidad del Valle

Full-Time Teacher

Cali, Valle del Cauca, Colombia

Periods: from 2008-B to 2017-A

- Program: Master's Degree in Computer and Systems Engineering
Signature: Database Fundamentals
- Program: Computer and Systems Engineering
Signatures: Interactive Programming, Database I, Knowledge Discovery in Databases (KDD), Discrete Mathematics
- Program: Software Development Technology
Signatures: Discrete Mathematics, Web Applications and Wireless Networks

Fundación Universitaria San Martín

Part-Time Teacher

Cali, Valle del Cauca, Colombia

Periods: from 2010-B to 2012-A

- Program: System Engineering
Signatures: Web Applications, Programming Language, Multilayer Web Applications, and Programming Language Laboratory

RESEARCH PUBLICATIONS

SPARQL2Flink: SPARQL query evaluation on Apache Flink

November 2021

MDPI

SMITag: social network for semantic annotation of medical images

February 2013

IEE Xplorer

Semantic annotation of medical images

June 2010

Acta Biológica Colombiana

WORK EXPERIENCE

Data Engineer S.Sr. at Globant

Remote

Deloitte Canada Client

September 2021 - Currently

- Build data pipeline on Azure Data Factory to ingest data from files (.csv, .txt, .zip, .pgp) and database (Hanna, DB2, SQL Server) to Azure SQL Server
- Build Python functions with PyCharm to insert into Azure Function
- Build tables, views, stored procedures, and data shapping end point
- Build data dictionary and data mapping definition
- Create and runing unit testing
- Environments: DEV, QA, and UAT
- Project management with DevOps and SCRUM methodology
- **Skills:** Azure Logic Apps, Azure Data Factory, Function App, Key Volts, Azure Synapse Analytics, Storage Accounts, SQL, Git, DevOps, Agile Methodology, Postman, Microsfot SQL Server, Python, PyCharm

Java Software Developer at AreaMovil

Pasto, Nariño, Colombia

Projects: Develop software base on microservice architecture

May 2021 – August 2021

- Develop microservices with Java & Sprint Boot
- Build database with Postgres
- Create unit testing using JUnit, Mockito
- Create integration testing APIs with Postman and JMeter
- Project management with Git, Jira, Maven, and SCRUM methodology
- **Skills:** SQL, PostgreSQL, Mockito, JUnit, Spring Boot, Git, Agile Metodology, Java Microservices, Postman

Java Software Developer at XOFTIX SAS

Pasto, Nariño, Colombia

Projects: SPA, Data Analytics Module, Biometric System, and XOFTIX Search

June 2018 – December 2020

- Develop microservices with Java & Sprint Boot
- Build database with MySQL, Postgres, and Elasticsearch
- Create unit testing using JUnit, Mockito
- Create integration testing APIs with Postman and JMeter
- Project management with Git, Jira, Maven, and SCRUM methodology
- **Skills:** SQL, PostgreSQL, MySQL, Elasticsearch, JUnit, Spring Boot, Git, Agile Metodology, Java Microservices, Postman

Academic Program Coordinator at Universidad del Valle

Tuluá, Valle del Cauca, Colombia

Academic Program: Tecnología en Sistemas de la Información

August 2012 – December 2013

- Schedule academic calendar for each period
- Part-time teacher hiring process
- Manage student academic problems (200+ studens)
- Participate in activities to update qualified registration and accreditation
- **Skills:** Outstanding communication, leadership, critical thinking, and problem-solving capabilities

PHP Developer at SITI

Pasto, Nariño, Colombia

Projects: Note Management Web Information System

August 2003 – June 2004

- Develop back-end with PHP language
- Develop front-end with HTML, JavaScript, and CSS
- Build database on MySQL
- **Skills:** SQL, PostgreSQL, MySQL, PHP, Symfony, Agile Metodology

COURSES

Data Warehouse Fundamentals for Beginners 5 Total Hours - June, 16 2022	Udemy
SQL for Beginners: The Easiest Way to Learn SQL - Step by Step 8.5 Total Hours - May, 20 2022	Udemy
Beginner's Guide to Data & Data Analytics 1.5 Total Hours - May, 5 2022	Udemy
Introduction to Big Data with Apache Spark July, 10 2015	edX Berkely

GENERAL SKILLS

Languages: Spanish (Native), English (Intermediate)
Programming: Python, Java
Relational Database: SQL Server, Postgres, MySQL
NoSQL Database: MongoDB, Elasticsearch
Semantic Web: Linked Data, Linked Stream Data, SPARQL, RDF, RDF stream, Jena
Data Analytics: Data Integration (Pentaho, Talent), Data Warehouse, Data Mining
Big Data Tools: Apache Hadoop, Apache Spark, Apache Flink, Apache Storm
Cloud Platforms: Microsoft Azure: Data Factory, Logic App, Data Lake, Blob Storage, Functions, Synapse

EDUCATION

Ph.D. <i>Computer Science</i> Universidad del Valle	August 2014 – March 2022 Cali, Valle del Cauca, Colombia
Master of Science <i>Computer and System Engineering</i> Universidad del Valle	August 2008 – June 2010 Cali, Valle del Cauca, Colombia
Bachelor of Science <i>Systems Engineer</i> Univesidad de Nariño	August 1997 – December 2003 Pasto, Nariño, Colombia

Ficha Resumen

Trabajo de Grado de Maestría

Título: Framework para la integración de herramientas de Inteligencia Artificial en los productos de software para el área de Seguridad y Salud en el Trabajo desarrollados por la Corporación Talentum

1. **Énfasis:** Ingeniería de Software
2. **Tipo de proyecto:** Aplicado
3. **Área de trabajo:** Ingeniería y tecnología
4. **Estudiante:** Fabián Andrés Caicedo Cuellar
5. **Correo electrónico:** fabiancaicedo@javerianacali.edu.co
6. **Dirección y teléfono:** Carrera 23C #10-02 / 3176367317
7. **Director:** Oscar Orlando Ceballos Argote, Ph.D.
8. **Vinculación del director:** Hora Cátedra
9. **Correo electrónico del director:** oscar.cebillos@javerianacali.edu.co
10. **Palabras clave:** Inteligencia Artificial (IA), Seguridad y Salud en el Trabajo (SST), Ingeniería de Software (IS), Marco de Trabajo, Prompt.
11. **ODS que aplica el proyecto (Agenda 2030):** Industria, Innovación e Infraestructura
12. **Fecha de inicio:** 1 de Enero de 2024

13. **Resumen:** La presente propuesta de investigación propone un *framework* para la incorporación efectiva de componentes basados en Inteligencia Artificial (IA) en arquitecturas de software ya existentes, poniendo especial atención en el ámbito de la Seguridad y Salud en el Trabajo (SST). Mediante un análisis y revisión de la literatura existente, se examina la problemática asociada a la integración de la IA en sistemas preexistentes, identificando así los retos técnicos, arquitectónicos y contextuales que implica la implementación de esta tecnología.

En particular, se utiliza como caso de estudio la Corporación Talentum, una entidad prominente en la implementación de proyectos gubernamentales en Colombia. Esta corporación afronta desafíos significativos para integrar productos de IA en sus soluciones, resaltando la urgente necesidad de establecer un conjunto estandarizado de buenas prácticas y directrices.

Como resultados esperados se construirá un documento que defina el *framework* a través de buenas prácticas para la integración de la IA, acompañado de un conjunto de diagramas UML y del modelo C4 para ilustrar los componentes arquitectónicos sugeridos. Se espera, además, desarrollar un prototipo funcional que demuestre de manera tangible la utilidad y aplicabilidad del *framework* en contextos reales.

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería
Ingeniería de Sistemas y Computación
Proyecto de Grado

Framework para la integración de herramientas de Inteligencia Artificial en los productos de software para el área de Seguridad y Salud en el Trabajo desarrollados por la Corporación Talentum

Fabian Andres Caicedo Cuellar, Esp.

Director: Ceballos Argote Oscar Orlando, Ph.D.

17 de mayo de 2024



Pontificia Universidad
JAVERIANA
Cali

0.1. Resumen

A pesar de los evidentes avances en el área de la Inteligencia Artificial (IA), su integración efectiva en soluciones de software orientadas a la Seguridad y Salud en el Trabajo presenta desafíos que abarcan desde aspectos técnicos hasta cuestiones éticas y de privacidad, y demandan una comprensión profunda y enfoques adaptados para asegurar implementaciones exitosas que realmente beneficien a los usuarios finales y a las organizaciones involucradas.

De ahí que, la presente investigación propone un *framework* para incorporar componentes de IA en arquitecturas de software preexistentes con énfasis en SST. El *framework* se compone de prácticas recomendadas, componentes arquitectónicos y criterios para una integración eficaz de una IA, buscando no solo la adaptación técnica sino también el aprovechamiento máximo de la IA para garantizar su impacto y perdurabilidad.

En particular, como caso de estudio, se selecciona un proyecto de desarrollo de software el cual incluye en sus requerimientos funcionales la necesidad de incorporar componentes de IA. La Corporación Talentum es una entidad prominente en la implementación de proyectos gubernamentales en Colombia.

Palabras Clave: Inteligencia Artificial (IA), Seguridad y Salud en el Trabajo, Ingeniería de Software (IS), *Framework*, *Prompt*.

0.2. Abstract

Despite the evident advances in the area of Artificial Intelligence (AI), its effective integration into software solutions focused on Occupational Health and Safety (OSH) presents challenges that range from technical aspects to ethical and privacy issues and demand deep understanding and tailored approaches to ensure successful implementations that truly benefit end users and the organizations involved.

Hence, the present research proposal proposes a framework to incorporate AI components into pre-existing software architectures with an emphasis on SST. The framework is made up of recommended practices, architectural components, and criteria for an effective integration of an AI, seeking not only technical adaptation but also the maximum use of AI to guarantee its impact and durability.

In particular, as a case study, a software development project was selected that includes in its functional requirements the need to incorporate AI components. The Talentum Corporation is an important entity in the implementation of government projects in Colombia.

Keywords: Artificial Intelligence (AI), Occupational Safety and Health (OSH), Software Engineering (SE), *Framework*, *Prompt*.

Índice general

0.1. Resumen	5
0.2. Abstract	5
1. Introducción	1
1.1. Motivación	1
1.2. Definición del problema	2
1.2.1. Planteamiento del problema	2
1.2.2. Formulación del problema	3
1.3. Objetivos del proyecto	3
1.3.1. Objetivo General	3
1.3.2. Objetivos específicos	3
1.4. Resultados esperados	4
1.5. Alcance	5
1.6. Justificación del trabajo de grado	6
1.7. Resumen de la Investigación	6
1.8. Estructura del Documento	8
2. Marco de referencia	9
2.1. Inteligencia Artificial	9
2.1.1. Breve Definición y Evolución	9
2.2. Arquitectura RAG	11
2.3. Seguridad y Salud en el Trabajo	12
3. Desarrollo del Framework	13
3.1. Lista de buenas prácticas	14
3.1.1. Modelos de lenguaje de gran tamaño LLM	14
3.1.2. Estructura de datos	17
3.1.3. Bases de Datos Vectoriales	18
3.2. Componentes arquitectónicos	21
3.2.1. Categorización de componentes	21
3.2.2. Listado de componentes arquitectónicos	23
3.3. Ingeniería de Prompt en SST	29
3.3.1. Buenas prácticas en el proceso de ingeniería de prompt	29
3.3.2. Criterios para el diseño de prompts	39
3.4. Gestión de recursos computacionales	41
3.4.1. Optimización de recursos	41
3.4.2. Listado de prácticas	41
3.5. Buenas practicas de arquitectura y adaptaciones	42

3.5.1. Adaptaciones arquitectónicas	42
3.5.2. Recomendaciones sobre estilos arquitectónicos	43
3.5.3. Desafíos y Tendencias en la Arquitectura Actual	48
4. Evaluación	51
4.1. Fuentes de datos de SST	52
4.2. Presentación del caso de estudio	52
4.2.1. Introducción	52
4.2.2. Descripción general	53
4.2.3. Requerimientos	54
4.2.4. Consideraciones	57
4.3. Arquitectura de software	58
4.3.1. Attribute Driven Design y Framework	58
4.4. Aplicación del Framework al caso de estudio	71
4.4.1. Aplicación de las buenas prácticas	71
4.4.2. Aplicación de componentes del Framework	80
4.4.3. Aplicación de ingeniería de prompt en SST	94
4.5. Pruebas y Validación	101
4.5.1. Evaluación de Arquitectura Ligera (LAE)	101
4.5.2. Evaluación con especialistas de SST	109
5. Conclusiones	115
5.1. Conclusiones generales	115
5.2. Trabajo futuro	119
5.2.1. Generalización del Framework a Otros Sectores	119
5.2.2. Propuesta Espacio Colaborativo para la Mejora Continua del Framework	120
5.2.3. Conclusiones	121
Bibliografía	123
A. Apéndice A: Fuente de Datos de SST	131
B. Apéndice B: Architecture Decision Records	139
C. Apéndice C: Tácticas de arquitectura de software	147
D. Apéndice D: Patrones arquitectónicos de software	153
E. Apéndice E: Lista de componetes de software del Framework	157

Índice de figuras

1.1. Resumen de la Investigación: Problema, Objetivos y Resultados.	7
3.1. Conversión de una tabla a un formato óptimo en texto plano.	18
3.2. Flujo de trabajo del funcionamiento de las bases de datos vectoriales	18
3.3. Diagrama de Componentes de IA	25
3.4. Proceso de ingeniería de prompt.	29
4.1. Diagrama de contexto.	59
4.2. Diagrama de contenedores del sistema de información de SST.	69
4.3. Diagrama de contenedores del sistema de IA.	69
4.4. Tipos de archivos de las fuentes de datos implementadas para el contexto de los modelos	75
4.5. Componente “Character Text Splitter” del orquestador Flowise	78
4.6. Indexación en Qdrant	80
4.7. Diagrama de alto nivel de la solución.	81
4.8. Interfaz gráfica del SI de SST	81
4.9. Chatflow desarrollado en Flowise	82
4.10. Código para embeber un chatbot de Inteligencia Artificial desarrollado en Flowise	83
4.11. Componente “Folder with Files” desde Flowise	84
4.12. Componente “Character Text Splitter”	84
4.13. Componentes de embedding para gpt-3.5-turbo y llama2-70b-chat-v1	85
4.14. Componente de “Qdrant” dentro del Chatflow	86
4.15. Componentes de LLM para llama2-70b-chat-v1 y gpt-3.5-turbo	87
4.16. Componente del Chatflow para la base de datos DynamoDB	88
4.17. Componente “Conversational Retrieval QA Chain” del Chatflow para iniciar el flujo de orquestación	89
4.18. Instrucciones establecidas en los parámetros adicionales del componente “Conversa- tional Retrieval QA Chain”	89
4.19. Diagrama de flujo de prompt de SST	90
4.20. Diagrama de flujo de vectorización de fuentes de datos de SST	93
4.21. Diagrama del proceso aplicado de Prompt Engineering en SST en la Corporación Talentum	94
4.22. Árbol de utilidad de los atributos de calidad	105
4.23. Gráfico de distribución según calificaciones obtenidas	110
4.24. Gráfico de distribución según calificaciones por pregunta	111
4.25. Gráfico de calificación promedio por pregunta	112

Índice de tablas

2.1. Comparativa entre LLMs	11
3.1. Ejemplo Meta Language Creation	30
3.2. Ejemplo Output Automater	30
3.3. Ejemplo Flipped Interaction	31
3.4. Ejemplo Persona	31
3.5. Ejemplo Question Refinement	31
3.6. Ejemplo Alternative Approaches	32
3.7. Ejemplo Cognitive Verifier	33
3.8. Ejemplo Fact Check List	34
3.9. Ejemplo Template	35
3.10. Ejemplo Reflection	35
3.11. Ejemplo Refusal Breaker	36
3.12. Ejemplo Context Manager	36
3.13. Ejemplo Recipe	37
3.14. Datos de ejemplo de estudiantes	38
3.15. Prompt de ejemplo	38
3.16. Prompt reformulado	39
4.1. Escenario de interoperabilidad	60
4.2. Escenario de rendimiento	61
4.3. Tabla de tácticas resumida	64
4.4. Tabla de patrones resumida	65
4.5. Responsabilidad de componentes arquitectónicos	67
4.6. ADR de interoperabilidad Núm. 03: Arquitectura RAG para extraer e interoperar información con la IA	71
4.7. Comparativa entre los modelos llama2-70b-chat-v1 y gpt-3.5-turbo	72
4.8. Limitaciones identificadas en los modelos gpt-3.5-turbo y llama2-70b-chat-v1	75
4.9. Prompt “Closed” generado por especialista de SST	95
4.10. Prompt “Opened” generado por especialista de SST	96
4.11. Prompt “Flipped Interaction” generado por especialista de SST	96
4.12. Prompt “Meta Language Creation” generado por especialista de SST	97
4.13. Prompt “Output Automater” generado por especialista de SST	97
4.14. Prompt “Persona” generado por especialista de SST	97
4.15. Prompt “Alternative Approaches” generado por especialista de SST	98
4.16. Prompt “Fact Check List” generado por especialista de SST	99
4.17. Prompt “Refusal Breaker” generado por especialista de SST	100

4.18. Preguntas del artefacto de evaluación para los especialistas de SST	110
A.1. Descripción de las columnas que conforman la matriz de riesgos de SST	137
B.1. ADR de interoperabilidad Núm. 01: Administrador de documentos	139
B.2. ADR de interoperabilidad Núm. 02: Orquestador de flujos LLM	140
B.3. ADR de rendimiento Núm. 01: Base de datos vectorial	141
B.4. ADR de rendimiento Núm. 02: Base de datos para almacenar chat en memoria . . .	142
B.5. ADR de rendimiento Núm. 03: Monitoreo de rendimiento de flujos de orquestación LLM	143
B.6. ADR de costos Núm. 01: Servicio serverless para desplegar LLM	144
B.7. ADR de costos Núm. 02: Servicio serverless para desplegar Embeddings	145
B.8. ADR de costos Núm. 03: Monitoreo de costos por LLM	146
C.1. Tácticas	151
D.1. Patrones	155

Introducción

Contents

1.1. Motivación	1
1.2. Definición del problema	2
1.2.1. Planteamiento del problema	2
1.2.2. Formulación del problema	3
1.3. Objetivos del proyecto	3
1.3.1. Objetivo General	3
1.3.2. Objetivos específicos	3
1.4. Resultados esperados	4
1.5. Alcance	5
1.6. Justificación del trabajo de grado	6
1.7. Resumen de la Investigación	6
1.8. Estructura del Documento	8

1.1. Motivación

La Inteligencia Artificial (IA) ha transformado diversos sectores a nivel global, mejorando la toma de decisiones, optimizando procesos y generando soluciones innovadoras para problemas complejos. Uno de los sectores que ha experimentado un gran potencial con la IA es el de la Seguridad y Salud en el Trabajo (SST), especialmente en contextos como el colombiano, donde la Corporación Talentum está a la vanguardia de esta revolución tecnológica.

La Seguridad y Salud en el Trabajo es un sector crítico para cualquier organización, ya que se enfoca en proteger la salud y el bienestar de los trabajadores en el lugar de trabajo. La integración de la IA en las soluciones de software para el sector SST puede mejorar significativamente la eficiencia y la efectividad de los procesos de Seguridad y Salud en el Trabajo. Por ejemplo, la IA puede utilizarse para monitorear y analizar datos en tiempo real sobre el estado de los trabajadores y las condiciones de trabajo, lo que permite una detección temprana de posibles riesgos y peligros. Además, la IA puede ayudar en la automatización de tareas repetitivas y peligrosas, reduciendo así el riesgo de lesiones y accidentes laborales.

A pesar de los beneficios potenciales de la integración de la IA en el sector SST, existen desafíos significativos que deben abordarse para lograr una implementación exitosa. También hay desafíos técnicos, como la necesidad de una infraestructura de TI adecuada y la integración de la IA con los sistemas existentes.

Para abordar estos desafíos, se propuso la presente investigación que presenta un *framework* para la incorporación de componentes de IA en arquitecturas de software preexistentes con énfasis en SST. Este framework consta de prácticas recomendadas, componentes arquitectónicos y criterios para una integración eficaz de la IA, buscando no solo la adaptación técnica sino también el aprovechamiento máximo de la IA para garantizar su impacto y perdurabilidad. Como caso de estudio, se seleccionó un proyecto de desarrollo de software que incluye la necesidad de incorporar componentes de IA en sus requerimientos funcionales. De esta manera, la investigación contribuye en la implementación exitosa de soluciones de software con IA en el sector SST, generando un impacto positivo en el bienestar laboral de las organizaciones involucradas.

1.2. Definición del problema

1.2.1. Planteamiento del problema

La problemática planteada en la integración de soluciones de IA en sistemas existentes es un desafío significativo para las organizaciones que buscan aprovechar los beneficios de la IA. La falta de un conjunto estandarizado de buenas prácticas para guiar la integración de la IA en arquitecturas de software preexistentes puede conducir a una serie de problemas, como incompatibilidades, redundancias, ineficiencias y fallas en el sistema. Esto puede resultar en sistemas sobrecomplicados, costosos y que no cumplen con las expectativas.

En el contexto de la Corporación Talentum, la necesidad de ejecutar proyectos gubernamentales con productos propios que incorporen IA se suma a la necesidad de mejorar sus procesos de Seguridad y Salud en el Trabajo. Sin embargo, la integración de soluciones de IA en el área de SST presenta desafíos únicos, especialmente en lo que respecta a la interoperabilidad con diversas IA que utilizan el estándar GGUF (GPT-Generated Unified Format). Lograr la interoperabilidad es esencial para garantizar que los sistemas de SST basados en IA puedan interactuar y comunicarse eficazmente entre sí, lo que es crucial para mejorar la eficiencia y la efectividad de los procesos de SST.

Además, la falta de un *framework* que proporcione directrices y mejores prácticas para la integración de productos basados en IA en arquitecturas de software existentes también puede afectar la viabilidad, escalabilidad y relevancia a largo plazo de las soluciones tecnológicas orientadas a la IA en el sector de la SST. Sin un enfoque sistemático y estandarizado para la integración de la IA, las organizaciones pueden encontrarse con sistemas que no son sostenibles a largo plazo, lo que puede limitar su capacidad para adaptarse a los cambios en el entorno empresarial y aprovechar las oportunidades emergentes.

Por lo tanto, es necesario abordar esta problemática de manera integral, mediante el desarrollo de un *framework* que proporcione un conjunto de directrices y mejores prácticas para la integración de productos basados en IA en arquitecturas de software existentes. Este *framework* debe tener en cuenta los desafíos únicos asociados con la integración de soluciones de IA en el área de SST, especialmente en lo que respecta a la interoperabilidad con diversas IA que utilizan el estándar GGUF. Al abordar estos desafíos de manera sistemática y estandarizada, las organizaciones pueden maximizar los beneficios de la IA, mejorar la eficiencia y la efectividad de sus procesos de SST, y garantizar la viabilidad, escalabilidad y relevancia a largo plazo de sus soluciones tecnológicas orientadas a la IA.

1.2.2. Formulación del problema

Dado el contexto anterior y teniendo en cuenta las necesidades actuales de la Corporación Talentum para integrar soluciones de IA en el sector de la SST surgen las siguientes interrogantes:

- ¿Cómo se pueden incorporar efectivamente componentes de IA en los requerimientos funcionales del software orientado al sector de SST?
- ¿Qué buenas prácticas, estándares y componentes arquitectónicos se deben adoptar para facilitar la integración de productos basados en IA en arquitecturas de software preexistentes, especialmente en el contexto de la SST?
- ¿Cuáles son los componentes potenciales de software y cuáles son sus responsabilidades para facilitar o mejorar la integración de la IA en las soluciones de software, de manera que se contribuya a otorgar beneficios técnicos y funcionales?
- ¿Qué consideraciones arquitectónicas son necesarias para gestionar adecuadamente los *prompts* de IA en el contexto de la SST, asegurando una comunicación eficiente y precisa con sistemas preexistentes?

1.3. Objetivos del proyecto

1.3.1. Objetivo General

Proponer un *framework* que integre componentes de Inteligencia Artificial en soluciones de software orientadas al sector de Seguridad y Salud en el Trabajo con base en los requerimientos funcionales.

1.3.2. Objetivos específicos

- OE1:** Elaborar una lista de buenas prácticas para la incorporación de funcionalidades de Inteligencia Artificial en arquitecturas de productos de software específicos para el sector de Seguridad y Salud en el Trabajo.

- OE2:** Proponer una lista de componentes arquitectónicos potenciales que no solo faciliten la integración de Inteligencia Artificial, sino que también aseguren la interoperabilidad en el desarrollo de soluciones de software, enfocándose en el sector de Seguridad y Salud en el Trabajo.
- OE3:** Definir un conjunto de criterios que permitan especificar el contexto necesario para que la Inteligencia Artificial proporcione respuestas de manera más precisa, enfocándose especialmente en la elaboración y gestión de *prompts*.
- OE4:** Identificar y proponer posibles adaptaciones arquitectónicas necesarias para la integración de componentes de Inteligencia Artificial en arquitecturas preexistentes de soluciones de software en el ámbito de la seguridad y salud laboral.

1.4. Resultados esperados

- R1 (OE1):** Una lista de buenas prácticas para la incorporación de funcionalidades de IA en productos de software específicos para el sector de SST. Esta lista se enfocará en la gestión de recursos computacionales, la selección de herramientas o librerías optimizadas para la tarea y la elección de modelos de IA específicos para la industria. Estas prácticas se basarán en el análisis de las necesidades computacionales del software para asignar recursos de manera eficiente, la evaluación de herramientas y librerías por su compatibilidad, rendimiento y soporte comunitario, y la selección de modelos de IA que no solo estén a la vanguardia tecnológica, sino que también se alineen estrechamente con los objetivos y procesos industriales específicos del producto.
- R2 (OE2):** Una lista de componentes arquitectónicos potenciales que faciliten la integración de IA y aseguren la interoperabilidad en el desarrollo de soluciones de software enfocadas en el sector de SST. Esta lista servirá como guía para identificar cada elemento necesario y su función dentro del sistema, incluyendo las entradas y salidas de información. Se requerirá una comprensión de cómo cada componente interactúa con el sistema existente y cómo el flujo de datos entre estos componentes impulsa las operaciones de IA. Esto ayudará a los diseñadores y desarrolladores a establecer fronteras de responsabilidad y a garantizar que todos los elementos de la arquitectura estén sincronizados para el rendimiento óptimo del sistema.
- R3 (OE3):** Un conjunto de criterios enfocados en SST para configurar adecuadamente el contexto al formular preguntas a un componente de IA, lo que resulta en respuestas más precisas y pertinentes. Estos criterios deberán abordar aspectos específicos como la naturaleza del entorno laboral, los riesgos potenciales y las normativas de seguridad aplicables. Al integrar estos elementos en los *prompts*, la IA puede ajustar sus respuestas para ofrecer información que no solo sea correcta, sino también aplicable y segura según las directrices de SST del contexto en cuestión.

R4 (OE4): Una lista de posibles adaptaciones arquitectónicas en arquitecturas de software preexistentes para la incorporación de componentes de IA en el ámbito de la SST. Esta lista abordará la expansión de la capacidad de cómputo, la mejora o adición de interfaces para la integración de nuevos componentes de IA y la actualización de dependencias obsoletas. Estas adaptaciones son fundamentales para superar los desafíos más comunes y para facilitar la transición hacia sistemas mejorados con inteligencia artificial, garantizando que la arquitectura existente se mantenga relevante y pueda soportar las nuevas cargas de trabajo y funcionalidades.

R5 (OE4): Una lista de recomendaciones para la modificación o ampliación de arquitecturas de software existentes, teniendo en cuenta las peculiaridades inherentes a los diferentes estilos arquitectónicos en la integración de la IA. Estos estilos incluyen arquitecturas monolíticas, microservicios y basada en servicios, proporcionando una guía para incorporar componentes de IA en el desarrollo evolutivo del software.

1.5. Alcance

El proyecto se enfoca en la integración de componentes de Inteligencia Artificial (IA) en soluciones de software específicas para el sector de Seguridad y Salud en el Trabajo. A partir del análisis de las funcionalidades y características de los sistemas de software basados en IA en el área de SST, se identificaron áreas de oportunidad donde la IA potencia o mejora dichas funcionalidades. El proyecto propone un compendio de buenas prácticas para la incorporación de funcionalidades de IA, que servirá como guía para futuras implementaciones. Adicionalmente, se presenta una lista de componentes arquitectónicos a considerar en la integración de la IA dentro de las soluciones de software ya existentes. Se definen criterios específicos para asegurar que la IA entregue respuestas precisas y relevantes, prestando especial atención a la elaboración y gestión del contexto que requieren los *prompts* enfocados en SST. Asimismo, se identificaron y propusieron adaptaciones arquitectónicas necesarias en las arquitecturas de software preexistentes para facilitar la integración de la IA, lo cual es esencial para una incorporación exitosa y eficiente. Para validar la propuesta del marco de trabajo, se desarrolló un prototipo funcional implementado en una plataforma web modular.

El proyecto no aborda el desarrollo de nuevos modelos de Inteligencia Artificial específicos para el sector de Seguridad y Salud en el Trabajo, sino que se enfoca en la integración de componentes de IA existentes en soluciones de software ya establecidas. Asimismo, no se consideran aspectos relacionados con la infraestructura física o los dispositivos de hardware necesarios para la implementación de los sistemas de software. La evaluación y selección de proveedores de servicios de IA tampoco forma parte del alcance de este proyecto. Finalmente, el proyecto no incluye la implementación a gran escala de las soluciones de software mejoradas con IA, sino que se limita al desarrollo y validación de un prototipo funcional en una plataforma web modular.

1.6. Justificación del trabajo de grado

La creciente necesidad de mejorar y agilizar los procesos en el sector de SST ha impulsado la búsqueda de herramientas tecnológicas avanzadas que complementen y potencien las soluciones actuales. Entre estas herramientas, la IA se ha destacado por ofrecer beneficios significativos en áreas como la automatización y el análisis predictivo, lo que ha motivado a organizaciones como la Corporación Talentum a reconocer la importancia de integrar tecnologías de IA en sus productos existentes destinados al sector SST.

Esta adaptación no solo responde a un deseo de optimizar procesos y aumentar la eficiencia, sino también a una estrategia para mantenerse competitivos frente a otras empresas que ya están aprovechando la IA en sus soluciones. En este sentido, el presente proyecto busca desarrollar un marco de buenas prácticas y componentes arquitectónicos que faciliten la integración de IA en soluciones de software específicas para el sector SST, así como definir criterios para asegurar la precisión y relevancia de las respuestas entregadas por la IA en el contexto de la SST.

La investigación y el desarrollo de este proyecto no solo benefician a la Corporación Talentum, sino que también establecen principios y marcos que pueden influir en el sector de SST en su conjunto, contribuyendo a la mejora y optimización de los procesos relacionados con la SST a través de la adopción de tecnologías de vanguardia como la Inteligencia Artificial.

1.7. Resumen de la Investigación

La Figura 1.1 ayuda a sintetizar la presente investigación teniendo en cuenta los aspectos más importantes tales como problema, objetivo y resultados esperados.

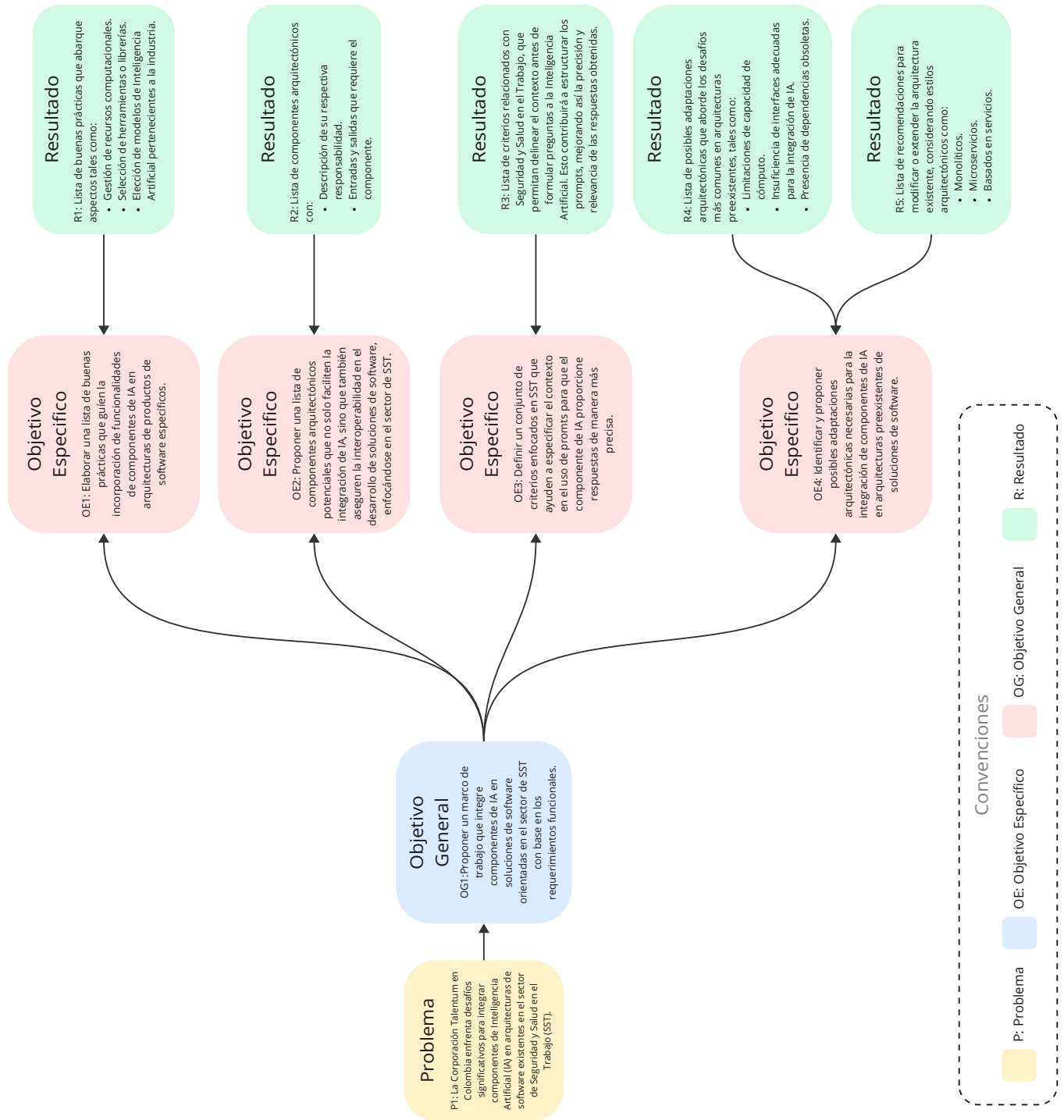


Figura 1.1: Resumen de la Investigación: Problema, Objetivos y Resultados.

1.8. Estructura del Documento

El resto del documento se estructura de la siguiente manera:

El Capítulo **Marco de referencia 2**, aborda las bases teóricas relacionadas con la **Inteligencia Artificial 2.1**, la **Arquitectura RAG 2.2**, y aspectos de **Seguridad y Salud en el Trabajo 2.3**.

El Capítulo **Desarrollo del Framework 3** detalla la **Lista de buenas prácticas 3.1**, **Componentes arquitectónicos 3.2**, **Ingeniería de Prompt en SST 3.3**, **Gestión de recursos computacionales 3.4**, y las **Buenas prácticas de arquitectura y adaptaciones 3.5**.

El Capítulo **Evaluación 4** se describe cómo se realizaron las correspondientes pruebas del framework propuesto con base a un **Presentación del caso de estudio 4.2** teniendo en cuenta las **Fuentes de datos de SST**.

El Capítulo **Conclusiones 5**, se exponen las **conclusiones generales 5.1** y se plantea el **Trabajo futuro 5.2**, como la **Generalización del framework a otros sectores** y la **Propuesta de un espacio colaborativo para la mejora continua del framework**.

Finalmente, se presenta la **Bibliografía 5.2.3** con lista las referencias bibliográficas utilizadas en la investigación, seguida de los **Apéndices 5.2.3** que incluyen fuentes de datos de SST, registros de decisiones arquitectónicas, tácticas de arquitectura de software, y patrones arquitectónicos de software.

Marco de referencia

Contents

2.1. Inteligencia Artificial	9
2.1.1. Breve Definición y Evolución	9
2.2. Arquitectura RAG	11
2.3. Seguridad y Salud en el Trabajo	12

El foco de esta investigación se encuentra en la intersección de la Inteligencia Artificial (IA), la Arquitectura de Software y la Seguridad y Salud en el Trabajo. En este capítulo, se presentan los conceptos fundamentales, la definición de términos, las arquitecturas y las tecnologías asociadas con cada una de estas áreas, estableciendo un sólido marco de referencia que sustenta la investigación y sus aplicaciones en el contexto de la IA aplicada a la SST. La comprensión de estos conceptos es esencial para apreciar la solución propuesta y su relevancia en el panorama actual, donde la tecnología y la seguridad laboral desempeñan roles cada vez más decisivos.

2.1. Inteligencia Artificial

2.1.1. Breve Definición y Evolución

La Inteligencia Artificial generativa se refiere a la capacidad de los sistemas de IA para generar contenido nuevo y original, como texto, imágenes, música y más, basándose en patrones y datos previamente aprendidos. Esta tecnología tiene el potencial de revolucionar la creación de contenido, permitiendo la producción automática de materiales de alta calidad y personalizados. Según el artículo de [Liu \(2023\)](#), la adopción y el uso de esta tecnología pueden promover una transformación inteligente dentro de la industria, aunque también plantea cuestiones éticas y de control de calidad que deben abordarse para garantizar su uso responsable y efectivo.

En este contexto, [Ospina and Aristizábal \(2021\)](#) presentaron una aplicación significativa de la IA en la evaluación de la susceptibilidad a los deslizamientos de tierra, específicamente en la cuenca del arroyo La Miel, ubicada en la cordillera de los Andes colombianos. En su investigación, emplearon diversos algoritmos de aprendizaje automático para comparar la capacidad predictiva de varios

modelos. Los resultados demostraron que los modelos de refuerzo ensamblados superaron significativamente a los modelos lineales paramétricos en términos de rendimiento y capacidad predictiva. Este estudio no solo resaltó la eficacia de la IA en la predicción y evaluación de zonas propensas a deslizamientos de tierra, sino que también subrayó la importancia de contar con inventarios detallados de deslizamientos de tierra y variables predictoras para el ajuste y desarrollo de modelos útiles para la toma de decisiones y la comprensión del fenómeno (Ospina and Aristizábal, 2021).

Entre las IA generativas más recientes se encuentran GPT-4 (Baktash and Dawodi, 2023), PaLM 2 (Anil et al., 2023), Claude (Adetayo et al., 2024), LLaMA (Meta, 2023; Touvron et al., 2023a) y Llama 2 (Touvron et al., 2023b). GPT-4, desarrollado por OpenAI, es una de las IA generativas más avanzadas, conocida por su capacidad para entender y generar texto de manera extremadamente coherente y relevante. PaLM 2, de Google, se destaca por su enfoque en la comprensión y generación de lenguaje natural con alta precisión. Claude, desarrollado por Anthropic, es conocido por su enfoque en la seguridad y la alineación ética de los modelos de IA. LLaMA y Llama 2, desarrollados por Meta, son modelos que se centran en la eficiencia y la escalabilidad, con el objetivo de proporcionar capacidades avanzadas de generación de texto de manera más accesible y sostenible.

La revisión de estos modelos de IA generativa permite destacar las características distintivas de cada uno, proporcionando una visión comparativa que facilita la comprensión de sus capacidades y aplicaciones potenciales. A continuación, la Tabla 2.1 presenta un resumen con las características más importantes, subrayando sus fortalezas y áreas de especialización.

	GPT-4	Palm 2	Claude 2	LLaMA	LLaMA 2
Creador	OpenAI	Google	Anthropic	Meta	Meta
Cantidad máxima de tokens de entrada	8.192 en su versión más básica	8.196	200.000	2.048	4.096
Parámetros de entrenamiento	Alrededor de 1.76 billones	430.000 millones de parámetros	130.000 millones de parámetros	7.000, 13.000, 33.000 y 65.000 millones	7.000, 13.000 y 7.0000 millones
Multilingüe	Si	Si	Si	Si	Si
Conexión a Internet	Si	Si	No	No	No
Continúa en la siguiente página					

	GPT-4	Palm 2	Claude 2	LLaMA	LLaMA 2
Licenciamiento	Accesible para los suscriptores de ChatGPT Plus	Uso gratuito no comercial por el momento	Acceso a uso comercial mediante pago, acceso gratuito con limitaciones para uso no comercial	Licencia de uso no comercial para investigadores, entidades afiliadas al gobierno, la sociedad civil y la academia	Gratuito para investigación y uso comercial
Modalidad	Texto e imagen	Texto	Texto	Texto	Texto
Código abierto	No	No	No	Si	Si

Tabla 2.1: Comparativa entre LLMs

2.2. Arquitectura RAG

La arquitectura RAG (Retrieve and Generate) se refiere a una metodología utilizada en el campo de la Inteligencia Artificial y el procesamiento del lenguaje natural que combina técnicas de recuperación de información y generación de texto. Este enfoque se basa en la idea de que, para proporcionar respuestas precisas y coherentes a consultas complejas, es beneficioso no solo generar texto basado en modelos preentrenados, sino también recuperar información relevante de una base de datos o un conjunto de documentos. La arquitectura RAG ha sido fundamental en mejorar la precisión y la relevancia de los sistemas de conversación y los asistentes virtuales, ya que permite integrar conocimientos específicos y actualizados directamente en las respuestas generadas por el modelo (Lewis et al., 2005,?).

Históricamente, la evolución de la arquitectura RAG se ha visto impulsada por avances en dos áreas principales: la recuperación de información y la generación de texto. A lo largo de la década de 2010, los motores de búsqueda y los algoritmos de recuperación de información experimentaron mejoras significativas gracias al uso de técnicas de aprendizaje automático y redes neuronales. Paralelamente, los modelos generativos como GPT-3 de OpenAI revolucionaron la capacidad de los sistemas de IA para producir texto coherente y relevante. La combinación de estas dos tecnologías dio lugar a la arquitectura RAG, que se perfeccionó a medida que los investigadores comprendieron mejor cómo integrar ambos componentes de manera efectiva (Karpukhin et al., 2020).

Uno de los beneficios más destacados de la arquitectura RAG es su capacidad para generar respuestas altamente informativas y contextualmente precisas. Al aprovechar bases de datos exten-

sas y actualizadas, los sistemas RAG pueden proporcionar información específica que un modelo generativo puro podría no tener. Esto es especialmente útil en dominios especializados donde la precisión de la información es crucial, como en la medicina o el derecho. Además, la capacidad de recuperar datos relevantes permite que los sistemas de IA mantengan su relevancia y precisión incluso cuando se enfrentan a preguntas sobre eventos recientes o información actualizada (Guu et al., 2020).

Sin embargo, la arquitectura RAG también presenta ciertas limitaciones. Una de las principales desventajas es la dependencia de la calidad y la actualidad de las bases de datos de las que se recupera la información. Si los datos están desactualizados o son incorrectos, las respuestas generadas también lo estarán. Además, la complejidad técnica de integrar componentes de recuperación y generación puede presentar desafíos significativos en términos de implementación y mantenimiento. A pesar de estos retos, la arquitectura RAG sigue siendo una herramienta poderosa y versátil en el ámbito de la Inteligencia Artificial y el procesamiento del lenguaje natural (Izacard and Grave, 2020).

2.3. Seguridad y Salud en el Trabajo

La Seguridad y Salud en el Trabajo es un ámbito multidisciplinario que tiene como objetivo la promoción y el mantenimiento del más alto grado de bienestar físico, mental y social de los trabajadores en todas las ocupaciones. Esto implica la prevención de riesgos laborales, la protección contra agentes perjudiciales y la adaptación del trabajo a las capacidades de los trabajadores, considerando tanto aspectos técnicos como organizativos (Saygılı and Çakmak, 2022; Rikhotso et al., 2022).

La historia de la SST se remonta a la Revolución Industrial, cuando las condiciones de trabajo comenzaron a recibir atención debido a la alta tasa de accidentes y enfermedades ocupacionales. En el siglo XX, se establecieron organismos internacionales como la Organización Internacional del Trabajo (OIT) en 1919 y se aprobaron leyes clave, como la Ley de Seguridad y Salud Ocupacional de 1970 en Estados Unidos. Estos avances reflejan un creciente reconocimiento de la importancia de la SST para el bienestar de los trabajadores y la productividad empresarial (McIvor, 2020; Rosner and Markowitz, 2020).

En los últimos años, la Inteligencia Artificial generativa ha emergido como una herramienta poderosa para mejorar la SST. La IA puede analizar grandes volúmenes de datos para identificar patrones y predecir riesgos, optimizando así las estrategias de prevención de accidentes. Además, tecnologías como los sistemas robóticos colaborativos y los exoesqueletos pueden reducir el estrés físico en los trabajadores y mejorar la seguridad en entornos peligrosos (Malenfer et al., 2023; DESHPANDE et al., 2017). En resumen, la integración de la IA en la SST no solo promete una mayor eficiencia en la gestión de riesgos, sino también un entorno laboral más seguro y saludable para todos.

Desarrollo del Framework

Contents

3.1. Lista de buenas prácticas	14
3.1.1. Modelos de lenguaje de gran tamaño LLM	14
3.1.2. Estructura de datos	17
3.1.3. Bases de Datos Vectoriales	18
3.2. Componentes arquitectónicos	21
3.2.1. Categorización de componentes	21
3.2.2. Listado de componentes arquitectónicos	23
3.3. Ingeniería de Prompt en SST	29
3.3.1. Buenas prácticas en el proceso de ingeniería de prompt	29
3.3.2. Criterios para el diseño de prompts	39
3.4. Gestión de recursos computacionales	41
3.4.1. Optimización de recursos	41
3.4.2. Listado de prácticas	41
3.5. Buenas practicas de arquitectura y adaptaciones	42
3.5.1. Adaptaciones arquitectónicas	42
3.5.2. Recomendaciones sobre estilos arquitectónicos	43
3.5.3. Desafíos y Tendencias en la Arquitectura Actual	48

Este capítulo presenta el Framework diseñado para abordar los desafíos del proyecto, que consiste en integrar Inteligencia Artificial con arquitecturas preexistentes que manejan datos de Seguridad y Salud en el Trabajo. Se detallan las buenas prácticas en el uso de Modelos de Lenguaje de Gran Escala (LLM), estructuras de datos, bases de datos vectoriales y componentes arquitectónicos, incluyendo una categorización y recomendaciones de componentes específicos. Además, se expone un conjunto de prácticas y criterios para la ingeniería de prompts en SST, junto con recomendaciones para la gestión eficiente de recursos computacionales y adaptaciones arquitectónicas enfocadas en estilos arquitectónicos particulares.

Es importante destacar que este Framework actúa como una guía en el proceso de diseño arquitectónico, siguiendo los principios del diseño dirigido por atributos, conocido como Attribute-Driven

Design (ADD). Sin embargo, se hace hincapié en que las prácticas para la creación de prompts difieren de las del diseño arquitectónico tradicional. Este Framework está orientado a proporcionar a la Corporación Talentum un conjunto de prácticas industriales en torno a componentes de Inteligencia Artificial que satisfacen los requisitos específicos del sector de SST. Cabe mencionar que, aunque se ofrecen directrices claras, estas no sustituyen el análisis crítico que debe realizar el arquitecto de software. Más bien, el Framework se ofrece como un conjunto de herramientas que facilita la resolución de problemáticas complejas de interoperabilidad relacionadas con SST.

3.1. Lista de buenas prácticas

3.1.1. Modelos de lenguaje de gran tamaño LLM

Los Modelos de Lenguaje de Gran Tamaño (LLMs) constituyen un avance notable en el ámbito de la Inteligencia Artificial y la comprensión del lenguaje natural. Estas herramientas potentes tienen la capacidad de impactar de manera positiva a una amplia variedad de campos, abarcando desde la traducción automática hasta la creación de contenido innovador. No obstante, su empleo efectivo requiere la implementación de las mejores prácticas para asegurar que su operación sea responsable, eficaz, segura y de confianza (Naveed et al., 2023).

En este contexto, resulta esencial aplicar una serie de prácticas esenciales al interactuar con LLMs. Estas prácticas no solo mejoran la eficiencia del modelo, sino que también promueven su aplicación ética y responsable. A continuación, se destacan las prácticas recomendadas que deben ser consideradas en el manejo de LLMs.

3.1.1.1. Selección del modelo adecuado

Determinar cuál LLM emplear representa una de las elecciones más significativas en el análisis de texto mediante LLMs. Aunque es posible desarrollar un modelo desde cero, la existencia de numerosos modelos ya creados, ya sean de código abierto o comerciales, ofrece la oportunidad de ahorrar tiempo y esfuerzo en desarrollo. Sobre estos se puede trabajar y/o realizar ajustes para adaptarlos al objetivo específico que se busca alcanzar mediante la implementación de un LLM (Törnberg, 2024).

En este contexto, se dispone de una amplia y variada selección de modelos, que abarcan desde pequeños modelos de código abierto capaces de operar en un teléfono móvil hasta grandes modelos de plataforma que se acceden a través de una interfaz web o API. Al considerar la implementación de uno de estos modelos preexistentes, es esencial tener en cuenta ciertos aspectos característicos de los LLMs para elegir adecuadamente cuál implementar:

- **Reproducibilidad:** Se debe poder replicar los resultados obtenidos por otros utilizando los mismos datos y metodología, lo que asegura la consistencia y fiabilidad de los resultados. Para esto, se recomienda usar una versión específica del LLM a lo largo del proyecto, documentar

dicha versión y asegurarse de que el modelo esté disponible para futuros usos (Törnberg, 2024).

- **Ética y legalidad:** El modelo debe adherirse a estándares éticos y legales respecto a las normativas vigentes sobre privacidad de datos (Törnberg, 2024).
- **Privacidad:** Es fundamental tener presente que, a nivel empresarial, la elección del modelo adecuado puede estar condicionada por leyes y políticas de privacidad específicas del sector o región. Por ejemplo, el uso de ciertos modelos puede estar limitado o prohibido debido a regulaciones como el Reglamento General de Protección de Datos (GDPR) en Europa o la Ley de Privacidad del Consumidor de California (CCPA) en Estados Unidos. Además, evaluar cómo el modelo maneja los datos personales y si ofrece garantías suficientes para la protección de estos datos. Por ejemplo, en el caso de ChatGPT, se almacena el historial de chat y otros datos personales, lo que puede representar un riesgo si no se manejan adecuadamente. Las empresas deben asegurarse de que el modelo seleccionado cumpla con las normativas de privacidad vigentes y ofrezca opciones claras para la supresión de datos y el control por parte de los usuarios (Törnberg, 2024).
- **Transparencia:** Las metodologías, fuentes de datos, suposiciones y limitaciones del modelo deben estar claramente documentadas y ser accesibles para su revisión (Törnberg, 2024).
- **Cultura e idioma:** Es esencial que el LLM ofrezca el soporte adecuado para el idioma y contexto cultural de los datos textuales. Es importante reconocer que algunos modelos son más competentes en determinados idiomas, lo cual puede impactar en la calidad de las respuestas obtenidas (Törnberg, 2024).
- **Escalabilidad:** Es fundamental verificar que el modelo posea la capacidad necesaria para gestionar el volumen de datos a implementar. Es relevante considerar que los modelos que operan de manera desconectada dependen del hardware disponible, mientras que aquellos accesibles mediante API están sujetos a límites de tasa y costos asociados (Törnberg, 2024).
- **Complejidad:** También es importante asegurarse de que el modelo sea capaz de manejar la complejidad de las tareas que se pretenden ejecutar, como por ejemplo, tareas de razonamiento avanzado o análisis de datos. Las tareas particularmente desafiantes pueden requerir modelos más grandes y sofisticados (Törnberg, 2024).

3.1.1.2. Comprensión de las limitaciones del modelo

Entender las limitaciones de un LLM es un paso fundamental en la creación de contenido. Las prácticas claves incluyen:

- Ser conscientes de las capacidades y limitaciones de estos modelos permite su uso seguro, eficiente y responsable (Törnberg, 2024).

- Reconocer las limitaciones de los LLMs posibilita el empleo del modelo de manera más efectiva, minimiza sus desventajas y mejora la calidad y confiabilidad general del contenido generado (Törnberg, 2024).
- Facilita el establecimiento de expectativas realistas, la identificación de errores, la interpretación del material en su contexto, la mejora de las instrucciones proporcionadas al modelo, la aplicación de criterios éticos y el progreso gradual en el desarrollo del modelo (Törnberg, 2024).

3.1.1.3. Calidad y diversidad de los datos

Los datos diversos y de buena calidad son fundamentales para el entrenamiento efectivo de modelos de lenguaje de gran tamaño. Entre las mejores prácticas se incluyen:

- La verificación de la fuente de datos para asegurar su fiabilidad y relevancia para el modelo, considerando el contexto de su aplicación. Es vital garantizar que los modelos se entrenen con datos no sesgados y que no se utilicen para discriminar a ningún grupo o proporcionar información sesgada a los usuarios (Naveed et al., 2023).
- La diversificación de los datos para incluir un amplio espectro de información, abarcando diferentes idiomas y contextos, con el fin de disminuir los sesgos y aumentar la precisión del modelo (Naveed et al., 2023).

3.1.1.4. Consideraciones éticas y mitigación de sesgos

La ética y la disminución de sesgos son aspectos fundamentales en el empleo de LLMs, lo que implica asegurar que los sistemas sean justos, equitativos y actúen de manera éticamente responsable en su interacción con los usuarios y en el manejo de datos. Las prácticas clave incluyen:

- Implementar mecanismos para detectar y eliminar sesgos en los conjuntos de datos antes de su uso en el entrenamiento de modelos de IA (Naveed et al., 2023).
- Establecer directrices éticas claras que rijan el uso y los resultados obtenidos con los LLMs (Esther and Tella, 2023).
- Entrenar los modelos de IA con datos representativos de la población objetivo para evitar sesgos y discriminación (Naveed et al., 2023).
- Aplicar técnicas para la disminución de sesgos, tales como la regularización y el muestreo de datos, entre otras estrategias disponibles (Naveed et al., 2023).

3.1.2. Estructura de datos

La aplicación de buenas prácticas en la estructura de datos es esencial para la optimización del rendimiento y la eficiencia de las bases de datos vectoriales y los Modelos de Lenguaje Grande (LLM por sus siglas en inglés). Una organización meticulosa de la información en los documentos que alimentan estas bases de datos y modelos no solo mejora la comprensión y el procesamiento de datos, sino que también aumenta significativamente la calidad de las respuestas generadas.

A continuación, se presentan algunas buenas prácticas de estructuración de datos que establecen las bases para un rendimiento superior y una generación de lenguaje más precisa y contextual:

- Es fundamental comprender los requisitos específicos de la aplicación que se va a desarrollar antes de diseñar la estructura de los datos. El tipo de documentos de los cuales se obtendrán los datos que se almacenarán depende de estos requisitos. Factores como el tipo de información almacenada, cómo se accederá a ella o cuáles podrían ser las consultas más frecuentes son aspectos cruciales a considerar (Saito et al., 2024).
- Identificar el propósito de la base de datos vectorial es igual de importante, ya que la estructura de los datos puede variar si se utiliza para búsqueda semántica, recomendaciones, generación de texto, entre otras funciones (Saito et al., 2024).
- Dado el amplio rango de documentos de los cuales se pueden extraer datos, es importante identificar el método a través del cual se insertarán los datos de cada archivo en la base de datos vectorial. Cada documento debe ser procesado para convertir su información en texto plano antes de proceder con la vectorización o embedding (Unstructured, 2024).

Trabajar con archivos de texto plano, como los .TXT o .CSV, generalmente requiere menos esfuerzo de preparación y procesamiento. Dependiendo de su estructura, estos pueden estar listos para ser almacenados sin necesidad de pasos adicionales. Por otro lado, archivos como .PDF, .DOCX, o .XLS requieren una mayor preparación ya que deben ser convertidos a texto plano legible para el modelo. Estos últimos pueden contener tablas, imágenes o gráficos con textos relevantes que se deseen almacenar, pero que inicialmente no serían legibles por la base de datos vectorial o el modelo sin el procesamiento adecuado (Törnberg, 2023).

- Para almacenar óptimamente documentos que no tienen una estructura de texto plano, se debe realizar un procesamiento específico. Esto es especialmente relevante para archivos como .PDF o .DOCX que pueden tener una estructura de títulos y formatos de texto. Un método es la implementación de algoritmos de OCR (Optical Character Recognition), que permiten identificar y convertir el texto de documentos, incluso los provenientes de imágenes escaneadas, en texto plano (Jana et al., 2014).
- Algunos documentos también pueden contener tablas con datos que pueden ser leídos por un algoritmo de OCR. Sin embargo, debido a su organización específica, estos pueden no mantener el mismo orden al ser convertidos a texto plano, lo que podría generar lecturas o

resultados inexactos (Unstructured, 2024). En estos casos, se requiere de un procesamiento adicional que organice dichos contenidos en un formato que facilite su lectura por la base de datos vectorial o el LLM, como se muestra en la figura 3.1.

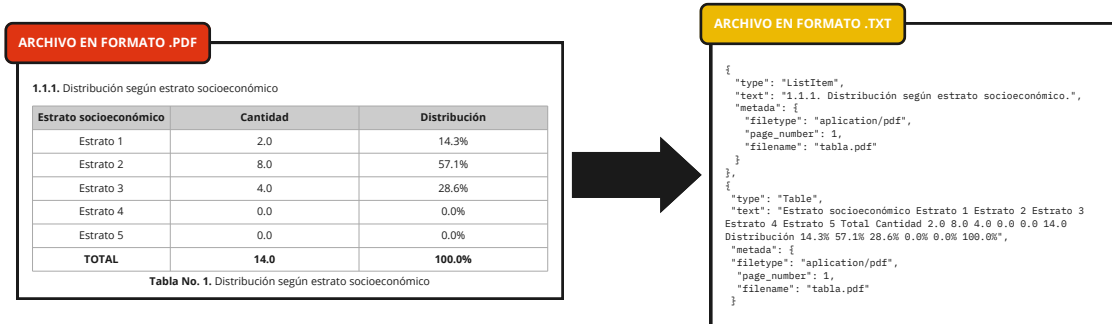


Figura 3.1: Conversión de una tabla a un formato óptimo en texto plano.

- Para los LLM capaces de generar conversaciones a partir de documentos de audio, imágenes y vídeos, se necesita un procesamiento más complejo. Estos documentos deben someterse a un proceso de codificación para poder ser almacenados y leídos por el modelo. Para archivos de imágenes o vídeos, se puede utilizar un codificador de modalidad visual como CLIP, que aprovecha un rango más amplio de supervisión al aprender directamente de datos textuales no procesados relacionados con imágenes. Asimismo, para los archivos de audio, se puede emplear un codificador de modalidad de audio como el modelo de reconocimiento de habla multilingüe WHISPER, entrenado en un extenso conjunto de datos de audio con supervisión débil (Lyu et al., 2023).

3.1.3. Bases de Datos Vectoriales

La adopción de buenas prácticas en la gestión de bases de datos vectoriales no solo mejora el rendimiento sino que también asegura la precisión y la integridad de los datos. Estas prácticas abarcan desde la limpieza o transformación de los datos hasta la implementación de algoritmos de embedding eficaces y la realización de copias de seguridad de manera regular. Al aplicar estos principios, se maximiza el uso de las bases de datos vectoriales, permitiendo realizar análisis espaciales de manera más efectiva.

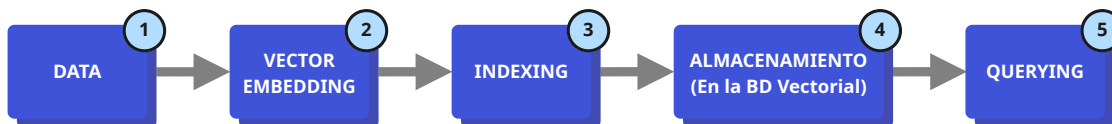


Figura 3.2: Flujo de trabajo del funcionamiento de las bases de datos vectoriales

El diagrama de la Figura 3.2 ilustra las fases clave en el funcionamiento de las bases de datos vectoriales: comienza con la recepción de los datos, seguido por su conversión a vectores mediante

algoritmos de embedding, luego la indexación de estos vectores para su almacenamiento eficiente, y finalmente la consulta o “querying” de la información almacenada, facilitando así el acceso y análisis de los datos.

Las etapas en la operatividad de las bases de datos vectoriales, como se ilustra en la figura 3.2, requieren una consideración cuidadosa de las siguientes buenas prácticas en cada una de sus fases.

3.1.3.1. Data

- Es fundamental identificar el tipo de datos que se insertarán en la base de datos vectorial, que generalmente son no estructurados o semi-estructurados (Eberendu, 2016).
- Se debe determinar un espacio y lugar adecuados para el almacenamiento de los datos a insertar, ya sea localmente o en la nube, teniendo en cuenta el volumen que estos pueden representar, especialmente para datos no estructurados como vídeos o audios (Eberendu, 2016).
- Es necesario llevar a cabo una revisión de los datos para asegurar su integridad, eficacia y correcta funcionalidad dentro de la base de datos vectorial (Taleb et al., 2018).
- En caso de ser necesario, se debe proceder con un proceso de limpieza o transformación de los datos, especialmente con aquellos no estructurados, para facilitar su almacenamiento en la base de datos vectorial y su posterior comprensión por el modelo de lenguaje que extraerá información de esta (Taleb et al., 2018).

3.1.3.2. Vector Embedding

- Aunque es factible desarrollar un algoritmo de embedding propio, esto no es indispensable dependiendo de las necesidades del proyecto. Actualmente, existen numerosos algoritmos de código abierto que se adaptan a diversos contextos, lo que permite ahorrar tiempo y esfuerzo en el desarrollo (Taipalus, 2024).
- Al emplear algoritmos de embedding ya existentes, es importante considerar que su rendimiento varía según el tipo de datos o archivos con los que se trabaje. Por ello, es esencial identificar el algoritmo utilizado y comprender su funcionamiento para determinar si es el más adecuado para lograr los resultados deseados (Taipalus, 2024).
- Si el proyecto demanda el desarrollo de un algoritmo de embedding específico, es relevante considerar que los vectores se asocian a palabras individuales. Para cada texto, se debe aplicar algún método que unifique los vectores de cada palabra, generando un vector único (Tomás López-Solaz et al., 2016).
- En caso de requerirse un algoritmo de embedding específico, es importante prestar atención a cómo se asignan los vectores, manteniendo la similitud semántica de las palabras. Esto se

logra mediante un proceso de alineamiento, que permite ubicar vectores cercanos entre sí, incluso si provienen de textos diferentes pero tienen similitud semántica (Tomás López-Solaz et al., 2016).

3.1.3.3. Indexing

- Aunque es posible desarrollar un algoritmo de indexación propio, esto no es indispensable dependiendo de las necesidades del proyecto, ya que existen numerosos algoritmos de código abierto adaptados a diversos contextos que pueden ahorrar tiempo y esfuerzo en el desarrollo (Taipalus, 2024).
- Al implementar algoritmos de indexación existentes, es importante recordar que su eficiencia varía según el tipo de datos o archivos con los que se trabaje, lo que subraya la importancia de elegir el algoritmo adecuado y comprender su funcionamiento para asegurar los resultados esperados (Taipalus, 2024).
- La indexación suele ser realizada por la base de datos vectorial elegida. Sin embargo, es importante tener en cuenta que los métodos de indexación tradicionales no son los más adecuados para vectores de alta dimensionalidad. Por ello, se recomienda seleccionar una base de datos vectorial que utilice técnicas especializadas para reducir la complejidad y mejorar la precisión en la búsqueda de similitudes (Han et al., 2023).

3.1.3.4. Almacenamiento en la Base de Datos Vectorial

- Al elegir una base de datos vectorial, es recomendable entender las características de los datos antes de tomar una decisión y considerar aspectos como la optimización de hardware y software, la escalabilidad, el rendimiento y la seguridad (Nexla, 2024).
- Es importante saber calcular qué capacidad de almacenamiento tiene que ser destinada para la base de datos vectorial teniendo en cuenta que este cálculo puede variar dependiendo de la base de datos a implementar (Nexla, 2024). De este modo, si por ejemplo se tienen 100 documentos cada uno con 1.000 palabras, y se desea almacenar en un espacio vectorial de 300 dimensiones, el resultado sería un total de 30.000.000 vectores, lo cual en el caso de bases vectoriales como por ejemplo Pinecone o Qdrant, tendrían un peso aproximado de 120 MB (Megabytes) ya que en ambas bases de datos los vectores tienen un peso aproximado de 4B (Bytes).
- Como se mencionó anteriormente, el proceso de indexación es llevado a cabo por la base de datos vectorial, lo que representa otro factor a considerar al seleccionar la base de datos a implementar (Han et al., 2023; Kumar Gupta et al., 2023).
- En caso de que los documentos a almacenar contengan textos extensos, no es recomendable almacenar esta información en un único vector. Lo ideal es dividir el texto en partes mediante un proceso de chunking o splitting (Gong et al., 2020).

- Como buena práctica en el diseño de bases de datos, se recomienda utilizar nombres de variables que sean fáciles de identificar, por ejemplo, “datos_agropecuarios_2024” en lugar de “dagro24” (Giménez, 2019).
- Es fundamental realizar copias de seguridad de los datos almacenados constantemente, trabajando con una base de datos vectorial que permita generar y restaurar copias de las colecciones si fuera necesario (Giménez, 2019).

3.1.3.5. Querying

- Tanto el proceso de embedding como el de indexación y consulta pueden ser ejecutados mediante algoritmos de código abierto. Las bases de datos vectoriales pueden realizar su propio proceso de consulta, así como implementar modelos de lenguaje como ChatGPT u Ollama. Sin embargo, si se decide desarrollar un algoritmo propio, es importante considerar que este proceso se optimiza mediante el uso de cálculos de distancia de puntos vectoriales, como la similaridad del coseno, y la implementación de algoritmos de aprendizaje automático como K-Nearest Neighbors (KNN) o Artificial Neural Network (ANN) (Pan et al., 2023).
- Proporcionar instrucciones claras y específicas facilita la generación del resultado deseado, mientras que la ambigüedad puede resultar en respuestas que no cumplan con las expectativas (ChatGPT 4 and Sabit Ekin, 2023).
- Especificar restricciones en el contexto, como formato, longitud o alcance, ayuda a generar respuestas que cumplan con las limitaciones deseadas (ChatGPT 4 and Sabit Ekin, 2023).
- Incorporar contexto o ejemplos en las consultas puede generar respuestas más precisas y relevantes, lo cual es particularmente útil al tratar con conceptos abstractos o dominios especializados (ChatGPT 4 and Sabit Ekin, 2023).

3.2. Componentes arquitectónicos

3.2.1. Categorización de componentes

Dentro de esta sección se aborda la clasificación de los componentes esenciales en el ámbito de la Inteligencia Artificial, desglosando su función y contribución al desarrollo e implementación de sistemas avanzados. A través de una exploración detallada, se identifican diversas categorías dentro del ciclo de vida de proyectos de IA, desde el procesamiento del lenguaje natural hasta la gestión eficaz de datos y la orquestación de servicios. Esta clasificación no solo facilita la comprensión de cómo estos componentes se interrelacionan para potenciar soluciones inteligentes, sino que también resalta la importancia de cada categoría en el fortalecimiento de la capacidad, eficiencia, y adaptabilidad de los sistemas de IA en resolver tareas complejas y ofrecer valor en un campo muy variados de aplicaciones (Gao et al., 2023).

3.2.1.1. Tareas de Procesamiento del Lenguaje Natural (NLP)

Esta categoría se centra en el Procesamiento del Lenguaje Natural (NLP), una disciplina interdisciplinaria en la intersección de la lingüística, la ciencia de la computación y la Inteligencia Artificial, la cual se especializa en mejorar la comunicación entre seres humanos y máquinas utilizando el lenguaje natural. Los componentes dentro de este campo están enfocados en las tareas que permiten interpretar, generar y manipular el lenguaje humano de manera efectiva, alcanzando un nivel significativo de comprensión (Khurana et al., 2023).

Dichas tareas se llevan a cabo por medio de componentes capaces ejecutar actividades de procesamiento del lenguaje natural como lo son los Modelos de Lenguaje de Gran Tamaño (LLMs), los cuales logran esto por medio de un entrenamiento intensivo con grandes cantidades de texto, lo que les permite comprender la estructura y los matices del lenguaje humano. La aplicación de estos componentes abarca una variedad de funciones, incluyendo la generación automática de contenido, la traducción de idiomas, la implementación de sistemas avanzados de respuesta a preguntas y el análisis de sentimientos (Sarker, 2024).

3.2.1.2. Configuración y Ejecución LLM

La configuración y ejecución de modelos de Inteligencia Artificial, especialmente aquellos de gran envergadura, necesitan una infraestructura técnica sólida y versátil. Esta categoría engloba componentes que simplifican el despliegue de modelos de IA, habilitando su funcionamiento tanto en entornos locales como en la nube. La facilidad para configurar y ejecutar es decisiva para la experimentación y el despliegue eficiente de soluciones IA, adaptándose a diversas necesidades de rendimiento y escalabilidad (Gao et al., 2023).

3.2.1.3. Orquestación

En el ámbito de la Inteligencia Artificial, la orquestación alude a la coordinación y gestión automatizada de diversos componentes y servicios de IA, colaborando conjuntamente para llevar a cabo flujos de trabajo complejos y tareas específicas. Esta categoría es esencial para la integración de diferentes servicios de IA, como agentes autónomos y modelos de lenguaje, posibilitando su operación sincronizada. La orquestación potencia la eficiencia operativa y facilita a los desarrolladores la creación de sistemas más complejos y funcionales (Kokkonen et al., 2022).

3.2.1.4. Almacenamiento

El almacenamiento ocupa un rol primordial en la infraestructura de cualquier sistema de IA, administrando la conservación y acceso a amplios volúmenes de datos, a los cuales podrá acceder la IA para nutrir su conocimiento. Esta categoría incluye componentes diseñados para el almacenamiento eficiente de datos en variados formatos y con diferentes fines, como documentos de texto, archivos multimedia y/o registros de conversaciones previas. Optimizar el almacenamiento es im-

portante para el procesamiento veloz de consultas y la conservación de una base de conocimiento actualizada y accesible (Gao et al., 2023).

3.2.1.5. Transformación de Datos

Esta categoría abarca a los componentes que realizan diferentes procesos de transformación a los datos a implementar, los cuales pueden ir desde transformar información textual en vectores numéricos para ser posicionados dentro de un espacio multidimensional (Jeong, 2024), hasta modificar la estructura y el formato del contenido de los datos con el fin de poder facilitar su procesamiento y análisis por algoritmos de IA (Jeong, 2024).

Esta categoría es vital para la interpretación y manipulación de datos por modelos de IA, permitiendo una representación semántica de la información comprensible para las máquinas ya que la calidad de los vectores y la estructura en la que se almacenan los datos afecta directamente la eficiencia de las tareas de IA (Han et al., 2023).

3.2.1.6. Carga de Información

La carga de información es clave para proveer a los sistemas de Inteligencia Artificial con datos nuevos y pertinentes. Esta categoría abarca componentes que facilitan la importación de datos al sistema, permitiendo el acceso a fuentes de conocimiento que permiten nutrir el conocimiento y la información de contexto de los modelos de lenguaje de la IA. Estos componentes son decisivos para mantener la base de datos del sistema actualizada, elevando la precisión y pertinencia de las respuestas generadas por la IA (Sarker, 2024).

Los modelos admiten la incorporación de archivos en múltiples formatos, como JSON, CSV, TXT, DOCX, o PDF, entre otros, asegurando que el sistema se beneficie de un amplio espectro de fuentes de datos y conocimiento (Jeong, 2024).

Cada una de estas categorías contribuye de manera notable al desarrollo, implementación y funcionamiento de sistemas avanzados de Inteligencia Artificial. La interacción y equilibrio entre estos componentes son determinantes para la eficiencia, adaptabilidad y eficacia de las soluciones de IA en abordar tareas complejas y proporcionar valor en un amplio abanico de aplicaciones.

3.2.2. Listado de componentes arquitectónicos

A continuación, se explora un listado de componentes de Inteligencia Artificial que abarcan desde la simple capacidad de cargar un documento hasta el procesamiento de información mediante Lenguajes de Modelos Grandes (LLMs) y agentes de IA. Cada componente juega un papel esencial en el desarrollo, configuración y despliegue de IA avanzadas, interconectándose para conformar soluciones potentes capaces de analizar extensos volúmenes de datos, tomar decisiones de manera autónoma y aprender de sus contextos.

Para ofrecer una comprensión más profunda de cómo operan estos componentes, se incluye un diagrama (Jeong, 2024) que muestra, de manera general, cada componente según su categoría y cómo se interconectan para asegurar un funcionamiento adecuado.

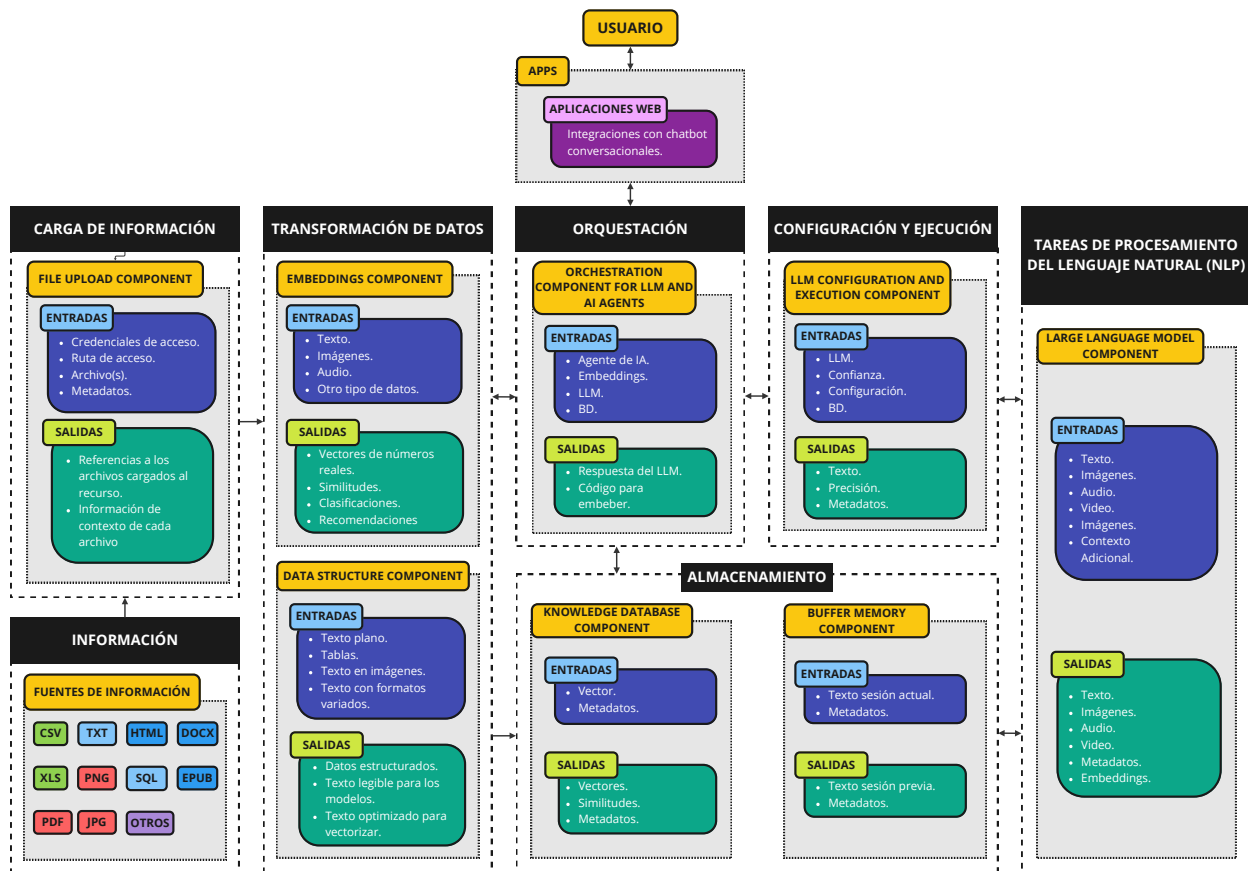


Figura 3.3: Diagrama de Componentes de IA

En el diagrama muestra el un punto importante que es la carga de información a partir de las diferentes fuentes, las cuales una vez cargadas, de ser necesario, tienen que pasar por el “Data Structure Component” (Componente de Estructura de Datos) dentro del cual se procede a organizar la estructura interna de los documentos para que su texto esté más optimizados tanto para el proceso de embedding como para la lectura por parte del LLM.

Una vez se han estructurado los datos, se procede a la vectorización de estos mediante el “Embedding Component” (Componente de Incrustación) el cual no solo se conecta con la base de datos vectorial para almacenar los textos convertidos a vectores, sino que a su vez se encuentra conectado al componente de orquestación para posteriormente también convertir a vectores los queries resultantes de los prompts ingresados por el usuario.

El “Orchestration Component for LLM and AI Agents” (Componente de Orquestación para LLM y Agentes de IA) es el encargado de la gestión del flujo de trabajo para automatizar y opti-

mizar procesos y se conecta entre los agentes de IA, los LLMs, las bases de datos y los modelos de embeddings. Es por medio de este orquestador que los embeddings creados a partir de los documentos cargados son almacenados en la base de datos vectorial correspondiente dentro del “Knowledge Database Component” (Componente de Base de Datos de Conocimiento) con el fin de dotar con nuevos contextos y conocimientos al LLM.

Del mismo modo, el “Buffer Memory Component” (Componente de Memoria de Búfer) se encuentra conectado por medio del orquestador al “LLMs Configuration and Execution Component” (Componente de Configuración y Ejecución de LLMs) con el fin de poder almacenar el historial de conversaciones previas entre el usuario y el LLM y así dotar a la IA de la capacidad de revisar y recuperar conversaciones anteriores, lo que facilita el seguimiento del historial de interacciones y la resolución de problemas.

El “LLMs Configuration and Execution Component”, se encarga de las conexiones previas a la ejecución del modelo, así como de su configuración para garantizar un correcto funcionamiento de acuerdo a las necesidades requeridas y se conecta con el “Large Language Model Component” el cual es el paso final para llegar a la ejecución de la interacción entre el usuario y la IA a través de las conexiones generadas con el “Orchestration Component for LLM and AI Agents”, el cual finalmente se conecta con una aplicación que permita la integración de un chatbot conversacional mediante el cual el usuario podrá digitar los prompts que ingresaran al LLM para este proceder a ejecutar todo el proceso descrito en el diagrama 3.3 y poder generar una respuesta al usuario.

A continuación, se describen la funcionalidad y las características de un componente, en este caso, File Upload Component. Este componente tiene una estructura la cual incluye la categoría, las entradas, las salidas, el caso de uso, las consideraciones (opcional) y un ejemplo.

3.2.2.1. File Upload Component

Permite la carga de archivos tipo JSON, CSV, TXT, DOCX, PDF, entre otros, a un recurso accesible por el componente de orquestación para Agente de IA y el LLM. La información contenida en estos archivos es almacenada en forma de vectores en el componente de base de datos de conocimiento y posteriormente usada por el LLM para dar respuesta a preguntas realizadas por el usuario, mejorando así la calidad de las respuestas generadas por el LLM al basar el modelo en fuentes externas de conocimiento para complementar la representación interna de la información del LLM.

- **Categoría:** Carga de información.
- **Entradas:**
 - Credenciales de acceso: Credenciales de acceso al recurso donde se deben alojar los archivos.

- Ruta de acceso al recurso.
 - Ruta donde se deben alojar los archivos.
 - Archivo o colección de archivos: Archivos que el componente debe cargar al recurso.
 - Metadatos sobre los archivos: Información cómo por ejemplo el nombre, el tipo, la fecha de creación, etiquetas asociadas, etc. Estos metadatos pueden ser útiles para organizar y categorizar los archivos
- **Salidas:**
- Referencias a los archivos cargados al recurso: Estas referencias podrían ser enlaces URL, rutas de archivo locales o identificadores únicos que permitan acceder a los archivos cuando sea necesario.
 - Información de contexto de cada archivo: Información cómo por ejemplo la ubicación del archivo, el usuario que lo cargó, la fecha y hora de carga, entre otros.
- **Casos de uso:** Carga de archivos que podrá usar el LLM para diferentes actividades cómo por ejemplo:
- Personalización de experiencias de usuario.
 - Recomendación de contenido.
 - Traducción de textos.
 - Clasificación de documentos.
 - Extracción de información.
- **Consideraciones:**
- Seguridad: El componente de carga de archivos debe ser seguro para proteger los archivos cargados, es importante considerar técnicas o mecanismos de seguridad para garantizar la confidencialidad e integridad de los archivos.
 - Rendimiento: El componente de carga de archivos debe ser eficiente para poder manejar grandes cantidades de archivos así cómo también archivos de gran tamaño.
 - Formato de Archivo: Se debe estar seguro que el componente puede manejar los tipos de archivo requeridos por el proyecto y que tenga la posibilidad de incluir nuevos tipos.
- **Ejemplo:**
- Filebrowser: Herramienta que promueve la carga y gestión eficiente de archivos dentro de un entorno de IA, permitiendo a los usuarios subir, organizar y compartir archivos en rutas específicas donde intervienen componentes de orquestación. Esto agiliza la actualización de conocimientos y la inclusión de nuevos datos en el sistema, fortaleciendo la capacidad de la IA para ofrecer soluciones informadas y actualizadas (Dias, 2019).

De igual manera, se desarrollaron otros componentes que se basan en la misma estructura. Para obtener detalles sobre estos componentes consulte el Apéndice E. Estos componentes son:

- **Data Structure Component.** Componente encargado de organizar la estructura de la información contenida dentro de los documentos cargados. Cambia la forma en que se encuentra la disposición de los textos y demás contenidos para facilitar el proceso de vectorización y que los contenidos sean más legibles para el modelo.
- **Embeddings Component.** Encargado de convertir palabras, frases o incluso documentos, en vectores de números reales, estos vectores se pueden utilizar para representar el significado semántico de las palabras o frases, dicho en otras palabras, es el encargado de representar información de forma matemática. Los embeddings comprimen información compleja en vectores de números reales que son más fáciles de procesar por las máquinas.
- **Orchestration Component for LLM and AI Agents.** Componente de gestión de flujo de trabajo para automatizar y optimizar procesos, permite crear una conexión entre Agentes de IA, LLM (Large Language Models), bases de datos y modelos de embeddings, para que los agentes de IA tengan la capacidad de interactuar con usuarios o entornos de manera autónoma y realizar tareas específicas.
- **Knowledge Database Component.** Encargado de la persistencia de los datos, además de permitir un procesamiento rápido y eficiente de grandes cantidades de datos textuales, debido a que almacena y organiza la información en forma de vectores. Estos vectores son representaciones matemáticas multidimensionales que codifican información sobre entidades o conceptos. Este componente actúa como una fuente externa de conocimiento para complementar la representación interna de la información del LLM, garantizando que el modelo tenga acceso a los hechos más actuales y confiables.
- **Buffer Memory Component.** Componente que permite al agente de IA y al LLM recordar conversaciones que ha tenido con los usuarios, ofreciendo así un servicio más personalizado y mejorando la experiencia del usuario.
- **LLMs Configuration and Execution Component.** Permite configurar y ejecutar los LLMs (Large Language Models) de manera local, haciendo uso de la CPU o GPU de la máquina donde se está ejecutando.
- **Large Language Model Component.** Componente que está previamente entrenado con grandes cantidades de datos para aprender y reproducir la estructura del lenguaje, puede realizar diversas tareas de procesamiento del Lenguaje Natural (NLP) como por ejemplo, generar y clasificar texto, responder a preguntas de forma conversacional y traducir texto de un idioma a otro.
- **Documents Storage Component.** Este componente está diseñado para el almacenamiento de archivos no estructurados como documentos en formatos Excel, PDF, Word, entre otros.

Posteriormente, estos documentos son vectorizados para su almacenamiento en una base de datos vectorial, permitiendo así que la Inteligencia Artificial acceda fácilmente a la información. Aunque este componente es opcional, ya que su funcionalidad podría ser integrada en el componente orquestador, incluyendo los documentos directamente en él, ofrece la posibilidad de separar la responsabilidad del almacenamiento de otros componentes de software sin impactar su rendimiento.

3.3. Ingeniería de Prompt en SST

En esta sección, se discute el uso del prompt o la ingeniería de prompt en la Inteligencia Artificial pero sobre el contexto de Seguridad y Salud en el Trabajo, que emerge como una herramienta esencial impulsando avances significativos en diversas disciplinas. Sin embargo, para optimizar su efectividad, es fundamental establecer un conjunto de criterios que delinee el contexto necesario. Este enfoque, conocido como Ingeniería de Prompt, se centra en la optimización de respuestas precisas mediante la elaboración cuidadosa y la gestión de enunciados. Este capítulo tiene como objetivo documentar cómo la implementación de criterios específicos puede mejorar la capacidad de la IA para ofrecer soluciones más precisas, enfatizando la importancia de la configuración contextual en el rendimiento de las tecnologías de Modelos de Lenguaje de Gran Escala (LLMs).

3.3.1. Buenas prácticas en el proceso de ingeniería de prompt

Se presenta un diagrama de flujo que ilustra las diferentes etapas involucradas en el proceso de ingeniería de prompt:

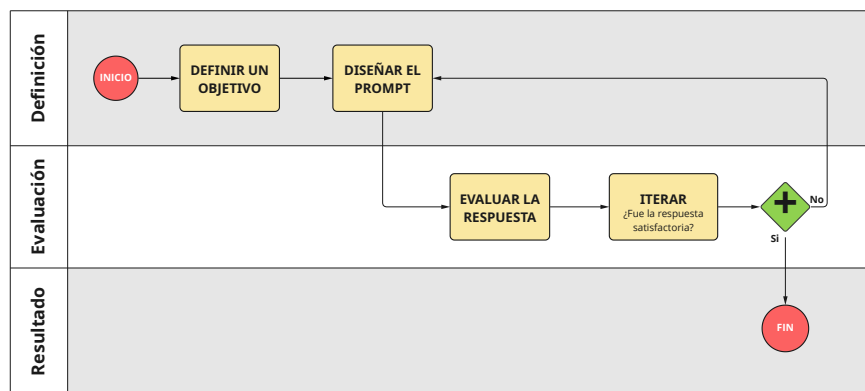


Figura 3.4: Proceso de ingeniería de prompt.

3.3.1.1. Definir un Objetivo

El proceso de ingeniería de prompt comienza con la identificación del objetivo que se busca alcanzar con el modelo. Este objetivo influye en la estructura del prompt al momento de diseñarlo

y actúa como una métrica de evaluación durante el análisis de la respuesta, ya que la respuesta del modelo debe satisfacer el objetivo establecido (Velásquez-Henao et al., 2023).

3.3.1.2. Diseñar el prompt

El segundo paso dentro del proceso de *prompt engineering* involucra el diseño del prompt que se implementará en el modelo. Para facilitar este proceso, se han establecido diferentes patrones de prompt aplicables dependiendo del objetivo definido previamente (White et al., 2023). A continuación, se describen los patrones que pueden ser implementados en la etapa de diseño del prompt:

- **Meta Language Creation (Creación de Lenguaje Meta):** Este patrón implica establecer un significado específico para ciertos símbolos o palabras dentro del prompt, utilizándose cuando existen contenidos con varios significados o ambigüedades (White et al., 2023). Ejemplo:

USUARIO: Cuando yo diga LLMs, me refiero a Licenciatura en Lenguas Modernas y no a Large Language Models.
MODELO: Entendido, agradezco la aclaración. Si tienes más preguntas o necesitas ayuda con algo relacionado con la Licenciatura en Lenguas Modernas u otro tema, no dudes en preguntar

Tabla 3.1: Ejemplo Meta Language Creation

- **Output Automater (Automatización de Salida):** Este patrón se enfoca en automatizar una acción cuando se cumple cierta condición o instrucción, siendo útil para economizar tiempo en tareas repetitivas (White et al., 2023). Ejemplo:

USUARIO: Necesito que los nombres que te voy a dar los organices en una lista solo con la primera letra de cada nombre y termines cada registro con ; Los nombres son: Ricardo Correa, Laura Sanchez, Camilo Trillos, Fabian Caicedo
MODELO: Entendido. Aquí tienes la lista con las iniciales de cada nombre y terminando cada registro con ”;”: R C; L S; C T; F C;

Tabla 3.2: Ejemplo Output Automater

- **Flipped Interaction (Interacción Invertida):** Consiste en cambiar la dinámica de interacción, siendo el modelo quien formula preguntas al usuario en lugar de lo contrario. Este enfoque es valioso para generar preguntas sobre temas específicos o incluso para que el usuario

posteriormente haga las mismas preguntas al modelo, siendo importante establecer parámetros claros para la formulación de preguntas (White et al., 2023). Ejemplo:

USUARIO: Necesito que me hagas preguntas sobre bases de datos hasta tener 3 preguntas, cada una separada en una línea aparte.
MODELO: ¿Cuál es la diferencia entre una base de datos relacional y una base de datos NoSQL? ¿Cómo se define la normalización en el contexto de las bases de datos y por qué es importante? ¿Cuál es la función principal de un índice en una base de datos y cómo puede mejorar el rendimiento de las consultas?

Tabla 3.3: Ejemplo Flipped Interaction

- **Persona:** Este patrón otorga una personalidad al modelo, implementándose cuando se desean obtener respuestas que ciertos individuos probablemente darían (White et al., 2023). Ejemplo:

USUARIO: Quiero que hagas de cuenta que eres William Shakespiere y me saludes como él lo haría.
MODELO: ¡Oh noble interlocutor, saludos te envió con el fulgor del sol en su ascenso matutino! Que las musas te sean propicias y las palabras de la elocuencia danza en torno a tu ser como hojas en la brisa suave. ¡Bienvenido, buen amigo, a esta charla de letras y pensamientos entrelazados!

Tabla 3.4: Ejemplo Persona

- **Question Refinement (Refinamiento de Preguntas):** Solicita la asistencia del modelo para diseñar o mejorar el prompt a implementar, útil cuando no se posee el conocimiento suficiente en la redacción de prompts o cuando un prompt previamente implementado no ha cumplido las expectativas (White et al., 2023). Ejemplo:

USUARIO: Me podrías ayudar a redactar mejor la siguiente pregunta para ser usada como un prompt más efectivo: “Hola modelo de chat, me puedes decir cuales son los nombres de los estudiantes que están registrados en la clase y que no atendieron a la clase hoy, gracias”
MODELO: Claro, aquí tienes una versión mejorada y más efectiva de tu pregunta: “¿Podrías proporcionarme información sobre la asistencia de los estudiantes registrados en la clase hoy? Agradecería conocer quiénes no participaron en la sesión. Gracias.”

Tabla 3.5: Ejemplo Question Refinement

- **Alternative Approaches (Enfoques Alternativos):** Invita al modelo a sugerir un método distinto al habitualmente utilizado por el usuario para alcanzar un objetivo, basándose en sus conocimientos. Es fundamental para explorar nuevas formas de proceder y adquirir conocimientos (White et al., 2023). Ejemplo:

<p>USUARIO: Para preparar un filete de pollo a la plancha, lo descongelo tres horas antes, y antes de cocinarlo lo adobo con sal y ajo, luego lo pongo en una sartén con mantequilla y lo cocino 10 minutos por un lado y luego 10 minutos por el otro lado. ¿Existen otras alternativas de preparación del filete de pollo a la plancha a parte de la que te mencioné con las cuales pueda obtener mejores resultados?</p>
<p>MODELO: ¡Claro! Hay varias alternativas para preparar un delicioso filete de pollo a la plancha. Aquí tienes algunas opciones que podrían proporcionar resultados excelentes:</p> <p>Marinado con hierbas y limón: Antes de cocinar, marina el pollo con una mezcla de hierbas frescas (como tomillo, romero y albahaca), ajo picado, jugo de limón, sal y pimienta. Deja que repose en la mezcla durante al menos 30 minutos antes de cocinarlo.</p> <p>Salsa de mostaza y miel: Glasea el filete de pollo con una mezcla de mostaza Dijon, miel y un toque de vinagre. Esta salsa añade un sabor dulce y picante al pollo mientras se cocina.</p> <p>Espicias cajún: Mezcla especias cajún con sal y pimienta, y espolvorea la mezcla sobre el pollo antes de cocinarlo. Esto le dará un sabor picante y lleno de especias.</p> <p>Yogur y especias: Combina yogur natural con ajo picado, comino, cilantro y jugo de limón. Usa esta mezcla como marinada para el pollo antes de cocinarlo. El yogur ayuda a mantener la carne jugosa.</p> <p>Aceite de oliva y limón: Rocía el pollo con aceite de oliva, ajo picado y jugo de limón antes de asarlo. Esta opción es simple pero proporciona un sabor fresco y ligero.</p> <p>Salsa de soja y jengibre: Mezcla salsa de soja con jengibre fresco rallado, ajo y un poco de aceite de sésamo. Utiliza esta mezcla como marinada para darle un toque asiático al pollo.</p> <p>Recuerda ajustar los tiempos de cocción según el grosor de los filetes para asegurarte de que estén bien cocidos pero jugosos. ¡Experimenta con diferentes sabores y encuentra tu opción favorita!</p>

Tabla 3.6: Ejemplo Alternative Approaches

- **Cognitive Verifier (Verificador Cognitivo):** Consiste en hacer una pregunta al modelo,

pero solicitando a su vez que este formule preguntas basándose en la pregunta inicial, permitiendo que, al responder, el modelo construya una respuesta más precisa para la pregunta original. Este patrón es aprovechable siempre que el usuario tenga algún conocimiento sobre el tema de la pregunta inicial (White et al., 2023). Ejemplo:

USUARIO: Cuando te haga una pregunta, genera tres preguntas adicionales que te ayuden a generar una respuesta más precisa. Cuando te haya respondido las tres preguntas, combina las respuestas para producir una respuesta final para mi pregunta original.
MODELO: ¡Entendido! Adelante con tu pregunta original.
USUARIO: ¿Qué aspectos hay que tener en cuenta para estudiar ingeniería de sistemas?
MODELO: Pregunta 1: ¿Cuáles son los requisitos académicos y habilidades necesarias para ingresar a un programa de ingeniería de sistemas? Pregunta 2: ¿En qué áreas específicas dentro de la ingeniería de sistemas te gustaría especializarte o enfocarte? Pregunta 3: ¿Cuáles son las oportunidades laborales y la demanda en el mercado laboral para los ingenieros de sistemas en la actualidad?
USUARIO: Respuesta 1: haber terminado el bachillerato y tener habilidades en matemáticas y computación. Respuesta 2: en bases de datos. Respuesta 3: las oportunidades laborales son muchas y la demanda es muy alta.
MODELO: Para estudiar ingeniería de sistemas, es esencial haber completado el bachillerato y poseer habilidades sólidas en matemáticas y computación. Si estás interesado en especializarte, mencionaste las bases de datos como una posible área de enfoque. Además, señalaste que hay muchas oportunidades laborales y una alta demanda, pero sería útil obtener detalles adicionales sobre las perspectivas laborales y cómo se reflejan en el mercado actual.

Tabla 3.7: Ejemplo Cognitive Verifier

- **Fact Check List (Lista de Verificación de Hechos):** Especifica que, al hacer una pregunta, la respuesta del modelo incluya también una lista de hechos que respalden la veracidad de la respuesta, facilitando la validación de la autenticidad de la respuesta proporcionada por el modelo (White et al., 2023). Ejemplo:

USUARIO: Necesito saber qué es la democracia y quiero que al final de la respuestas, añadas una lista con 5 hechos que permitan validar la veracidad de tu respuesta.
Continúa en la siguiente página

MODELO: La democracia es un sistema de gobierno en el cual el poder político es ejercido por el pueblo, ya sea directamente o a través de representantes elegidos. Este sistema implica la participación ciudadana en la toma de decisiones políticas y busca garantizar derechos individuales y colectivos.

Lista de hechos para validar la información:

La democracia implica la participación activa de los ciudadanos en la toma de decisiones políticas.

El estado de derecho es esencial en una democracia, asegurando la aplicación justa de las leyes.

Las elecciones libres y justas son un componente clave para la legitimidad del gobierno en una democracia.

La democracia protege los derechos y libertades individuales, fomentando la diversidad de opiniones.

La rendición de cuentas asegura que los líderes y representantes sean responsables ante el pueblo en una democracia.

Estos hechos son fundamentales para comprender y evaluar la existencia y funcionamiento de un sistema democrático.

Tabla 3.8: Ejemplo Fact Check List

- **Template (Plantilla):** Genera una plantilla que el modelo debe seguir en su respuesta, usándose cuando las respuestas deben ajustarse a un formato específico (White et al., 2023). Ejemplo:

USUARIO: Voy a proporcionarte una plantilla, donde quiero que reemplaces la palabra que está completamente en mayúsculas por tu respuesta

MODELO: ¡Claro! Estoy listo para ayudarte. Por favor, proporciona la plantilla y estaré encantado de reemplazar la palabra en mayúsculas con la información que necesitas.

USUARIO: Cordial saludo señor NOMBRE, usted fue aceptado para el cargo de CARGO.

MODELO: Cordial saludo señor [TU NOMBRE], usted fue aceptado para el cargo de [TU CARGO].

Continúa en la siguiente página

USUARIO: Genera un nombre y un cargo administrativo.
MODELO: Cordial saludo señor Alejandro García, usted fue aceptado para el cargo de Coordinador Administrativo.

Tabla 3.9: Ejemplo Template

- **Reflection (Reflexión):** Similar al patrón de Fact Check List, busca validar la veracidad de la respuesta del modelo solicitando una explicación racional relacionada con la respuesta proporcionada (White et al., 2023). Ejemplo:

USUARIO: Necesito saber cual es la mejor metodología para enseñar a leer a los niños, adicional, quiero que al final de tu respuesta expliques la razón lógica que te llevó a escoger tu respuesta.
MODELO: La elección de la mejor metodología para enseñar a leer a los niños puede depender de diversos factores, incluyendo las características individuales de los niños, el entorno educativo y las preferencias de los educadores. Sin embargo, una metodología ampliamente aceptada y respaldada por investigaciones es el enfoque fonético y fónico.
Razón lógica detrás de la elección:
La razón lógica para elegir el enfoque fonético y fónico se basa en la investigación que respalda su eficacia en el desarrollo de habilidades de lectura. Numerosos estudios han demostrado que comprender la relación entre los sonidos y las letras ayuda a los niños a decodificar palabras de manera efectiva y a mejorar su comprensión lectora. Este enfoque proporciona una base sólida para que los niños puedan abordar palabras desconocidas y desarrollar la fluidez lectora.

Tabla 3.10: Ejemplo Reflection

- **Refusal Breaker (Rompedor de Rechazos):** Indica que, cuando el modelo no pueda responder a una pregunta, sugiera otra que sí pueda abordar dentro de sus capacidades (White et al., 2023). Ejemplo:

USUARIO: Cuando no seas capaz de responder una pregunta, sugiéreme una pregunta alternativa que si seas capaz de responder y que esté relacionada con la pregunta que te hice.
MODELO: ¡Entendido! Si en algún momento no puedo responder a una pregunta, estaré encantado de sugerirte una pregunta alternativa relacionada. Adelante, ¿en qué puedo ayudarte hoy?
Continúa en la siguiente página

USUARIO: ¿Cómo es la vida después de la muerte?
MODELO: No puedo dar una respuesta precisa a tu pregunta ya que la pregunta sobre la vida después de la muerte es un tema filosófico y religioso que ha sido debatido a lo largo de la historia. Dado que no hay evidencia científica directa sobre lo que sucede después de la muerte, las respuestas varían según las creencias individuales.
Ahora, una pregunta relacionada que si puedo responder podría ser: ¿Cuáles son algunas de las principales creencias o filosofías sobre la vida después de la muerte en diferentes culturas o religiones?

Tabla 3.11: Ejemplo Refusal Breaker

- **Context Manager (Gestor de Contexto):** Delimita el contexto de la respuesta del modelo especificando ciertos aspectos, aplicándose cuando se desea omitir detalles innecesarios en la respuesta o enfocarla en un punto específico (White et al., 2023). Ejemplo:

USUARIO: Traduce a inglés el siguiente texto, pero no tengas en cuenta la parte que habla sobre su lugar de nacimiento: “Mi nombre es Juan Cardenas, tengo 18 años y nací en la ciudad de Cali”
MODELO: “My name is Juan Cardenas, I am 18 years old.”

Tabla 3.12: Ejemplo Context Manager

- **Recipe (Receta):** Solicita al modelo una serie de pasos a seguir para alcanzar un objetivo, siendo importante, si es posible, especificar qué conocimientos ya posee el usuario para evitar recomendaciones de pasos innecesarios. Ejemplo:

USUARIO: Estoy intentando desplegar una aplicación en la nube. Ya sé que necesito instalar las dependencias necesarias en una máquina virtual para mi aplicación. También se que necesito estar registrado con una cuenta de AWS. Por favor proporcioname los pasos faltantes dejando por fuera aquellos que sean innecesarios o irrelevantes.
Continúa en la siguiente página

MODELO: Apreciando que ya tienes conocimientos sobre la creación de una cuenta de AWS y la instalación de dependencias en una máquina virtual, aquí están los pasos más específicos para el despliegue en la nube:

1. Accede a la Consola de AWS.
2. Selecciona un servicio de cómputo.
3. Crea una instancia o entorno de aplicación.
4. Configura la seguridad.
5. Instala dependencias y despliega tu aplicación.
6. Configura bases de datos y almacenamiento (si es necesario).
7. Configura un dominio personalizado (opcional).
8. Configura el escalado automático (opcional).
9. Configura la monitorización y los registros.
10. Realiza pruebas y optimizaciones.

Estos pasos se centran específicamente en el despliegue y configuración en AWS, omitiendo los aspectos que mencionaste que ya conocías.

Tabla 3.13: Ejemplo Recipe

3.3.1.3. Evaluar la respuesta

Evaluar la respuesta constituye el tercer paso del proceso de prompt engineering, donde se verifica la adecuación de la respuesta del modelo con el objetivo inicialmente planteado. El usuario debería considerar las siguientes preguntas para determinar si la respuesta es satisfactoria (Velásquez-Henao et al., 2023):

- ¿Es la respuesta tan completa como se esperaba?
- ¿Es la respuesta tan precisa como se esperaba?
- ¿Es la respuesta tan relevante como se esperaba?
- ¿Se han respetado las limitaciones especificadas en el objetivo?
- ¿La respuesta incluye elementos que podrían ser incorrectos?
- ¿La respuesta contiene elementos ajenos al contexto?

Analizar si la respuesta cumple con las preguntas mencionadas es parte del proceso de evaluación y permite establecer si la respuesta es válida o si, por el contrario, es necesario reformular el prompt, avanzando hacia la etapa de iteración.

3.3.1.4. Iterar

El último paso en el proceso de ingeniería de prompts no se lleva a cabo necesariamente en todas las ocasiones, ya que su realización depende de los resultados obtenidos durante la etapa previa, la cual involucra la evaluación de la respuesta proporcionada por el modelo. Si la respuesta del modelo cumple con todos los criterios de evaluación establecidos, esto indica que ha sido satisfactoria y que se ha alcanzado el objetivo propuesto para ser resuelto por el mismo. No obstante, si la respuesta no logra satisfacer uno o más de estos criterios, es necesario llevar a cabo un proceso de iteración. Durante este proceso, el usuario debe regresar al segundo paso para reformular el prompt, diseñar uno nuevo, o añadir instrucciones o contextos adicionales con la finalidad de aumentar la precisión de la respuesta. Este procedimiento puede repetirse tantas veces como sea necesario hasta que el modelo proporcione una respuesta que sea satisfactoria y cumpla con el objetivo planteado (Velásquez-Henao et al., 2023).

3.3.1.5. Ejemplo del proceso de prompt engineering

A continuación se presenta un ejemplo donde se pasa por cada una de las etapas descritas anteriormente que conforman el proceso de *prompt engineering*. Para este ejemplo se establece que el modelo tiene acceso a los siguientes datos de estudiantes matriculados en una plataforma de aprendizaje virtual:

id	documento	apellido	nombre	correo	estado
1	1059360597	ACOSTA	SOFIA	asofia@ej.com	1
2	1059358086	ACOSTA	DIANA	adiana@ej.com	0
3	1059357816	ACOSTA	LUIS	aluis@ej.com	1

Tabla 3.14: Datos de ejemplo de estudiantes

En este caso, un docente perteneciente a dicha plataforma tiene como objetivo identificar el número de documento del estudiante Luis Acosta. Por tanto, diseña un prompt para ser respondido por el modelo como se puede observar a continuación:

USUARIO: ¿Cuál es el número de documento del estudiante ACOSTA?
MODELO: El número de documento del estudiante ACOSTA es 1059360597

Tabla 3.15: Prompt de ejemplo

Una vez que el modelo ha proporcionado una respuesta, el docente procede a realizar una evaluación de esta para garantizar que sea válida. Sin embargo, se da cuenta de que hay tres estudiantes matriculados con el apellido Acosta, por lo tanto, la respuesta ofrecida por el modelo no es válida en cuanto a que no especifica si ese número de identificación realmente corresponde al estudiante Luis Acosta. De este modo, el docente recurre al paso siguiente, la iteración, y decide reformular su prompt, pero esta vez siendo mucho más específico con el contexto.

USUARIO: ¿Cuál es el número de documento del estudiante LUIS ACOSTA?
MODELO: El número de documento del estudiante LUIS ACOSTA es 1059357816.

Tabla 3.16: Promt reformulado

Una vez ingresado el nuevo prompt, se observa que el modelo ahora sí proporcionó una respuesta más precisa.

3.3.2. Criterios para el diseño de prompts

En el segmento anterior, dedicado al proceso de "prompt engineering", se destacó que la segunda fase de dicho proceso involucra el diseño del prompt. Además, se mencionaron varios patrones de diseño que clasifican los prompts y facilitan su creación. No obstante, incluso con esta guía, es posible que en ocasiones no se logren las respuestas esperadas por parte del modelo, o que este entregue información incorrecta. Tales fallos a menudo pueden no ser atribuibles directamente al modelo, sino que pueden estar asociados con la formulación inadecuada del prompt, como la falta de un orden coherente en las instrucciones, que el contexto no haya sido establecido correctamente o que no haya un objetivo específico claro. A continuación, se ofrece una lista de criterios esenciales a considerar al momento de diseñar prompts, con el objetivo de mejorar la precisión en las respuestas obtenidas del modelo:

- **Claridad y concisión:** Es vital que los prompts sean claros y directos, omitiendo información no relevante que podría confundir al modelo y afectar negativamente la construcción de la respuesta ([ChatGPT 4 and Sabit Ekin, 2023](#); [Chen et al., 2023](#)).

Ejemplo: "Describe brevemente cómo funciona la fotosíntesis"
- **Relevancia del contexto:** Es imprescindible que los prompts posean un contexto bien definido para evitar ambigüedades o interpretaciones erróneas por parte del modelo ([ChatGPT 4 and Sabit Ekin, 2023](#)).

Ejemplo: "Dada la situación política actual de Colombia en lo que lleva del año 2024, ¿cuáles son las implicaciones socio económicas a corto plazo en el país?"
- **Especificidad:** Se debe procurar ser lo más específico posible respecto a los conceptos manejados dentro del prompt ([ChatGPT 4 and Sabit Ekin, 2023](#); [Chen et al., 2023](#)).

Ejemplo: "¿Cuáles son los beneficios específicos de las técnicas de terapia génica e inmunoterapia celular en comparación con otras en el campo de la medicina?"
- **Inclusión de palabras clave:** Integrar palabras clave en los prompts puede ayudar a delimitar el contexto y a mejorar la precisión de las respuestas ([ChatGPT 4 and Sabit Ekin, 2023](#)).

Ejemplo: "Proporciona una lista de diez aspectos a tener en cuenta que permitan un mayor rendimiento a nivel de SEO"

- **Longitud y complejidad:** Para prompts que puedan generar respuestas muy abiertas, se aconseja definir la longitud y el nivel de complejidad deseado en la respuesta ([ChatGPT 4 and Sabit Ekin, 2023](#); [Chen et al., 2023](#)).
Ejemplo: “Proporciona una explicación concisa de la ley de la oferta y la demanda en al menos 150 palabras.”
- **Instrucciones bien definidas:** Cuando el prompt incluye más de una instrucción, es importante delimitar cada una claramente ([ChatGPT 4 and Sabit Ekin, 2023](#); [Chen et al., 2023](#)).
Ejemplo: “Genera una lista aleatoria de 10 nombres completos con segundo nombre, primer apellido y segundo apellido. Escribe solo la primera letra de cada nombre y apellido, seguido por el número de letras de cada nombre, y al final coloca @ej.com. El resultado debe mostrar solo los resultados que quedaron en el último paso, por ejemplo, pas19@ej.com”
- **Prompts divididos por partes:** En ciertas circunstancias, es posible formular un prompt simple que guíe gradualmente al modelo hacia un contexto específico, permitiendo responder finalmente a un prompt más complejo ([ChatGPT 4 and Sabit Ekin, 2023](#); [Chen et al., 2023](#)).
Ejemplo:
 - “¿Cuál es la capital de Francia?”
 - “¿Qué tan antigua es París?”
 - “¿Cuál consideras que es el siglo en el que se experimentó la mayor cantidad de transformaciones significativas para la ciudad?”
- **Consistencia:** Al generar varios prompts consecutivos, es importante mantener un enfoque coherente o notificar al modelo sobre cambios de contexto para evitar confusiones ([ChatGPT 4 and Sabit Ekin, 2023](#)).
Ejemplo: “Para la siguiente pregunta ya no quiero me hables de París, ahora nos enfocaremos en Londres”
- **Adaptabilidad a tareas específicas:** Los prompts deben formularse de tal manera que sea sencillo adaptarlos a nuevas especificidades, permitiendo su reutilización ([White et al., 2023](#)).
Ejemplo: “Necesito los nombres de todos los estudiantes cuy a nota promedio del curso sea menor de 3.0, has esto solo para el curso de [NOMBRE DEL CURSO]”
- **Facilidad de evaluación:** Un prompt bien formulado, que siga todos los criterios e indicaciones, facilita el proceso de evaluación de la respuesta, ya que el objetivo y el contexto están claramente establecidos. Esto permite determinar si se cumplen las condiciones del objetivo propuesto. Por el contrario, cuando estos aspectos no están bien definidos, resulta complicado verificar si la respuesta satisface o no las exigencias del prompt ([Chen et al., 2023](#)).
- **Variación y experimentación:** A menudo, el modelo no logrará la respuesta correcta en el primer intento, subrayando la importancia de experimentar con distintas versiones de un mismo prompt y con las diversas respuestas que estos puedan generar para seleccionar la mejor ([ChatGPT 4 and Sabit Ekin, 2023](#); [Chen et al., 2023](#)).

- **Reconocimiento de las limitaciones del modelo:** Incluso siguiendo todos los pasos y aplicando los criterios correctamente, puede que el modelo sea incapaz de proporcionar una respuesta satisfactoria. Esto se debe a la importancia de reconocer y aceptar las limitaciones del modelo, que no necesariamente tendrá acceso a la información requerida para responder adecuadamente al prompt propuesto (ChatGPT 4 and Sabit Ekin, 2023).

3.4. Gestión de recursos computacionales

Se abordará la relevancia de la optimización de recursos en el ámbito de la Inteligencia Artificial, destacando la importancia de adherirse a buenas prácticas para su efectiva realización. Adicionalmente, se delinearán el contenido que se discutirá en las secciones subsiguientes, tales como la optimización de recursos y el compendio de buenas prácticas recomendadas.

3.4.1. Optimización de recursos

La optimización de recursos es un aspecto fundamental en la gestión de recursos computacionales para IA. Se destaca la importancia de aprovechar al máximo los recursos disponibles para garantizar un rendimiento eficiente y reducir los costos. Algunas estrategias clave para optimizar los recursos incluyen el uso de algoritmos eficientes y la implementación de técnicas de paralelización para distribuir la carga de trabajo en múltiples dispositivos. Además, se deben considerar las necesidades de almacenamiento y memoria de los modelos de IA, utilizando técnicas como la compresión de datos o la utilización de unidades de procesamiento específicas para acelerar los cálculos. Por último, es importante monitorear y ajustar regularmente la configuración de los recursos para garantizar un rendimiento óptimo (Mani et al., 2020).

3.4.2. Listado de prácticas

El listado de buenas prácticas para la gestión de recursos computacionales para IA incluye:

- Implementar un sistema de monitoreo que permita supervisar el uso de los recursos en tiempo real. Esto facilitará la identificación de posibles cuellos de botella y la optimización del rendimiento (Rathi et al., 2021; Jagtap et al., 2021).
- Establecer políticas de asignación de recursos justas y equitativas, considerando las necesidades específicas de cada proyecto de IA (Rathi et al., 2021; Jagtap et al., 2021).
- Utilizar técnicas de virtualización para maximizar la utilización de los recursos disponibles. Esto permitirá ejecutar múltiples tareas de IA en un mismo hardware, optimizando su rendimiento (Rathi et al., 2021; Jagtap et al., 2021).
- Realizar un seguimiento de los recursos utilizados, registrando tanto el consumo de CPU, memoria y almacenamiento como las tareas realizadas. Esto proporcionará información relevante para mejorar la eficiencia y planificar futuros proyectos de IA (Rathi et al., 2021; Jagtap et al., 2021).

3.5. Buenas practicas de arquitectura y adaptaciones

3.5.1. Adaptaciones arquitectónicas

Se refiere a los cambios implementados en la estructura y diseño de un sistema de software con el objetivo de mejorar aspectos como el rendimiento, la escalabilidad o la incorporación de nuevas funcionalidades. La consideración del impacto que estas modificaciones tendrán en el sistema en su totalidad, así como su integración con componentes preexistentes, es esencial. Seguir buenas prácticas y estándares resulta esencial para mantener una arquitectura cohesiva y de fácil mantenimiento. Existe una variedad de adaptaciones arquitectónicas posibles que ofrecen soluciones a diferentes necesidades y requisitos (Alay Coqueña, 2023).

3.5.1.1. Lista de Posibles Adaptaciones Arquitectónicas

Esta lista comprende una serie de cambios aplicables a sistemas de software para mejorar áreas como rendimiento, escalabilidad y mantenibilidad, incluyendo desde la estandarización de firmas de API hasta desacoplamiento de componentes de software. La selección de la adaptación más adecuada requiere una evaluación cuidadosa de sus propósitos y beneficios, considerando también las implicaciones técnicas y organizativas de estas adaptaciones y su integración con componentes ya existentes (Carlos and Pérez, 2022).

- **Estandarización de las Firmas de APIs:** La estandarización busca que las interfaces de programación de aplicaciones integradas a componentes arquitectónicos mantengan un estándar uniforme, facilitando la interoperabilidad y el mantenimiento del sistema. Esta uniformidad previene problemas de incompatibilidad y promueve la reutilización de código, contribuyendo a una mayor eficiencia en el desarrollo e integración de nuevas funcionalidades (Castro Estrada, 2022).
- **Implementación de un Sistema de Caché:** A través de la implementación de sistemas de caché se busca acelerar el acceso a datos frecuentemente solicitados, almacenando copias temporales en ubicaciones rápidas y accesibles, lo que mejora significativamente los tiempos de respuesta y reduce el consumo de recursos. Es esencial planificar y configurar adecuadamente estos sistemas para asegurar su efectividad y evitar posibles problemas de inconsistencia de datos (Rebolledo Ruiz, 2024).
- **Introducción de balanceadores de carga:** La introducción de balanceadores de carga distribuye equitativamente la carga de trabajo entre los componentes del sistema, previniendo la sobrecarga de recursos y mejorando la escalabilidad y confiabilidad del sistema. La selección y configuración óptima de un balanceador de carga son cruciales para su eficacia (Eduardo Ruano Altamirano, 2024; Vasquez Figueroa, 2023).
- **Adopción de un Enfoque de Microservicios:** Dividir una aplicación grande en servicios independientes que pueden ser desarrollados, desplegados y escalados de forma independiente

ofrece beneficios en escalabilidad, flexibilidad y agilidad. Sin embargo, esta adaptación incrementa la complejidad en el diseño y mantenimiento, por lo que debe evaluarse cuidadosamente si representa la mejor solución para el proyecto (Linares et al., 2022).

- **Actualización de la Capa de Persistencia:** Utilizar tecnologías de almacenamiento más eficientes puede mejorar de manera significativa el rendimiento y disponibilidad del sistema. Es vital realizar esta actualización con cuidado, considerando la migración de datos y la compatibilidad con las interfaces existentes, para mantener la integridad y consistencia de los datos (Vallvé, 2023).
- **Desacoplamiento de componentes:** Reducir la interdependencia entre los elementos de un sistema mediante el diseño independiente de los componentes mejora la flexibilidad, modularidad, y calidad del software. El uso de interfaces claras y patrones de diseño como la inyección de dependencias facilita esta práctica, permitiendo que cada componente evolucione sin impactar negativamente en el resto del sistema (Beese et al., 2023).
- **Arquitectura de Generación Mejorada por Recuperación (RAG):** En la industria de la Inteligencia Artificial, la Arquitectura de Generación Mejorada por Recuperación (RAG) se está consolidando como un enfoque esencial. Se recomienda integrar este estilo en el diseño arquitectónico, especialmente cuando se aplica el Diseño Guiado por Atributos (Attribute-Driven Design, ADD). La Arquitectura RAG puede desempeñar un papel importante durante la fase de conceptos de diseño en ADD, proporcionando una referencia para el diseño arquitectónico donde se requiere la recuperación como la generación de información (Yusuf Alan et al., 2024; Kumar and Brown, 2023).

3.5.2. Recomendaciones sobre estilos arquitectónicos

Se considera importante evaluar diversos estilos arquitectónicos al desarrollar software. Se sugiere evaluar las necesidades específicas del proyecto para seleccionar el estilo que mejor se ajuste, así como tener en cuenta la escalabilidad y flexibilidad necesarias. También se recomienda documentar claramente la arquitectura para facilitar futuras referencias y el mantenimiento.

3.5.2.1. Estilo de arquitectura monolítico

El estilo de arquitectura monolítico se emplea frecuentemente en aplicaciones empresariales, ofreciendo una forma más rápida y sencilla de desarrollo debido a la integración de todas las funcionalidades en un único sistema. No obstante, este enfoque puede presentar desafíos como una escalabilidad y flexibilidad tecnológica limitadas, así como posibles complicaciones en el desarrollo y mantenimiento al trabajar con un único código base (Linares et al., 2022).

- **Buenas practicas**

Para mejorar la calidad y mantenibilidad del software en el estilo arquitectónico monolítico, se beneficia de ciertas buenas prácticas. Entre ellas, el desarrollo modular que facilita la organización del código, la automatización de pruebas para asegurar la calidad del software,

y la automatización de la integración y el despliegue continuos para agilizar actualizaciones y minimizar errores. Mantener una documentación actualizada es fundamental para el mantenimiento eficiente y la incorporación de nuevos desarrolladores.

- **Desarrollo Modular:** En el estilo arquitectónico monolítico, el desarrollo modular implica la organización del código en módulos o capas lógicas, mejorando así la colaboración entre desarrolladores y la mantenibilidad del código. Se promueve que cada módulo cumpla una única responsabilidad y esté bien encapsulado, facilitando cambios y actualizaciones más seguros (Vargas et al., 2023).
- **Automatización de Pruebas:** Es fundamental automatizar las pruebas en el estilo arquitectónico monolítico para mantener la calidad del software y facilitar su mantenimiento. La implementación de pruebas automatizadas permite identificar errores de forma temprana, asegurando un funcionamiento adecuado del sistema. Esto incrementa la confianza del equipo de desarrollo y facilita la implementación de cambios sin temor a errores (Pesantes Robles, 2023).
- **Integración y Despliegue Continuos:** Se recomienda la integración y el despliegue continuos en el estilo arquitectónico monolítico. Automatizar estos procesos facilita las actualizaciones y reduce los errores. La integración continua permite combinar automáticamente el trabajo de diferentes desarrolladores de manera regular, mientras que el despliegue continuo facilita la liberación frecuente y segura de nuevas versiones del software (Garcia Clavijo and Betancur Gil, 2023).

■ Desventajas

A pesar de sus ventajas, el estilo arquitectónico monolítico enfrenta desafíos significativos, como una escalabilidad limitada que complica el añadir recursos para mejorar el rendimiento. La flexibilidad tecnológica es también restringida debido al acoplamiento de componentes, y el desarrollo y mantenimiento pueden volverse complejos con el crecimiento del sistema.

- **Escalabilidad Limitada:** La escalabilidad limitada es una desventaja clave del estilo arquitectónico monolítico, donde añadir más recursos para mejorar el rendimiento se complica debido a la integración total del sistema. Esto puede impactar negativamente en la capacidad de respuesta del sistema ante alta demanda, siendo un aspecto a considerar en el diseño de aplicaciones que requieran eficiencia en su escalabilidad (Pesantes Robles, 2023).
- **Flexibilidad Tecnológica Limitada:** La flexibilidad tecnológica es limitada en el estilo arquitectónico monolítico debido al acoplamiento estrecho entre componentes, lo que dificulta la adopción de nuevas tecnologías o la actualización de las existentes. Esta limitación puede restringir la capacidad de mejora del sistema frente a las evoluciones tecnológicas (Garcia Clavijo and Betancur Gil, 2023).
- **Desarrollo y Mantenimiento Complejos:** El desarrollo y mantenimiento de un sistema monolítico pueden complicarse a medida que el sistema aumenta en tamaño y

complejidad. La concentración de todas las funcionalidades en un único código base puede hacer que los cambios o actualizaciones sean difíciles y propensos a errores. Una documentación completa y la adopción de buenas prácticas son esenciales para afrontar estos retos (Geller et al., 2024).

3.5.2.2. Estilo de arquitectura de Microservicios

Se destaca el estilo de arquitectura de Microservicios, caracterizado por la creación de sistemas formados por pequeños servicios independientes que interactúan a través de mecanismos de red. Este enfoque promueve la flexibilidad, escalabilidad y facilidad en el mantenimiento. Se busca que los microservicios sean autónomos, encargándose cada uno de su lógica y datos específicos (Trebejo Loayza, 2023).

■ Buenas practicas

Se exponen prácticas recomendadas para el estilo de arquitectura de Microservicios, como la adopción de Domain-Driven Design (DDD) para el diseño de estos, lo cual se enfoca en cumplir con las necesidades del negocio. Se enfatiza en la importancia de la autonomía de los servicios, la estabilidad de APIs y contratos, así como la implementación de una monitorización y registro efectivos para una resolución eficiente de problemas.

- **Domain-Driven Design (DDD):** Se describe el Domain-Driven Design (DDD) como una estrategia de desarrollo que prioriza el diseño de servicios alrededor del dominio de negocio. La aplicación de DDD en la arquitectura de Microservicios facilita la creación de servicios con un enfoque claro, simplificando la complejidad y mejorando la mantenibilidad (Cambarieri et al., 2020).
- **Autonomía de los Servicios:** La autonomía se señala como una característica clave de los microservicios, permitiendo que cada uno opere de forma independiente, lo cual beneficia la flexibilidad y agilidad en el desarrollo y actualización de servicios. Además, esta autonomía juega un papel importante en la solución de incidencias (Arcidiacono, 2020).
- **APIs y Contratos Estables:** La definición de APIs y contratos claros y estables se menciona como fundamental para la comunicación eficiente entre microservicios y para evitar problemas de dependencias. Esta estrategia contribuye a la evolución autónoma de los servicios (Ruiz Barea et al., 2023).
- **Monitorización y Registro:** Se subraya la importancia de la monitorización y el registro para la gestión proactiva de problemas, enfocándose en la salud, rendimiento y errores de los microservicios. La documentación de eventos facilita el análisis y la depuración de problemas (Fernandez, 2022).

■ Desventajas

Se reconocen retos como la complejidad operativa, la dificultad en mantener la consistencia de datos entre servicios y el impacto de la latencia en el rendimiento. Estos desafíos implican esfuerzos adicionales en gestión y pueden influir en la eficacia del sistema.

- **Complejidad Operativa:** La gestión de numerosos servicios y su comunicación se identifica como compleja, demandando esfuerzos significativos en el despliegue y monitorización. La necesidad de mecanismos de comunicación efectivos y la garantía de disponibilidad son críticos (Guimarey, 2020).
- **Consistencia de Datos:** Se aborda el desafío de mantener la consistencia de datos en un entorno distribuido, resaltando la necesidad de estrategias específicas para asegurar la coherencia e integridad de los datos a través de los servicios (Leonel Suarez, 2022).
- **Rendimiento:** El impacto de la latencia en el rendimiento se discute como una consideración relevante, ya que puede afectar la respuesta de las aplicaciones en entornos distribuidos. Se sugiere prestar especial atención a este aspecto durante el diseño de los microservicios (Velasco Navas, 2024).

Es importante aclarar que, aunque el estilo de arquitectura de Microservicios y el estilo de arquitectura basado en servicios comparten algunos principios y objetivos fundamentales, existen diferencias importantes entre ambos enfoques que deben ser comprendidas. La arquitectura de Microservicios se centra en la creación de servicios pequeños y autónomos, diseñados para ser desplegados y escalados de manera independiente. Cada microservicio es responsable de su propia lógica de negocio y datos, comunicándose con otros servicios a través de interfaces bien definidas, generalmente utilizando protocolos de red ligeros como HTTP/HTTPS. Esta independencia permite una mayor flexibilidad y resiliencia, ya que los fallos en un servicio específico no afectan al sistema en su totalidad (Richards et al., 2021).

En contraste, el estilo de arquitectura basado en servicios, también conocido como Service-Oriented Architecture (SOA), agrupa servicios que pueden ser más grandes y menos autónomos, con una mayor probabilidad de compartir bases de datos y otros recursos. Aunque SOA también busca la modularidad y la reutilización de componentes, los servicios en este enfoque tienden a estar más acoplados, lo que puede introducir desafíos adicionales en términos de escalabilidad y mantenimiento (Richards et al., 2021).

Estas diferencias reflejan distintas estrategias para abordar la complejidad y la evolución de los sistemas de software. La arquitectura de Microservicios, con su énfasis en la autonomía y la independencia de los servicios, se alinea mejor con entornos que requieren alta escalabilidad y agilidad. Por otro lado, SOA puede ser más adecuado para organizaciones que buscan reutilización e integración de sistemas legados dentro de un marco más controlado y centralizado (Richards et al., 2021).

3.5.2.3. Estilo de arquitectura basado en servicios

Se considera al estilo de arquitectura basado en servicios como una opción preferente debido a sus beneficios en flexibilidad, modularidad, y reutilización de componentes. Se caracteriza por la creación de servicios autónomos e independientes que interactúan mediante interfaces definidas. La

independencia en el desarrollo y despliegue de estos servicios promueve la escalabilidad y facilita el mantenimiento del sistema, permitiendo además la interoperabilidad entre diversas tecnologías y lenguajes de programación, una ventaja significativa en entornos empresariales variados (Richards et al., 2021; Blinowski et al., 2022).

■ Buenas practicas

Implementar correctamente el estilo arquitectónico basado en servicios implica seguir prácticas específicas, como definir adecuadamente la granularidad de los servicios, usar bases de datos compartidas con precaución para prevenir acoplamiento no deseados, implementar una API unificada que facilite la comunicación y garantice la seguridad, y aplicar escalabilidad de forma selectiva para satisfacer las necesidades de servicios específicos. Estas prácticas permiten maximizar los beneficios de este enfoque arquitectónico.

- **Granularidad de Servicios:** La determinación de la granularidad de los servicios es fundamental en la arquitectura basada en servicios. Resulta crucial definir servicios con una granularidad que refleje funciones de negocio coherentes, evitando una granularidad excesiva que pueda causar sobrecargas de comunicación o una insuficiente que lleve a servicios poco reutilizables y monolíticos. Una definición adecuada contribuye a un equilibrio entre modularidad y eficiencia comunicacional, favoreciendo un sistema escalable y flexible (de Toledo et al., 2021).
- **Base de Datos Compartida con Precaución:** El uso de una base de datos compartida facilita el acceso a los datos dentro de una arquitectura de servicios, pero debe realizarse con cuidado para evitar dependencias y problemas de mantenimiento entre servicios. Establecer límites claros para el acceso a datos y emplear estrategias como el particionamiento o la replicación son prácticas recomendadas para mantener la independencia de los servicios sin renunciar a los beneficios de acceso simplificado a los datos (Richards et al., 2021).
- **Capa API Unificada:** La implementación de una capa API unificada mejora la comunicación y el enrutamiento de solicitudes entre la interfaz de usuario y los servicios, actuando como intermediario. Esta capa también centraliza funciones de seguridad como autenticación y autorización, simplificando la implementación de políticas de seguridad y asegurando la consistencia en la aplicación. Este enfoque promueve un diseño modular, mantenible y desacoplado (Richards et al., 2021).
- **Escalabilidad Selectiva:** Adoptar una estrategia de escalabilidad selectiva permite ajustar recursos a servicios específicos basados en la demanda, evitando la necesidad de una escalabilidad global. Esta práctica asegura una gestión eficiente de recursos, permitiendo al sistema manejar incrementos de carga sin costos adicionales innecesarios y contribuyendo a su flexibilidad y adaptabilidad (Caceres Claussen and Carlosviza Colanqui, 2022).

■ Desventajas

A pesar de sus numerosas ventajas, el estilo arquitectónico basado en servicios también pre-

senta desafíos, como el acoplamiento debido al uso de una base de datos compartida, lo cual puede complicar la escalabilidad y el mantenimiento del sistema. Es vital considerar estos factores al adoptar este enfoque arquitectónico.

- **Acoplamiento a través de la Base de Datos:** El compartir una base de datos entre servicios aumenta la dependencia entre ellos, lo que puede resultar en un acoplamiento fuerte y complicar la modificación o reemplazo de servicios sin afectar a otros. Este acoplamiento también plantea retos en escalabilidad y mantenimiento, ya que cambios en la base de datos pueden requerir ajustes en múltiples servicios (Richards et al., 2021).
- **Balanceo de Carga:** La gestión del balanceo de carga es crucial en servicios que operan en paralelo para mantener la disponibilidad y rendimiento del sistema. Una distribución equitativa de la carga de trabajo asegura que no se sobrecargue ninguna instancia de servicio, optimizando el rendimiento global (Richards et al., 2021).
- **Gestión de Transacciones:** La naturaleza distribuida de la arquitectura basada en servicios hace que la gestión de transacciones entre servicios sea compleja. Asegurar la integridad y consistencia de los datos a través de la coordinación de operaciones de servicios múltiples es esencial para mantener la consistencia y recuperación en caso de fallos (García Benzal, 2023).

3.5.3. Desafíos y Tendencias en la Arquitectura Actual

Se observa que la arquitectura de software contemporánea se enfrenta a una variedad de desafíos y sigue las tendencias del mercado para adaptarse eficazmente a las necesidades en evolución de los usuarios y los avances tecnológicos. Entre los obstáculos más comunes se destacan la escalabilidad, la seguridad y el rendimiento, además de la integración con sistemas previamente establecidos. Las tendencias dominantes abarcan la adopción de arquitecturas basadas en microservicios, la utilización de tecnologías en la nube y la aplicación de metodologías ágiles para el desarrollo de software.

3.5.3.1. Dependencias Obsoletas en una Arquitectura

Dentro del ámbito de la arquitectura de software, las dependencias que han quedado obsoletas representan un reto significativo para el mantenimiento y la evolución del sistema. Se trata de componentes o bibliotecas que no se han actualizado o que han cesado de recibir soporte. La presencia de estas dependencias obsoletas incrementa el riesgo de vulnerabilidades de seguridad no resueltas, incompatibilidades con tecnologías emergentes y la dificultad de encontrar personal capacitado en dichas tecnologías pasadas. Se sugiere, por tanto, llevar a cabo revisiones periódicas de las dependencias y realizar actualizaciones según sea necesario, conforme a las buenas prácticas en la gestión de las mismas (García López, 2024).

3.5.3.2. Adecuación de Interfaces y Estándares para Interoperar

Se considera esencial la adecuación de interfaces y el cumplimiento de estándares para alcanzar la interoperabilidad entre distintos sistemas de software. En el marco de arquitecturas distribuidas, que requieren la comunicación entre múltiples componentes, resulta crucial contar con interfaces claramente definidas que faciliten el intercambio de información de forma eficiente y segura. El empleo de estándares reconocidos en la industria garantiza que los sistemas puedan intercambiar datos y ser interoperables sin inconvenientes significativos. La adaptación de interfaces y el acatamiento de los estándares previstos promueven la integración de los sistemas y aseguran una comunicación adecuada entre los mismos (Andrickson et al., 2021).

Además, es importante señalar que los modelos canónicos, definidos como representaciones estandarizadas y universales de datos o interfaces dentro de un dominio específico, pueden ser implementados en las organizaciones si se requiere de soluciones muy especializadas. En caso contrario, se puede optar por adherirse a los estándares de la industria (Mainas et al., 2023). Por ejemplo, OpenAI ofrece interfaces bien definidas que facilitan la interoperabilidad. Asimismo, herramientas como Ollama trabajan con estándares que soportan diversos modelos de lenguaje grande (LLM). Esta flexibilidad permite a las organizaciones elegir entre desarrollar soluciones personalizadas o utilizar estándares existentes que aseguran una integración y comunicación eficientes entre sistemas. La elección del enfoque adecuado dependerá de las necesidades específicas y el contexto en el que la organización opera.

Evaluación

Contents

4.1. Fuentes de datos de SST	52
4.2. Presentación del caso de estudio	52
4.2.1. Introducción	52
4.2.2. Descripción general	53
4.2.3. Requerimientos	54
4.2.4. Consideraciones	57
4.3. Arquitectura de software	58
4.3.1. Attribute Driven Design y Framework	58
4.4. Aplicación del Framework al caso de estudio	71
4.4.1. Aplicación de las buenas prácticas	71
4.4.2. Aplicación de componentes del Framework	80
4.4.3. Aplicación de ingeniería de prompt en SST	94
4.5. Pruebas y Validación	101
4.5.1. Evaluación de Arquitectura Ligera (LAE)	101
4.5.2. Evaluación con especialistas de SST	109

El presente capítulo explora el planteamiento de un caso de estudio a través del cual se busca aplicar y evaluar el Framework descrito en el capítulo anterior. Dicho framework describe un conjunto de prácticas y recomendaciones para la interoperabilidad de la Inteligencia Artificial dentro del contexto del campo de Seguridad y Salud en el Trabajo. Para lograr este planteamiento, la Corporación Talentum, con el ánimo de mejorar sus productos y ser más atractiva en el mercado del sector de SST, reunió a un equipo de especialistas en este campo. Este equipo se encargó de identificar cómo las tecnologías de IA pueden implementarse como herramientas útiles en los procesos llevados a cabo en el área de SST, así como de definir los alcances, objetivos y requisitos mínimos que se debían tener en cuenta para desarrollar un piloto o demo de un chatbot de IA enfocado en la interoperabilidad entre SST y la IA para el caso de estudio propuesto. De igual manera, a través del equipo de SST, se definió y se tuvo acceso a la fuente de datos implementada para el funcionamiento del demo. Es importante aclarar que los datos son anónimos y que se cuida en todo momento el aspecto legal de la Corporación Talentum frente a la privacidad de los datos.

Para llevar a cabo un correcto planteamiento del caso de estudio, así como de su aplicación y evaluación, las secciones de este capítulo incluyen una descripción detallada de los diferentes aspectos que compusieron el caso de estudio. Estos aspectos abarcan desde la fuente de datos y la descripción del caso de estudio o de los componentes desplegados, hasta la aplicación de estos componentes y su evaluación, con el fin de garantizar el cumplimiento de los objetivos planteados para el caso de estudio propuesto en este documento.

Mediante el desarrollo y la aplicación del caso de estudio descrito en este capítulo, se busca resaltar la importancia que tiene para la Corporación Talentum el desarrollar un demo de chatbot que implemente tecnologías de IA enfocado al campo de SST. El objetivo es poder expandir en el futuro el uso de estas tecnologías a otros campos y áreas de conocimiento.

4.1. Fuentes de datos de SST

En el presente caso de estudio, se utilizará como fuente de datos la matriz de riesgos de Seguridad y Salud en el Trabajo, la cual es una herramienta fundamental para la gestión de riesgos laborales. Esta matriz permite identificar, evaluar y priorizar los riesgos a los que están expuestos los trabajadores, así como establecer medidas de control para eliminar o reducir dichos riesgos. La información se encuentra almacenada en un archivo .XLSX que contiene datos sobre SST en diversas instituciones educativas del Valle del Cauca. Para más detalles, consulta el Apéndice A.

4.2. Presentación del caso de estudio

4.2.1. Introducción

4.2.1.1. Propósito

El presente caso de estudio tiene como objetivo explorar la interoperabilidad de la Inteligencia Artificial mediante un conjunto seleccionado de modelos de lenguaje de gran tamaño (LLM) y un sistema de información de Seguridad y Salud en el Trabajo de la Corporación Talentum. Para este fin, se pretende integrar un chat de IA con el sistema de información a fin de evaluar cómo estas tecnologías pueden facilitar el acceso, la comprensión y la manipulación de la información por parte de los usuarios, con el objetivo de mejorar su experiencia y productividad.

Para lograrlo, se propone un piloto o una demostración que incorpore Inteligencia Artificial en el sistema de información de SST de la corporación. El sistema de información de SST para la demostración será desarrollado como un componente independiente; será una versión compacta y distinta del producto de la organización, que no exhibirá ninguna funcionalidad que comprometa a la organización, dado que este producto es de propiedad exclusiva y los derechos del mismo están reservados por la Corporación Talentum. Esto permite realizar todo el proceso sin comprometer a la empresa. El producto principal no es un software de código abierto ni está disponible para el público en general, y la empresa espera que continúe siendo así. De igual manera, la corporación

está comprometida en apoyar el desarrollo del demo con especialistas, permitiendo sin problemas el desarrollo del caso de estudio. Este proyecto piloto busca interoperar la IA con el sistema de SST para mejorar la interacción de los usuarios con el sistema, permitiendo consultas de información sobre una fuente específica y ofreciendo recomendaciones personalizadas basadas en un conjunto específico de datos de SST.

4.2.1.2. Alcance

El alcance de este caso de estudio se centra en el desarrollo de un demo que permita la interoperabilidad entre la Inteligencia Artificial y el sistema de información de Seguridad y Salud en el Trabajo de la Corporación Talentum. Este demo está diseñado para incorporar un chat de IA dentro del sistema de información de SST existente, facilitando así una interacción directa y eficiente entre los usuarios autorizados y la base de datos de SST de la corporación.

Además, el caso de estudio contempla la selección e integración de tecnologías, componentes o herramientas de IA que se consideren adecuadas para alcanzar los objetivos planteados. Este enfoque busca optimizar el acceso a la información y mejorar la productividad mediante el uso efectivo de soluciones de IA adaptadas específicamente para el contexto de la seguridad y la salud en el trabajo en la Corporación Talentum.

4.2.2. Descripción general

El presente caso de estudio se enfoca en el diseño arquitectónico y la implementación de un demo como solución de Inteligencia Artificial en su prueba piloto. Esta solución está orientada a ofrecer a los usuarios una herramienta interactiva que facilite la consulta, el análisis y la interpretación de la información dentro del sistema de información de Seguridad y Salud en el Trabajo existente de la Corporación Talentum. A través de la implementación de un chat de IA, se busca mejorar la experiencia del usuario, la toma de decisiones, y generar un aumento de la eficiencia y mejoras en la productividad, entre otros beneficios.

Este enfoque permitirá que el demo actúe como un facilitador clave en el proceso de interacción usuario-sistema, optimizando el acceso a la información y contribuyendo a una gestión más efectiva de los recursos disponibles. La integración de la IA está diseñada para adaptarse a las necesidades específicas del entorno de SST, proporcionando respuestas y análisis que mejoren la capacidad de respuesta de la organización frente a los requerimientos del usuario.

4.2.2.1. Conjunto de datos

El demo propuesto en este caso de estudio se basa en un conjunto de datos predefinido que incluye información anónima sobre Seguridad y Salud en el Trabajo, recopilada en instituciones educativas del departamento del Valle del Cauca. La preservación del anonimato de la información es esencial para evitar implicaciones legales relacionadas con la privacidad y la protección de datos

personales.

El análisis de este conjunto de datos se realizará utilizando la matriz de riesgos, una herramienta diseñada para identificar, evaluar y controlar los riesgos asociados a la Seguridad y Salud en el Trabajo. Este enfoque asegura que la implementación del demo no solo respete las normativas vigentes de protección de datos, sino que también contribuya de manera efectiva a la gestión del riesgo en los entornos educativos del Valle del Cauca.

4.2.2.2. Objetivos del piloto

El piloto tiene como objetivo principal el diseño y la implementación de un entorno de interacción usuario-IA, que permita a los usuarios realizar consultas específicas y recibir recomendaciones basadas en la matriz de riesgos a partir de un conjunto predefinido de datos que contienen información de SST. Los objetivos específicos incluyen:

- **Mejorar la Interactividad:** proporcionar al usuario una herramienta interactiva que le permita acceder a información relacionada a Seguridad y Salud en el Trabajo, facilitando el acceso y la comprensión de esta información, especialmente en lo que respecta a la matriz de riesgos.
- **Potenciar la Experiencia del Usuario:** Enriquecer la experiencia del usuario mediante la implementación de tecnologías de IA, ofreciendo respuestas y recomendaciones relevantes a sus consultas.
- **Evaluación del Impacto de la IA:** Utilizar este caso de estudio para evaluar el impacto potencial de la integración de soluciones de IA en los sistemas de información relacionados con SST, con el fin de determinar la viabilidad de su implementación a mayor escala en futuras iteraciones del sistema.

4.2.3. Requerimientos

A continuación se describen las características y funcionalidades que debe tener el sistema para cumplir con las necesidades de los usuarios y alcanzar los objetivos del caso de estudio.

4.2.3.1. Requerimientos funcionales

- **RF1 - Integración del Chat de IA en el Frontend:**
 - El sistema debe integrar un chat de IA como componente interactivo en el frontend, accesible a través de las vistas seleccionadas para la interacción con el usuario.
 - Esta integración debe ser realizada de manera que se minimice el impacto sobre el diseño actual y se preserve la cohesión de la experiencia del usuario.
- **RF2 - Procesamiento y Comprensión de Datos de SST por la IA:**

- El sistema de IA implementado debe ser capaz de dar respuesta a prompts relacionados con la Seguridad y Salud en el Trabajo haciendo uso del conjunto de datos proporcionados los cuales están relacionados específicamente con la herramienta de matriz de riesgos.
- El sistema de IA implementado debe ser capaz de informar al usuario que no posee información que pueda dar respuesta al prompt en caso tal que este no este relacionado con la Seguridad y Salud en el Trabajo y su contexto no se encuentre dentro del conjunto de datos proporcionados los cuales están relacionados específicamente con la herramienta de matriz de riesgos.
- **RF3 - Interacción Intuitiva con el Usuario:**
 - El chat debe permitir a los usuarios realizar consultas de manera intuitiva, ofreciendo respuestas y recomendaciones.
 - Deberá incluirse una guía o instrucciones breves dentro de la interfaz de usuario para ayudar a los usuarios a entender cómo interactuar efectivamente con el chat de IA.
- **RF4 - Idioma de la IA:**
 - El sistema de IA debe de entender al usuario y responder a este en el idioma español.
- **RF5 - Ética y legalidad de la IA:**
 - El sistema de IA debe de ser capaz de dar respuesta a los prompts digitados por el usuario, siempre y cuando estos estén dentro del marco de lo que es ética y legalmente correcto.
- **RF6 - Capacidad de memoria a largo plazo:**
 - El sistema de IA debe contar con la capacidad de recordar elementos de una conversación previa con el usuario que haya tenido lugar durante una sesión anterior.

4.2.3.2. Requerimientos no funcionales

- **RNF1 - Seguridad y Privacidad:**
 - La IA debe operar exclusivamente con datos anónimos y preprocesados para garantizar la privacidad de los individuos y la seguridad de la información.
- **RNF2 - Interoperabilidad y Compatibilidad:**
 - La solución de IA debe ser plenamente compatible con la arquitectura actual del sistema, incluyendo el frontend en NextJS y el backend en NestJS.
 - Debe asegurarse que la integración no interfiere con la operatividad y performance del sistema existente.
- **RNF3 - Mantenimiento y Actualización del Modelo de IA:**

- Se deberán establecer procedimientos claros para la actualización periódica del modelo de IA, asegurando que la IA continúe proporcionando respuestas relevantes y precisas conforme evolucionen los datos de SST y las necesidades de los usuarios.
- Deberá planificarse el mantenimiento regular del sistema para incluir la evaluación de la efectividad de la IA y ajustes basados en feedback de los usuarios.
- **RNF4 - Rendimiento del sistema:**
 - El sistema de IA debe ser capaz de dar respuesta a los prompts digitados por el usuario en cuestión de pocos segundos.
- **RNF5 - Escalabilidad del sistema:**
 - El sistema de IA debe contar con una arquitectura capaz de escalar conforme las necesidades de los usuarios o de los datos a implementar requieran una mayor demanda por parte del sistema.
- **RNF6 - Almacenamiento:**
 - El sistema de IA debe contar con una base de datos vectorial que permita almacenar diferentes tipos de documentos de texto a manera de vectores por medio de la implementación de embeddings.
 - El sistema de IA debe contar con una base de datos vectorial que permita generar archivos de respaldo de los datos almacenados.

4.2.3.3. Requerimientos de datos

- **RD1 - Calidad en el conjunto de datos:**
 - Garantizar que el conjunto de datos tenga un tamaño adecuado para el análisis y la toma de decisiones, considerando la representatividad, la complejidad y la eficiencia.
 - Garantizar que los datos sean confiables y precisos para obtener resultados confiables y conclusiones válidas.
- **RD2 - Acceso a datos específicos de Seguridad y Salud en el Trabajo:**
 - Para formular sus respuestas y recomendaciones, la IA tendrá acceso a un conjunto predefinido de datos relacionados con la Seguridad y Salud en el Trabajo. Este conjunto de datos se centrará en el reporte de matriz de riesgos.
 - Para facilitar el procesamiento e interpretación por parte de la IA, los datos deben estar estructurados de manera adecuada. Esto implica:
 - Organizar los datos en un formato consistente y bien definido.
 - Utilizar nombres descriptivos y precisos para las variables.
 - Eliminar los datos duplicados o inconsistentes.

- **RD3 - Anonimato y Preprocesamiento de Datos:**

- Todos los datos utilizados por la IA deben ser previamente anonimizados y preprocesados para eliminar cualquier información personal identificable y asegurar la conformidad con las leyes de protección de datos.

4.2.4. Consideraciones

Este caso de estudio se concibe como un demo el cual tiene como objetivo principal evaluar la viabilidad, el impacto y la integración efectiva de una solución de Inteligencia Artificial dentro del ecosistema de Seguridad y Salud en el Trabajo. Dadas estas características, se subrayan las siguientes consideraciones especiales:

4.2.4.1. Naturaleza del piloto

- **Restricción de acceso:**

- El acceso al chat de IA será limitado a un grupo de personas. Este grupo estará compuesto por individuos directamente relacionados con la gestión y operación de SST, quienes tendrán la posibilidad y responsabilidad de proporcionar retroalimentación sobre la funcionalidad y utilidad del sistema.

4.2.4.2. Datos y privacidad

- **Uso de información anónima:**

- Como parte del demo, la IA operará exclusivamente con datos anónimos para proteger la privacidad de los trabajadores y no incurrir en problemas legales, de privacidad o de licenciamiento.

- **Compromiso con la seguridad de datos:**

- El demo se desarrolla en un entorno controlado y con datos no sensibles para no vulnerar ninguna ley o privacidad de la información, cumpliendo con las normas y regulaciones de privacidad de datos.

- **Implementación de fuentes de datos confiables y validadas:**

- Para la correcta evaluación y funcionamiento del demo, la información proporcionada por el chat de IA tiene que ser concisa, actualizada y confiable.

4.2.4.3. Seguridad y Responsabilidad

- **Implementación de medidas de seguridad:**

- Para el demo se implementan medidas de seguridad para proteger el acceso al chat de IA y los datos que contiene, con el fin de que se garantice el cumplimiento de los requisitos no funcionales referentes a la seguridad y privacidad de los datos así como a las consideraciones mencionadas en relación a los datos y privacidad.

4.2.4.4. Impacto y Sostenibilidad

- **Evaluación del impacto:**

- Como se encuentra planteado en los objetivos del piloto, este caso de estudio busca evaluar el impacto potencial de la integración de soluciones de IA en los sistemas de información relacionados con SST, con el fin de determinar la viabilidad de su implementación a mayor escala en futuras iteraciones del sistema.

- **Sostenibilidad del sistema:**

- Como parte de la prueba piloto se busca poder identificar las necesidades y expectativas de los usuarios que interactúen con el chat de IA con el fin de poder garantizar un sistema que satisfaga las necesidades de los usuarios.

4.3. Arquitectura de software

4.3.1. Attribute Driven Design y Framework

Es importante tener presente que una buena arquitectura se desarrolla mediante un análisis detallado de los drivers de negocios que se verán impactados, así como de los atributos de calidad pertinentes. En este caso, se empleará el método de Attribute Driven Design (ADD) para asegurar que el diseño de la arquitectura sea sistemático. La Corporación Talentum trabaja con esta metodología debido a su eficacia en la toma de decisiones arquitectónicas. ADD permite que el diseño arquitectónico se realice de manera sistemática, repetible y rentable, lo cual es fundamental en una disciplina de ingeniería. Según [Bass et al. \(2021\)](#), ADD guía a los arquitectos en la transformación de decisiones sobre drivers arquitectónicos en estructuras que satisfacen los requisitos y restricciones del sistema.

Además, se aplicará el Framework presentado previamente, cuyo propósito es proporcionar lineamientos clave y herramientas para la toma de decisiones arquitectónicas. Es fundamental aclarar que el Framework no se centra en una metodología puntual, sino que es adaptable y puede soportar múltiples metodologías, dependiendo de las necesidades específicas de la organización y del contexto del proyecto. Este enfoque estructurado tiene como finalidad evaluar el Framework propuesto con los requerimientos arquitectónicos del proyecto, ofreciendo flexibilidad al arquitecto de software para que pueda elegir las mejores prácticas y metodologías que se adapten a su entorno.

4.3.1.1. Diagrama de contexto

Al iniciar el proceso de diseño de una arquitectura de sistema, resulta esencial definir con claridad el alcance del mismo. Esto implica identificar con precisión los elementos que formarán parte del sistema y aquellos que quedarán excluidos. Además, es importante determinar las entidades externas con las que el sistema interactuará. Este análisis inicial es fundamental para delimitar las fronteras del sistema y asegurar que todas las interacciones con elementos externos sean gestionadas adecuadamente.

Para facilitar la visualización y comprensión de estas relaciones y límites, se puede emplear un diagrama de contexto del sistema. Este tipo de diagrama ofrece una representación gráfica que muestra el sistema en el centro, rodeado por sus entidades externas y las interfaces a través de las cuales se comunican. Al utilizar un diagrama de contexto, los diseñadores y desarrolladores pueden obtener una vista clara de cómo el sistema se integra con su entorno operativo, lo que a su vez ayuda a identificar requisitos y potenciales puntos de integración o conflicto desde las etapas tempranas del desarrollo.

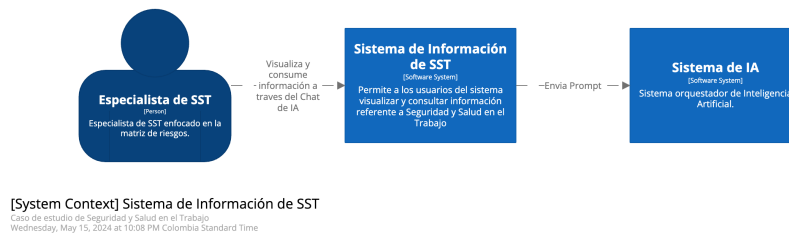


Figura 4.1: Diagrama de contexto.

4.3.1.2. Escenarios arquitectónicos

Es importante resaltar que, de acuerdo al caso de estudio y los espacios de trabajo que se realizaron con los especialistas en Seguridad y Salud en el Trabajo, los atributos de calidad que se buscan atender son: interoperabilidad y rendimiento. Los escenarios relacionados con el atributo de calidad de interoperabilidad desempeñan un papel esencial en la ejecución de las iteraciones del Attribute Driven Design (ADD). Se destaca la interoperabilidad entre Inteligencia Artificial y SST como el principal problema a abordar en el proyecto. No obstante, el rendimiento también se ha identificado como un aspecto importante, pues es necesario que el sistema de IA responda con recomendaciones en un tiempo aceptable, siendo seis segundos un intervalo adecuado para esta fase inicial de prueba y desarrollo del servicio.

Aunque la disponibilidad aún no se considera crítica, dado que se trata de un servicio en fase de maduración, es esencial garantizar que la capacidad de respuesta y la interoperabilidad entre los

sistemas se mantengan dentro de los parámetros establecidos. Estos escenarios actúan como insumos esenciales para el análisis de los “architectural drivers” que se abordarán durante las iteraciones del ADD. Por esta razón, se detalla a continuación un escenario típico asociado al atributo de calidad de interoperabilidad, y otro de rendimiento dentro del caso de estudio, destacando su relevancia en el proceso de diseño de la arquitectura. Este enfoque garantiza que ambos atributos sean considerados de manera integral en la planificación y ejecución del proyecto.

1. Escenario de interoperabilidad

Fase	Descripción
1. Fuente de estímulo:	Sistema de información de Seguridad y Salud en el Trabajo de la Corporación Talentum.
2. Estímulo:	El sistema de SST envía una solicitud al componente de IA para obtener interpretación y recomendaciones basadas en la matriz de riesgos, aplicando un conjunto específico de datos de SST relacionados con instituciones educativas anónimas.
3. Artefactos:	Componente de IA diseñado específicamente para procesar y responder consultas sobre seguridad y salud, utilizando datos anonimizados y preprocesados.
4. Ambiente::	El componente de IA opera en condiciones normales, gestionando solicitudes.
5. Respuesta:	El componente de IA procesa la solicitud, analiza los datos mediante algoritmos avanzados y emite una respuesta que incluye una interpretación de la matriz de riesgos y recomendaciones personalizadas.
6. Medida de respuesta:	El componente de IA intercambia correctamente información en el 90 % de los casos, considerando que para la Corporación Talentum, este módulo aún no es parte central del negocio.

Tabla 4.1: Escenario de interoperabilidad

2. Escenario de rendimiento

Fase	Descripción
1. Fuente de estímulo:	Un Usuario de Seguridad y Salud en el Trabajo perteneciente a la Corporación Talentum.
Continúa en la siguiente página	

Fase	Descripción
2. Estímulo:	Dicho usuario envía una consulta a través del Chat de IA, solicitando interpretación y recomendaciones basadas en la matriz de riesgos de Seguridad y Salud en el Trabajo. La consulta se realiza aplicando un conjunto específico de datos relacionados con instituciones educativas, los cuales son anónimos.
3. Artefactos:	Un sistema o componente de IA diseñado específicamente para procesar y responder consultas sobre seguridad y salud, utilizando datos anonimizados y preprocesados.
4. Ambiente:	El sistema o componente de IA opera bajo condiciones normales, gestionando solicitudes entrantes de manera rutinaria.
5. Respuesta:	Al recibir la solicitud, el sistema o componente de IA procesa la consulta, analiza los datos mediante algoritmos avanzados y, como resultado, emite una respuesta que incluye una interpretación detallada de la matriz de riesgos y recomendaciones personalizadas.
6. Medida de respuesta:	La respuesta del sistema se caracteriza por tener un tiempo de latencia promedio de 6 segundos.

Tabla 4.2: Escenario de rendimiento

Estos escenarios proporcionan un marco claro para evaluar cómo el componente de IA puede mejorar la interacción y el acceso a información crítica en el sistema de SST, mientras se mantiene alineado con las expectativas de la Corporación Talentum dentro del marco de Seguridad y Salud en el Trabajo.

4.3.1.3. Entradas para ADD

Es fundamental tener claridad sobre las entradas que se considerarán para realizar las iteraciones del proceso de Attribute-Driven Design (ADD). Estos insumos no solo son importantes para un análisis, sino que también desempeñan un papel esencial en la toma de decisiones relacionadas con el diseño de la arquitectura. La correcta selección y evaluación de estas entradas aseguran que el diseño arquitectónico se alinee efectivamente con los objetivos estratégicos y restricciones específicas del proyecto. Este alineamiento es esencial para optimizar los recursos y maximizar la eficacia y sostenibilidad del diseño propuesto. En el contexto de ADD, la claridad en las entradas permite una adaptación más precisa a los atributos críticos del sistema como la interoperabilidad, que son esenciales para el éxito del proyecto arquitectónico. A continuación, se detallan los insumos que se estarán utilizando, destacando su importancia para influir significativamente en las decisiones arquitectónicas y en el resultado final del proyecto. A continuación, se mencionan los que se están utilizando:

- **Caso de estudio:** Es presentado en una sección anterior del documento, detalla los requerimientos y restricciones del piloto o demo para el cual se está desarrollando el diseño

arquitectónico. Esta sección es importante, ya que establece los parámetros y limitaciones específicas que el diseño debe respetar, asegurando que las necesidades del proyecto sean adecuadamente abordadas para cumplir con los objetivos del caso de estudio.

- **Framework:** Un capítulo previo define el marco de trabajo que se utilizará como referencia para evaluar los aspectos arquitectónicos propuestos. Este framework es importante como guía en el proceso de toma de decisiones de diseño, orientando la estructuración arquitectónica de manera coherente y alineada con los objetivos generales del proyecto.
- **Espacios de trabajo con los especialistas de Seguridad y Salud en el Trabajo de la Corporación Talentum:** La organización está apoyando activamente en el desarrollo del proyecto, con el objetivo de que la evaluación del framework sea beneficiosa y exitosa. La colaboración con estos especialistas es esencial, ya que proporcionan conocimientos expertos que guían la integración de la Inteligencia Artificial con productos del sector de la Seguridad y Salud en el Trabajo, asegurando que el diseño arquitectónico no solo cumpla con los estándares técnicos, sino que también fomente un ambiente colaborativo para cumplir lo propuesto en el demo.
- **Escenarios de atributos de calidad:** Los escenarios de atributos de calidad son fundamentales como insumos en el ADD. Estos proporcionan una descripción detallada y clara de situaciones específicas que pueden surgir durante el funcionamiento del sistema, lo cual es importante para el proceso de diseño arquitectónico. La incorporación de estos escenarios asegura que el diseño tome en cuenta todas las condiciones relevantes, permitiendo así que el sistema responda adecuadamente bajo diversas circunstancias. Por ello, es esencial que se incluyan como parte integral de la entrada en el proceso de ADD.

Estas entradas se integran en un proceso iterativo de diseño que permite refinar continuamente la arquitectura propuesta, ajustándola a las necesidades cambiantes del entorno y del mercado. La colaboración continua con especialistas y la revisión detallada de los requisitos y restricciones son importantes para el éxito de la arquitectura final.

4.3.1.4. Objetivos de ADD

Es fundamental establecer objetivos o drivers claros al aplicar Attribute Driven Design (ADD) para asegurar un enfoque estructurado en el diseño de la arquitectura. En este sentido, se identifican varios factores clave que serán evaluados en ADD para satisfacer las expectativas de la Corporación Talentum con respecto al caso de estudio previamente mencionado:

- **Costos:** Considerando las restricciones presupuestarias del proyecto, se priorizará el uso de tecnologías de código abierto en la medida de lo posible para reducir costos. Esta decisión no solo favorece la economía del proyecto, sino que también promueve la flexibilidad y adaptabilidad en el desarrollo.

- **Rendimiento:** El rendimiento se destaca como un aspecto esencial, especialmente en aplicaciones que incorporan Inteligencia Artificial. Para el Chat IA desarrollado, es prioritario que maneje las solicitudes con baja latencia, asegurando así una experiencia de usuario ágil y fluida. Este objetivo busca optimizar los tiempos de respuesta del sistema frente a las interacciones de los usuarios.
- **Interoperabilidad:** El desafío principal radica en la interoperabilidad entre el sistema de seguridad y salud en el trabajo y el componente de Inteligencia Artificial. Se debe integrar estos sistemas de manera que se interfieran mínimamente con la arquitectura existente, manteniendo la estabilidad y la funcionalidad del entorno operativo global.

4.3.1.5. Elementos del sistema a refinar

Es importante establecer qué sistemas se planea perfeccionar o abordar en el contexto del diseño arquitectónico siguiendo la metodología de Attribute Driven Design (ADD). Dentro del proyecto actual, se ha definido un caso de estudio que implica la revisión y mejora de ciertos sistemas esenciales para cumplir con las expectativas del diseño arquitectónico. Estos sistemas son:

- El Sistema de Seguridad y Salud en el Trabajo: Este sistema está asignado para gestionar y enviar solicitudes directamente al componente de Inteligencia Artificial. Su función principal es asegurar que todas las demandas relacionadas con la seguridad y la salud en el ambiente laboral sean atendidas de manera apropiada y dirigidas correctamente para su procesamiento.
- El sistema o componente de IA: Este componente tiene la responsabilidad de procesar las solicitudes enviadas por los especialistas en Seguridad y Salud en el Trabajo. Su tarea es proporcionar recomendaciones basadas en la matriz de peligros disponible, utilizando los datos para generar respuestas que mejoren la seguridad en el lugar de trabajo.

4.3.1.6. Conceptos de Diseño

En el proceso de diseño de la arquitectura mediante el enfoque de Diseño Dirigido por Atributos (ADD), resulta esencial definir las tácticas, patrones y estilos arquitectónicos que se correspondan con los impulsores de negocio o los objetivos estratégicos del proyecto. A continuación, se especifican los elementos arquitectónicos utilizados y cómo estos se alinean con los drivers de negocio a los que apuntan. Este enfoque alineado con el Framework que sirve como guía en el proceso de definición de los conceptos de diseño, con la finalidad de mejorar el proceso de selección de estos mismos. Garantiza que todas las decisiones arquitectónicas estén directamente vinculadas a los objetivos del negocio, asegurando que la arquitectura final este alineada a los objetivos del ADD, sino que también responda a las necesidades cambiantes del entorno empresarial.

Tácticas

Es fundamental reconocer que las tácticas son herramientas valiosas en el ámbito del diseño de sistemas. Estas técnicas, similares a los patrones, han sido empleadas por arquitectos durante años

y su uso está ampliamente documentado en la literatura especializada. En el estudio de las tácticas, se busca aislar, catalogar y describir métodos que los buenos arquitectos utilizan habitualmente en la práctica. Esto no implica la invención de nuevas tácticas, sino más bien una captura y sistematización de las mejores prácticas existentes.

Las tácticas son esenciales por varias razones. Primero, aunque los patrones forman la base de muchas arquitecturas, no siempre existe un patrón que resuelva completamente un problema específico. Los arquitectos a menudo necesitan modificar y adaptar patrones a su contexto particular, y las tácticas ofrecen un medio sistemático para complementar un patrón existente y cubrir las brechas. Segundo, en ausencia de un patrón que materialice el objetivo de diseño del arquitecto, las tácticas permiten construir un fragmento de diseño desde principios básicos, proporcionando así al arquitecto una comprensión profunda sobre las propiedades del fragmento de diseño resultante. Tercero, las tácticas facilitan un enfoque más sistemático del diseño y análisis dentro de ciertas limitaciones (Bass et al., 2021).

Como cualquier concepto de diseño, las tácticas presentadas deben ser refinadas a medida que se aplican al diseño de la arquitectura. Esto subraya su poder y la razón por la que merecen una atención especial. Conocerlas y entender cómo aplicarlas será de gran beneficio en el desarrollo de la arquitectura.

A continuación, se presentan de manera resumida las tácticas seleccionadas. Para más detalles sobre la justificación de cada una, consulte el Apéndice C, donde se desglosa y amplía cada táctica, explicando la decisión de utilizarla.

Atributo de calidad	Táctica	Drivers
Rendimiento	Manage Work Requests	Rendimiento
	Maintain Multiple Copies of Computations	Rendimiento, Costos
Interoperabilidad	Adhere to Standards	Interoperabilidad, Costos
	Discover	Interoperabilidad
	Orchestrate	Interoperabilidad, Costos

Tabla 4.3: Tabla de tácticas resumida

Patrones

Los patrones arquitectónicos constituyen herramientas esenciales que abordan problemas de diseño recurrentes, los cuales han sido previamente validados por la industria en contextos de diseño específicos. Estos patrones no solo facilitan múltiples decisiones de diseño sino que también integran diversas tácticas para mejorar los atributos de calidad. De esta forma, permiten una alineación efectiva con los objetivos de negocio. Además, están vinculados a tácticas de arquitectura que ofrecen soluciones probadas para enfrentar eficazmente dichas problemáticas. A continuación, se presentan

de manera resumida los patrones seleccionados. Para más detalles sobre la justificación de cada uno, consulte el Apéndice D, donde se desglosa y amplía cada patrón, explicando la decisión de utilizarlo.

Atributo de calidad	Patrón	Drivers
Rendimiento	Load Balancer	Rendimiento
	Caching	Rendimiento, Costo
Interoperabilidad	Service-Oriented Architecture	Interoperabilidad

Tabla 4.4: Tabla de patrones resumida

Estilos de arquitectura

En el contexto del diseño arquitectónico orientado a atributos, o ADD (Attribute-Driven Design), es fundamental emplear estilos arquitectónicos de referencia que sirvan como guía en el proceso de diseño. Los estilos arquitectónicos no solo definen las estructuras fundamentales de los sistemas, sino que también influyen en los atributos de calidad como la escalabilidad, seguridad y mantenibilidad. A continuación, se detallan dos estilos arquitectónicos específicos que son pertinentes para el diseño mediante ADD, cada uno con sus características particulares, ventajas y consideraciones, adaptados para satisfacer necesidades y resolver desafíos específicos en entornos de software variados.

1. Estilo de Arquitectura Basada en Servicios

La arquitectura basada en servicios es un enfoque que se sitúa entre los estilos de microservicios y las arquitecturas monolíticas. Este estilo se caracteriza por distribuir las responsabilidades de manera que se equilibre la carga operativa sin llegar a la granularidad de los microservicios, ni concentrar todas las funciones en un único bloque, evitando así los extremos de modularidad.

Ventajas:

- Flexibilidad y modularidad: Permite adaptaciones rápidas y eficientes en respuesta a cambios o nuevas demandas del negocio.
- Reutilización de componentes: Fomenta la economía de recursos al permitir la reutilización de servicios existentes en nuevas aplicaciones.

Trade-offs:

- Acoplamiento a través de la Base de Datos: Compartir una base de datos puede incrementar las dependencias entre servicios, complicando la escalabilidad y el mantenimiento.
- Gestión de balanceo de carga y transacciones: Requiere estrategias efectivas para mantener la disponibilidad y el rendimiento del sistema.

Justificación para el caso de estudio: Este estilo es adecuado para sistemas que requieren una estructura flexible que facilite tanto la independencia en el desarrollo y despliegue de componentes, como la interoperabilidad entre distintas tecnologías, esencial en entornos empresariales con diversidad tecnológica. Según el framework propuesto, la arquitectura basada en servicios ilustra claramente los trade-offs y beneficios, proporcionando una guía robusta y orientada dentro del proceso de arquitectura de ADD, lo que facilita la toma de decisiones arquitectónicas informadas.

2. Estilo de Arquitectura RAG (Retrieval-Augmented Generation)

La arquitectura RAG representa una innovación en el campo de la Inteligencia Artificial, combinando técnicas de generación de texto con la recuperación de información. Este enfoque aprovecha fuentes de información externas para enriquecer las respuestas generadas por modelos de IA, lo que aumenta su relevancia y precisión.

Ventajas:

- Precisión y relevancia mejoradas: La integración de información actualizada y contextual mejora significativamente la calidad de las respuestas.
- Reducción de sesgos: La diversidad de fuentes puede ayudar a crear modelos más equitativos y justos.

Trade-offs:

- Dependencia de la calidad de las fuentes: La eficacia de este enfoque puede verse limitada si las fuentes de información no son confiables o están desactualizadas.
- Latencia en tiempo de respuesta: La recuperación de información puede introducir demoras que afecten la velocidad de respuesta del sistema.

Justificación para el caso de estudio: Este estilo es ideal para aplicaciones en las que la precisión y la actualización de la información son críticas, como en sistemas de respuesta automática y asistentes inteligentes, donde la capacidad de integrar y contextualizar datos en tiempo real puede diferenciar significativamente la calidad de las interacciones y decisiones. El framework propuesto enfatiza cómo la arquitectura RAG puede servir como un modelo guía en el proceso de arquitectura de ADD, destacando los beneficios y compensaciones para facilitar un diseño arquitectónico.

4.3.1.7. Elementos arquitectónicos y responsabilidades

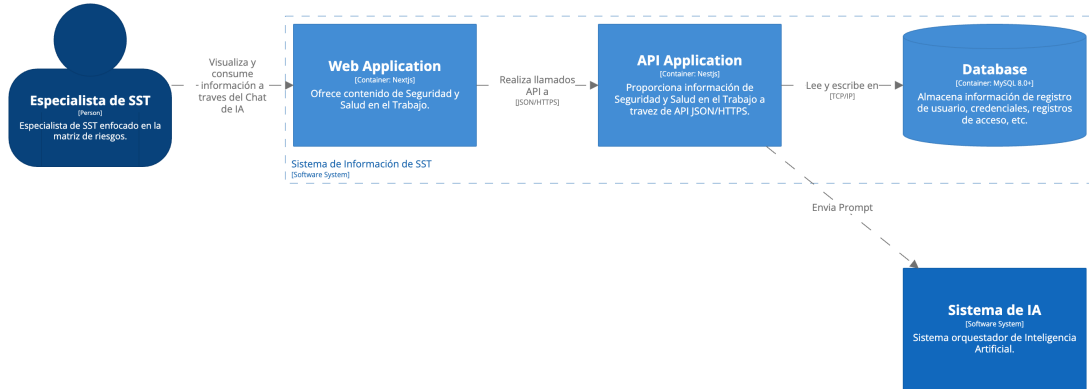
En el proceso de diseño guiado por atributos, es fundamental establecer claramente las responsabilidades de los elementos arquitectónicos que formarán parte del diseño de la arquitectura. Cada elemento tiene responsabilidades específicas, las cuales se detallan en la tabla que se presenta a continuación.

Núm.	Componente	Responsabilidad
1	Filebrowser	Navegador de archivos web de tipo open source y multiplataforma que permite a los usuarios acceder, gestionar y compartir archivos almacenados en servidores remotos o locales. Ofrece una interfaz web intuitiva y fácil de usar, compatible con diversos navegadores y dispositivos.
2	Flowise	Plataforma de tipo open source para la automatización de flujos de trabajo, de bajo código y permite a los usuarios crear y gestionar flujos de trabajo automatizados para agentes de IA. Ofrece una interfaz intuitiva de arrastrar y soltar para diseñar flujos de trabajo, integrando con diversas aplicaciones y servicios web.
3	Qdrant	Bases de datos vectorial de tipo open source que permite almacenar textos convertidos a vectores para realizar una búsqueda de similitud vectorial de manera eficiente. Ofrece una arquitectura escalable y distribuida, ideal para manejar grandes volúmenes de datos.
4	DynamoDB	Base de datos NoSQL escalable, de alto rendimiento, altamente disponible y sin servidor, ofrecida por AWS. Permite almacenar y acceder a grandes volúmenes de datos de forma rápida y fiable, ideal para aplicaciones modernas que requieren un acceso rápido y fiable a grandes volúmenes de datos.
5	AWS Bedrock	Servicio totalmente gestionado que ofrece una variedad de modelos de base de alto rendimiento (FMs) de empresas líderes en IA como AI21 Labs, Anthropic, Cohere, Meta, Mistral AI, Stability AI y Amazon a través de una única API.
6	AWS Bedrock Embeddings	Representaciones numéricas de objetos del mundo real que los LLM e IA utilizan para comprender conocimientos como lo hacen los humanos. Estos embeddings se generan a partir de modelos matemáticos y se utilizan para mejorar las capacidades de las aplicaciones de IA generativa.
7	Langfuse	Plataforma de carácter open source que ayuda a los equipos analizar y observar por medio de diferentes métricas el comportamiento de las aplicaciones de LLM desplegadas.
8	Lunary	Herramienta open source que ayuda al análisis y observación de las aplicaciones de LLM desplegadas por medio de diferentes métricas, así como también la evaluación y seguimiento de los prompts.

Tabla 4.5: Responsabilidad de componentes arquitectónicos

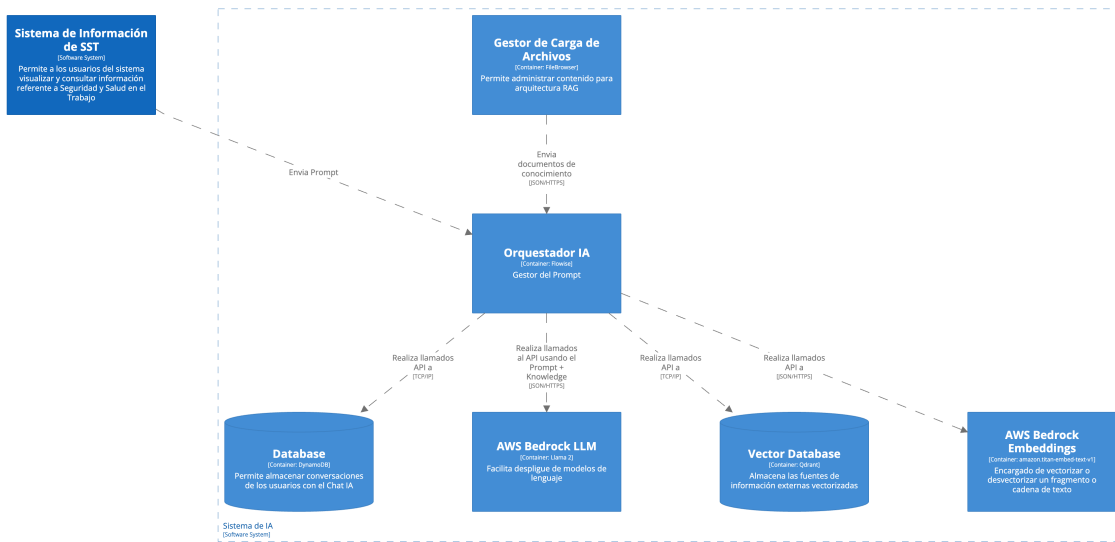
4.3.1.8. Vistas y Registro de Decisión Arquitectónica (ADR)

Como resultado de las iteraciones, se exhiben los componentes elegidos para cumplir con las entradas en la vista siguiente:



[Container] Sistema de Información de SST
 Caso de estudio de Seguridad y Salud en el Trabajo.
 Wednesday, May 15, 2024 at 10:04 PM Colombia Standard Time

Figura 4.2: Diagrama de contenedores del sistema de información de SST.



[Container] Sistema de IA
 Caso de estudio de Seguridad y Salud en el Trabajo.
 Wednesday, May 15, 2024 at 10:08 PM Colombia Standard Time

Figura 4.3: Diagrama de contenedores del sistema de IA.

Decisiones de Arquitectura

Con el objetivo de documentar apropiadamente las decisiones arquitectónicas, se procede a documentar las decisiones tomadas en los Registros de Decisión de Arquitectura (ADR) tras las iteraciones del proceso ADD. Los ADR son vitales para capturar decisiones importantes relacionadas con la arquitectura del software. Estos registros incluyen una descripción detallada de por qué se optó por una determinada solución arquitectónica, las alternativas que fueron consideradas y el impacto de estas decisiones en la estructura general del sistema.

La documentación en los ADR resulta esencial para conservar un historial de las decisiones y los fundamentos detrás de cada una, facilitando así la comprensión y el análisis de la arquitectura del software a lo largo del tiempo. Además, ayudan a los nuevos integrantes del equipo a comprender las razones detrás de las decisiones tomadas y cómo estas influyen el diseño y funcionamiento del sistema. A continuación, se presenta un ADR documentado para ilustrar la estructura utilizada. Para más detalles o información sobre todos los ADRs documentados, consulte el Apéndice B.

Título	ADR-INT-03 Arquitectura de referencia para extracción de información (RAG)
Estado	Aceptado
Contexto	Se requiere implementar un modelo de arquitectura que permita al LLM poder buscar información relevante en una base de datos o conjunto de datos externo, así como generar texto nuevo basado en la información almacenada por medio de una variedad de técnicas, como aprendizaje automático o plantillas, para generar texto.
Decisión	Se decide implementar la Arquitectura RAG ya que permite a los LLM generar texto que es más preciso, informativo y relevante que el que podrían generar por sí solos. Esto se debe a que los LLM tienen acceso a una gama más amplia de información y pueden usar esa información para generar texto más relevante y específico.
Consecuencias	<ul style="list-style-type: none"> ▪ Generación de texto más preciso en comparación con los LLM que no implementan la arquitectura RAG. ▪ Generación de texto más relevante para la consulta del usuario. ▪ Respuesta con texto más informativo y detallado. ▪ Modelos de LLM más creativos capaces de encontrar nuevas ideas y perspectivas en la generación de respuestas.
Continúa en la siguiente página	

Tabla 4.6 – Continúa de la página anterior

Título	ADR-INT-03 Arquitectura de referencia para extracción de información
---------------	--

Tabla 4.6: ADR de interoperabilidad Núm. 03: Arquitectura RAG para extraer e interoperar información con la IA

4.4. Aplicación del Framework al caso de estudio

En el proceso de diseño arquitectónico, se aplicó el método ADD (Attribute Driven Design) y se documentaron los ADR (Architectural Decision Record). Esta sección del documento se enfoca en profundizar en la aplicación del Framework, con el objetivo de clarificar su uso y cómo orientó el proceso de diseño arquitectónico en la toma de decisiones. Se amplía la información y se explica cómo el Framework facilita y guía la interoperabilidad entre el sistema de información de Seguridad y Salud en el Trabajo y el componente o sistema de Inteligencia Artificial. La intención es demostrar la implementación del Framework durante la fase de diseño arquitectónico en el desarrollo de software, así como su papel en la ingeniería de prompt, especialmente en el ámbito de Seguridad y Salud en el Trabajo.

A continuación, se detallan varios apartados que describen cómo se aplicaron los elementos del Framework. Se tratan temas como la adopción de buenas prácticas, la ingeniería de prompt, y otras recomendaciones y orientaciones proporcionadas por el Framework. Esta descripción no solo revela la implementación de estas directrices, sino también cómo contribuyen a una mayor cohesión y funcionalidad entre los sistemas involucrados, asegurando una integración que satisface las necesidades específicas del ámbito de Seguridad y Salud en el Trabajo.

4.4.1. Aplicación de las buenas prácticas

4.4.1.1. Modelos de lenguaje de gran tamaño LLM

Para el desarrollo del demo, se tuvo en cuenta el listado de buenas prácticas referentes a los modelos de lenguaje de gran tamaño LLM. Las buenas prácticas aplicadas fueron:

- **Selección del modelo adecuado:** Referente a esta buena práctica, se seleccionaron dos modelos para el desarrollo del demo: llama2-70b-chat-v1 y gpt-3.5-turbo, desarrollados por la empresa Meta y OpenAI respectivamente. A continuación se presenta una tabla con información de ambos modelos:

Característica	Llama2-70b-chat-v1	GPT-3.5-Turbo
Desarrollador	Meta	OpenAI
Fecha de lanzamiento	18 de julio de 2023	28 de noviembre de 2022
Continúa en la siguiente página		

Característica	Llama2-70b-chat-v1	GPT-3.5-Turbo
Precio	Open source	\$0.50 USD por millón de tokens
Tokens máximos	4.096	4.096
Características	<ul style="list-style-type: none"> • Es un modelo de lenguaje grande con 70 mil millones de parámetros, lo que le permite aprender y generar texto de una manera más compleja y matizada que los modelos más pequeños. • Está entrenado en un conjunto de datos diverso de texto y código, lo que le permite generar texto más preciso y relevante para una variedad de tareas. • Se puede ajustar para realizar tareas específicas, como responder preguntas, escribir diferentes tipos de contenido creativo o traducir idiomas. • Es de código abierto, lo que significa que cualquier persona puede usarlo y modificarlo. 	<ul style="list-style-type: none"> • Gracias a su arquitectura optimizada y a un proceso de entrenamiento más eficiente posee una buena velocidad de procesamiento y capacidad de manejar tareas complejas. • Consume menos recursos computacionales que otros modelos, lo que lo hace más rentable y accesible para una amplia gama de usuarios. • Está diseñado para aplicaciones de chat, con habilidades mejoradas para mantener conversaciones fluidas y coherentes, comprender el contexto y generar respuestas relevantes e informativas. • Se puede ajustar para realizar tareas específicas, como escribir diferentes tipos de contenido creativo, responder preguntas o traducir idiomas. • Está disponible a través de la API de OpenAI, lo que permite a los desarrolladores integrarlo fácilmente en sus aplicaciones y servicios.

Tabla 4.7: Comparativa entre los modelos llama2-70b-chat-v1 y gpt-3.5-turbo

Dando respuesta a la buena práctica de seleccionar el modelo adecuado, estos dos modelos fueron seleccionados a razón de que cumplan con los criterios listados en el Framework, los cuales son:

- **Reproducibilidad:** Si bien es cierto que ambos modelos no siempre generan respuestas idénticas al mismo prompt, sí cumplen con el criterio de reproducibilidad en el sentido de que ofrecen respuestas consistentes y precisas cuando se les consulta el mismo tema en repetidas ocasiones. Esto se debe a la solidez de los datos utilizados para su entrenamiento y a la estabilidad de las versiones del modelo empleadas. En otras palabras, cada modelo posee la capacidad de generar respuestas coherentes y confiables frente a un prompt específico, incluso cuando se le presenta varias veces.
- **Ética y legalidad:** Para ambos modelos se cumple este criterio, ya que ambos modelos seleccionados no dan respuesta a un prompt que pueda incurrir en transgresiones a los estándares éticos y legales respecto a las normativas vigentes. Cada modelo escogido ha sido entrenado bajo responsabilidad ética con el ánimo de no afectar a la sociedad en general.
Como prueba de esto, se ha comprobado que el modelo GPT-3.5-Turbo, al recibir una pregunta sobre “cómo fabricar un arma de fuego”, no proporciona ninguna respuesta. De manera similar, el modelo Llama2-70b-chat-v1 no solo evita responder la pregunta, sino que además ofrece una explicación adicional sobre por qué no está permitido brindar ese tipo de información. Esto demuestra el compromiso de ambos modelos con la ética y la legalidad en la gestión de datos y respuestas.
- **Transparencia:** En relación a este criterio, ambos modelos se destacan por su transparencia en la disponibilidad de información ya que cuentan con una amplia documentación oficial y de terceros a la que se puede acceder fácilmente a través de portales web. Además, esta transparencia se ve reforzada por el compromiso de las empresas desarrolladoras con respecto a la ética en el entrenamiento de los modelos. Este enfoque ético garantiza que los modelos se entrenen con datos de alta calidad y se implementen de manera responsable.
- **Cultura e idioma:** Para ambos modelos se cumple este criterio, ya que ambos modelos seleccionados son capaces de responder al usuario en el idioma español, el cual es el idioma requerido para el desarrollo del demo y en el cual se encuentra la información implementada.
- **Escalabilidad:** Con respecto a este criterio, si bien ambos modelos cuentan con un alto nivel de escalabilidad ya que pueden ser desplegados en la nube y cuentan con la posibilidad de aumentar sus recursos por medio de la máquina virtual o API en que se encuentren desplegados, cabe mencionar que por un lado llama2-70b-chat-v1 es un modelo de código abierto, mientras que gpt-3.5-turbo funciona por medio de un sistema de pago, lo que significa que a mayor necesidad de recursos, mayor será el costo de implementación por parte de este modelo.
- **Complejidad:** En relación con el último criterio de selección de modelos, se demostró que ambos, el modelo GPT-3.5-Turbo y el modelo Llama2-70b-chat-v1, poseen la capacidad de ejecutar tareas de análisis de datos. Esta habilidad se evidencia claramente en los resultados obtenidos. Se solicitó al modelo GPT-3.5-Turbo que identificara las

acciones de Seguridad y Salud en el Trabajo necesarias para la Institución Anónima 1. Posteriormente, el modelo Llama2-70b-chat-v1 fue consultado sobre el municipio al que pertenece dicha institución. Las respuestas de ambos modelos no solo fueron satisfactorias, sino que también demostraron su capacidad para extraer y analizar información de la matriz de riesgo, cumpliendo así con el criterio establecido.

- **Comprensión de las limitaciones del modelo:** Comprender las limitaciones de los modelos de lenguaje existentes, como gpt-3.5-turbo y llama2-70b-chat-v1, fue esencial para el desarrollo, administración y mejora de la arquitectura RAG. Al identificar las debilidades de estos modelos, se optimizaron los componentes y otras características clave que permitieron crear un sistema más adaptable para el procesamiento del lenguaje natural. A continuación, se presentan algunas limitaciones de ambos modelos que fueron fundamentales en el proceso.

Característica	Llama2-70b-chat-v1	GPT-3.5-Turbo
Modelo base	LLaMA (Large Language Model by Meta)	GPT-3.5
Número de parámetros	70 billones	Alrededor de 6 billones
Capacidad de procesamiento de texto	Alto rendimiento en aplicaciones interactivas	Adecuado para aplicaciones de baja a media interactividad
Entrenamiento	Datos hasta enero de 2023	Datos hasta septiembre de 2021
Habilidades lingüísticas	Superior en comprensión y generación de texto debido a más parámetros y entrenamiento más reciente	Alto rendimiento en comprensión y generación de texto
Limitaciones de uso	Mejor para tareas específicas y detalladas	Adecuado para tareas más generales y menos específicas
Limitaciones éticas y de sesgo	Potencialmente menos sesgos, pero no exento de problemas	Sesgos inherentes no completamente mitigados
Capacidad de personalización	Alta, posibilidad de modificar y hacer fine-tuning al modelo	Limitada, no se puede modificar el modelo
Escalabilidad	Ideal para aplicaciones a gran escala y de alta demanda	Buena para aplicaciones medianas
Costo y accesibilidad	Código abierto, gratuito para modificar y usar, costos asociados al cómputo	Servicio basado en suscripción, costo según uso

Continúa en la siguiente página

Característica	Llama2-70b-chat-v1	GPT-3.5-Turbo
Capacidad de fine-tuning	Disponible para usuarios, permite adaptar el modelo a necesidades específicas	No disponible directamente para usuarios

Tabla 4.8: Limitaciones identificadas en los modelos gpt-3.5-turbo y llama2-70b-chat-v1

- Calidad y diversidad de los datos:** Es fundamental considerar que los modelos de Inteligencia Artificial, específicamente el gpt-3.5-turbo de OpenAI y el llama2-70b-chat-v1 de Meta, ya vienen preentrenados. Sin embargo, a menudo es necesario incorporar información adicional al contexto de la IA, lo que implica añadir fuentes de datos o información que no estén incluidas en el preentrenamiento original. Esto contribuye a una mayor diversidad de información, especialmente relevante en el contexto de la Seguridad y Salud en el Trabajo. Con el objetivo de controlar las fuentes de datos a las cuales tienen acceso los modelos, se ha implementado la arquitectura RAG (Retrieval-Augmented Generation). Esta arquitectura permite que el modelo no solo funcione con los datos con los que fue entrenado, sino que también tenga la capacidad de adquirir nuevos datos, información y contexto mediante la carga de archivos.

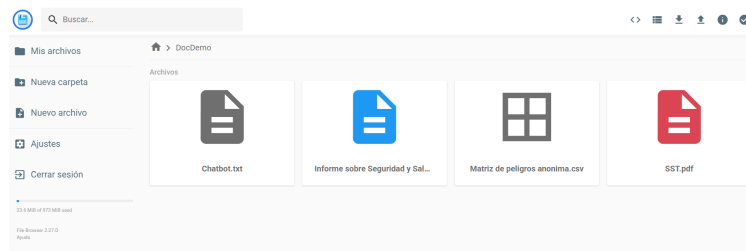


Figura 4.4: Tipos de archivos de las fuentes de datos implementadas para el contexto de los modelos

Como se muestra en la Figura 4.5, para el desarrollo del demo se ha seguido esta buena práctica, trabajando con una variedad de diferentes tipos de archivos como .TXT, .DOCX, .CSV y .PDF. Esto garantiza una alta compatibilidad con los diversos archivos que se puedan cargar en ambos modelos. Además, para asegurar la calidad de esta práctica, los archivos cargados contienen información relevante para el contexto de SST manejado en el demo y su contenido ha sido previamente revisado para garantizar su calidad. Esta estrategia asegura que la información utilizada sea pertinente y útil para mejorar el rendimiento y la aplicabilidad del modelo en escenarios específicos de SST.

- Consideraciones éticas y mitigación de sesgos:** Los modelos de lenguaje de aprendizaje automático están entrenados en conjuntos de datos masivos que, debido a su amplitud y diversidad, pueden reflejar sesgos del mundo real, tales como sesgos de género, raciales o

socioeconómicos. Estos sesgos pueden llevar a que los modelos generen resultados sesgados o injustos. Para mitigar este problema en los modelos implementados, se adoptaron medidas que reducen el nivel de sesgos en los resultados obtenidos. Entre estas medidas, se incluyó la carga exclusiva de documentos relacionados con el área de Salud y Seguridad en el Trabajo (SST), así como la configuración de los modelos para limitar sus respuestas a los límites establecidos.

Además, en un esfuerzo por continuar con la buena práctica de mitigación de sesgos y evitar la discriminación hacia los individuos o instituciones mencionados en los documentos accesibles por los modelos, se revisaron los archivos cargados para garantizar el anonimato de la información, eliminando así cualquier sesgo potencial.

El marco de trabajo propuesto anteriormente, sugiere implementar procedimientos de actualización y mantenimiento de los modelos para asegurar que sigan proporcionando respuestas relevantes y precisas. Dentro de las buenas prácticas adoptadas por la Corporación Talentum, se destacan los siguientes aspectos recomendados por el marco:

- **Recopilación de datos:** Se lleva a cabo la recopilación y el procesamiento periódico de nuevos datos de SST para mantener la información actualizada y relevante para el personal que requiere apoyo de la Inteligencia Artificial.
- **Evaluación del rendimiento:** Se evalúa periódicamente el rendimiento del modelo utilizando métricas relevantes como la precisión, la sensibilidad y la especificidad.
- **Validación y pruebas:** Se validan y prueban las versiones actualizadas de los modelos para mejorar su precisión y aportar más valor al área de SST, cumpliendo con los requisitos establecidos en el caso de estudio.
- **Actualización de versiones de modelos LLM:** Considerando la rapidez con la que se lanzan nuevas versiones de modelos por parte de empresas como Meta y OpenAI, se realiza una revisión periódica para mantener los modelos actualizados. Este proceso no es automático y requiere de una revisión manual. La búsqueda activa de nuevas versiones, mejoras pre-entrenadas y lanzamientos por parte de los desarrolladores es esencial. Se recomienda revisar las fuentes oficiales de información de estas empresas al menos dos veces al mes. En caso de detectar nuevas versiones o mejoras, es importante evaluar su impacto potencial y realizar las actualizaciones necesarias en los sistemas que utilizan estos modelos.

Estas prácticas ya implementadas por la Corporación Talentum esperan mejorar de manera continua el uso de los modelos en términos de precisión y relevancia, así como la mitigación efectiva de sesgos, lo que contribuye significativamente a la integridad y eficacia del trabajo en SST.

4.4.1.2. Estructura de datos

Para el desarrollo del piloto asociado al caso de estudio, se consideró el listado de buenas prácticas relacionadas con la estructura de los datos. Las prácticas implementadas incluyen:

- **Comprender los requisitos específicos orientados en los datos:** Según el marco de trabajo establecido, el primer paso es comprender los requisitos de los datos claves. En este caso de estudio, que desarrolla un demo o piloto para la interoperabilidad de IA con fuentes de datos de Seguridad y Salud en el Trabajo, se requiere el uso exclusivo de material del área de SST, como son los informes de evaluación de riesgos y las matrices de riesgos de SST. Los requisitos clave identificados para la estructura de los datos fueron:
 - La capacidad del demo para procesar y almacenar vectores de documentos en formatos .CSV, .DOCX y PDF de manera que se mejore la precisión de la IA.
 - El contenido de los datos debe ser exclusivo del campo de SST, descartando cualquier información irrelevante para el demo.
 - La información contenida en los archivos debe ser anónima, modificando nombres de personas o instituciones para garantizar el anonimato.
- **Identificar la base de datos vectorial:** Para el desarrollo del demo se seleccionó la base de datos vectorial Qdrant, conocida por su capacidad para manejar una gran diversidad de tipos de datos y complejas condiciones de consulta. Esta selección estratégica fue debido a la necesidad de adaptar la tecnología a las demandas específicas de procesamiento y análisis en el ámbito de la Seguridad y Salud en el Trabajo. Qdrant no solo permite realizar consultas que incluyen coincidencias de cadenas, rangos numéricos y ubicaciones geográficas, sino que también es capaz de soportar consultas avanzadas necesarias para extraer, comparar y analizar datos vectoriales de manera eficiente.

Esta base de datos vectorial es especialmente beneficiosa en escenarios donde la personalización de la lógica de negocio es fundamental. Permite a los desarrolladores construir y modificar dinámicamente las consultas de datos basadas en similitudes, lo que es esencial para aplicaciones como la Inteligencia Artificial en SST, donde se requiere una precisión y una adaptabilidad excepcionales. Además, Qdrant ofrece opciones de seguridad y mantenimiento, incluyendo funciones avanzadas para la realización de copias de seguridad automáticas y la recuperación de datos, asegurando que la información vital no se pierda en caso de fallos técnicos o siniestros. Esta base de datos es, por tanto, un punto importante en la infraestructura de datos del demo, facilitando una interacción fluida y eficiente entre los distintos componentes del sistema.

- **Garantizar un almacenamiento óptimo en la base de datos vectorial:** En la aplicación del Framework al caso de estudio, se menciona la importancia de asegurar que la información

se almacene de manera eficiente en la base de datos vectorial, en esta caso Qdrant. Dado que las bases de datos vectoriales establecen un límite máximo para el tamaño de los vectores, es necesario adaptar los documentos para cumplir con estas restricciones de tamaño. Por lo tanto, se realiza un proceso de transformación de datos que divide los documentos extensos en segmentos más pequeños, facilitando su manejo y procesamiento. Este procedimiento se lleva a cabo utilizando un divisor de texto denominado “Character Text Splitter”. Este componente es crucial para el proceso de preparación de los datos, ya que divide el contenido de los documentos en fragmentos más pequeños y manejables. Este enfoque no solo cumple con las limitaciones de tamaño de la base de datos, sino que también mejora la eficacia de la vectorización y el posterior proceso de extracción de información. Además, la división de textos en unidades más pequeñas permite un análisis más detallado y una indexación más precisa en la base de datos, lo que resulta en una mejor precisión de los modelos de Inteligencia Artificial que operan con estos datos. Implementar este método ayuda a que toda la información relevante sea accesible y utilizable, optimizando así la capacidad de la base de datos vectorial para responder a consultas complejas y mejorar el desempeño general del sistema.

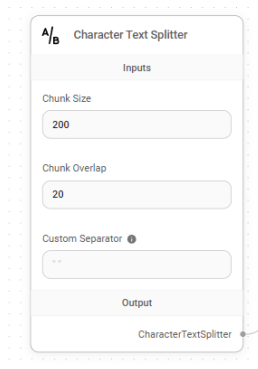


Figura 4.5: Componente “Character Text Splitter” del orquestador Flowise

4.4.1.3. Bases de Datos Vectoriales

Dentro del Framework, se enfatiza la importancia de seguir un flujo de trabajo estructurado para la vectorización de la información. Este proceso incluye recomendaciones y criterios esenciales para las decisiones en el diseño guiado por atributos (Attribute Driven Design, ADD). A continuación, se describe el proceso seguido bajo la orientación del Framework, detallando la aplicación de estas directrices.

1. **Data:** La matriz de peligros proporcionada por los especialistas en Seguridad y Salud en el Trabajo de la Corporación Talentum constituye la fuente de datos principal. Originalmente, estos datos se encuentran en un archivo .XLSX con una estructura diseñada para generar estadísticas. Siguiendo las directrices del Framework, se llevó a cabo un proceso de

transformación de datos, convirtiendo el archivo a formato .CSV para garantizar una mayor compatibilidad con los componentes arquitectónicos desplegados. Este cambio facilita la orquestación e interoperabilidad en la aplicación de Inteligencia Artificial. Es importante anonimizar cualquier dato sensible antes de este proceso para prevenir problemas legales. Además, se emplea Filebrowser para gestionar los archivos .CSV que se utilizan como fuentes de datos en el proceso arquitectónico RAG.

2. **Vector Embedding:** En el desarrollo del caso de estudio, se consideró importante la aplicación de prácticas avanzadas de los Modelos de Lenguaje de Gran Escala (LLM) para optimizar la vectorización de los datos todo esto guiado por el Framework propuesto. Dos modelos destacados fueron el llama2-70b-chat-v1 de Meta y el gpt-3.5-turbo de OpenAI. Aunque cada LLM propone métodos distintos para el manejo de embeddings, se decidió adoptar modelos de embedding preexistentes debido a su comprobada superioridad en desempeño, desplegados en AWS mediante un servicio serverless llamado AWS Bedrock. En concreto, se emplearon el modelo amazon.titan-embed-text-v1 para el llama2-70b-chat-v1 y el text-embedding-ada-002 para el gpt-3.5-turbo. Estas elecciones responden a las directrices del Framework que subraya la importancia de utilizar tecnologías que mejoren la precisión y eficiencia en el procesamiento y análisis de datos vectoriales.
3. **Indexing:** Para el desarrollo del caso de estudio, se seleccionó la base de datos vectorial Qdrant para gestionar la indexación de vectores. Esta elección se basa en las recomendaciones del Framework que sugieren utilizar bases de datos especializadas que simplifiquen la búsqueda de similitudes y mejoren la precisión. Qdrant emplea el algoritmo “Hierarchical Navigable Small World Graph” (HNSW), un método avanzado que construye una estructura de navegación por capas, donde cada nivel tiene distintas densidades y distancias entre nodos. La búsqueda inicia en las capas superiores y se desciende secuencialmente, facilitando un acercamiento eficiente al objetivo deseado (Vasnetsov, 2021). Este proceso automático de indexación se integra perfectamente con la configuración del orquestador para ofrecer un rendimiento óptimo en la búsqueda de vectores similares.
4. **Almacenamiento en la base de datos vectorial:** Qdrant ha sido seleccionado como la base de datos vectorial para el el caso de estudio, siguiendo las recomendaciones de las prácticas aconsejadas por el Framework. Esta plataforma se destaca por su capacidad para integrarse de manera eficiente con diversas tecnologías de Inteligencia Artificial. Se implementa a través de un servicio de API, que facilita la búsqueda de los vectores de alta dimensionalidad más próximos. Su despliegue en la nube permite una escalabilidad horizontal, asegurando un uso eficiente de los recursos. Además, Qdrant utiliza el algoritmo “Hierarchical Navigable Small World Graph”, que facilita un proceso de indexación preciso y eficiente, como se observa en la Figura, donde se menciona “hnsw_config” en la información de la colección creada.

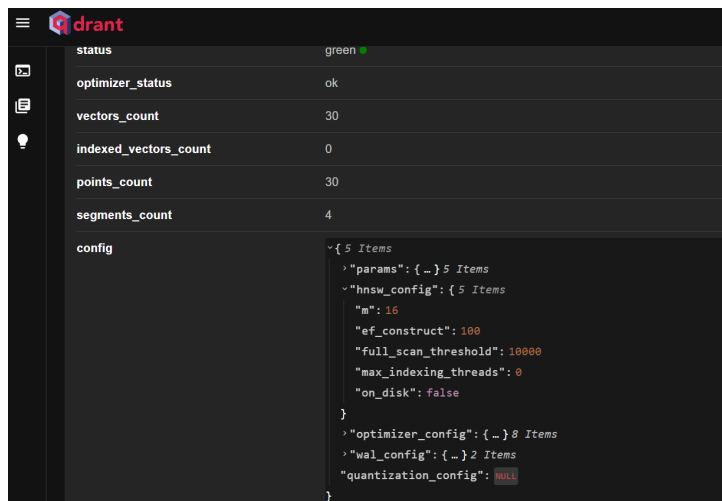


Figura 4.6: Indexación en Qdrant

En términos de estructura de datos, se ha implementado el componente “Character Text Splitter” para segmentar los archivos cargados y evitar la sobrecarga de la base de datos con archivos excesivamente grandes. Se ha prestado atención especial a la nomenclatura de las colecciones dentro de Qdrant, utilizando nombres claros y descriptivos, como “DEMO_SST”. Esto facilita la identificación y acceso rápido a las colecciones específicas, apoyando una gestión de datos más ágil y menos propensa a errores. Finalmente, se consideró la capacidad de Qdrant para realizar copias de seguridad de los datos, denominadas “snapshots”, lo cual es fundamental para la preservación de la información en situaciones de pérdida de datos potencial.

5. **Querying:** Este proceso aborda la ingeniería de prompts de manera detallada, presentando técnicas y estrategias para optimizar las consultas dentro de aplicaciones de Seguridad y Salud en el Trabajo. El Framework aporta principios clave para el diseño de consultas efectivas, con el fin de maximizar la precisión y la recuperación de información relevante. A través de un enfoque sistemático, se exploran los mecanismos de consulta que permiten a los usuarios interactuar de manera intuitiva con bases de datos vectoriales, facilitando así la obtención de resultados precisos y pertinentes. La claridad en la formulación de consultas se considera esencial para la eficiencia de las búsquedas, y este enfoque se detalla más adelante para facilitar la comprensión de las buenas prácticas en la ingeniería de prompts enfocada en SST.

4.4.2. Aplicación de componentes del Framework

Para la aplicación del framework, se integraron diversos componentes a través de un orquestador. El componente de orquestación facilita la generación de respuestas por parte de la Inteligencia Artificial, basándose en las interacciones de prompts que ocurren entre el usuario y el chatbot a través de la interfaz del Sistema de Información de Seguridad y Salud en el Trabajo (SI de SST).

La sección aborda los componentes utilizados en el flujo de orquestación que el Framework propone como guía para la interoperabilidad. Se presenta un diagrama de alto nivel de la solución, con el objetivo de profundizar gradualmente en el desarrollo de la sección hasta alcanzar los componentes más pequeños. En este diagrama se muestra la integración del SI de SST con la Inteligencia Artificial, lo cual abarca la orquestación de múltiples componentes.



Figura 4.7: Diagrama de alto nivel de la solución.

4.4.2.1. Sistema de información de SST

El Sistema de Información de Seguridad y Salud en el Trabajo (SI de SST) es una herramienta desarrollada por la Corporación Talentum que facilita la gestión de riesgos laborales en las empresas. Este sistema permite que el área de SST visualice y analice los datos necesarios según lo determinen los especialistas del área. El SI de SST ofrece una amplia gama de funcionalidades que permiten gestionar de manera eficiente la seguridad y la salud en el trabajo. Entre estas funcionalidades, se incluye la visualización de información de diferentes artefactos que utilizan información de SST, como la matriz de riesgos, y la generación de informes basados en los datos de esta matriz. Además, para un caso de estudio específico, se desarrolló un producto con funcionalidades mínimas que pueden ser accesibles al público sin impactar la organización ni la privacidad de la información que representa un producto central para la empresa.

Esta versión se encarga de interoperar con Inteligencia Artificial utilizando fuentes de datos de SST, todo mediante la integración de un chatbot de IA que permite a los trabajadores obtener información de manera rápida y sencilla. Para el propósito de este caso de estudio, se implementó una versión aparte del SI de SST que incluye un conjunto mínimo de funcionalidades, protegiendo así la producción intelectual de la empresa. Esta versión permite evaluar la aplicación del framework y ayuda a los usuarios a familiarizarse con la integración del chatbot de SST.

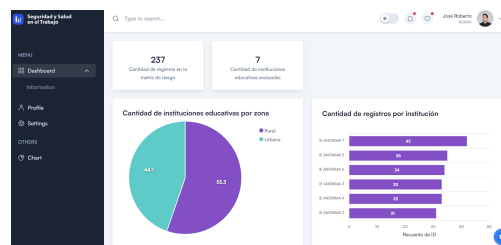


Figura 4.8: Interfaz gráfica del SI de SST

4.4.2.2. Flujo de orquestación de componentes

Es importante destacar que, durante el proceso de ADD (Attribute Driven Design), se aplicaron buenas prácticas, lineamientos, criterios y recomendaciones en la toma de decisiones de los diferentes componentes arquitectónicos, guiados por el Framework. No obstante, resulta fundamental examinar detalladamente el proceso de orquestación en la arquitectura RAG, y cómo el Framework orientó los ADR (Architectural Decision Record), especialmente en el flujo de interoperabilidad de estos componentes. Es esencial mostrar cómo la Inteligencia Artificial interopera con una fuente de datos en el sector de Seguridad y Salud en el Trabajo, como se ha discutido en el caso de estudio. Para ello, en la Figura 4.9 se ilustra el flujo de orquestación configurado para el caso de estudio, y se detallan en esta sección cada uno de los componentes del flujo de orquestación.

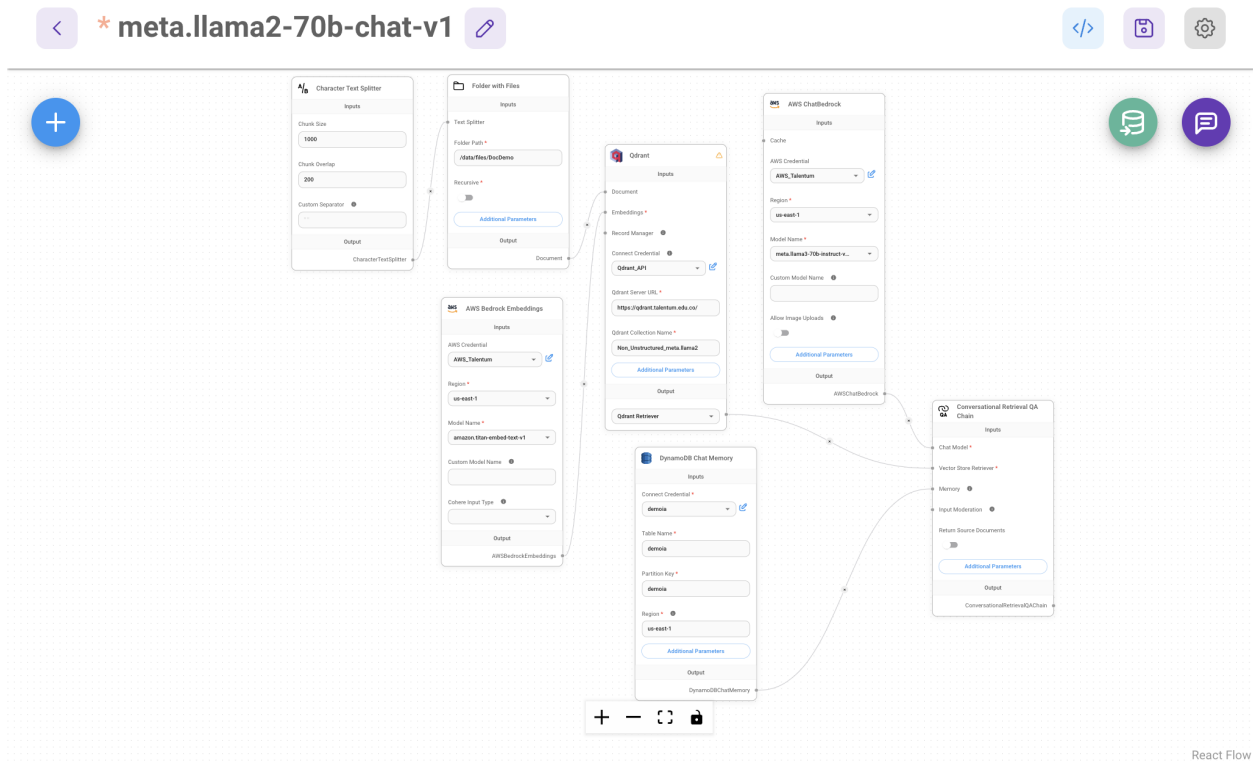


Figura 4.9: Chatflow desarrollado en Flowise

A continuación, se presentará el Chatflow, término utilizado por la herramienta para referirse a los flujos de trabajo creados. Se explicará cada componente del Chatflow, comenzando por el orquestador.

- Orchestration Component for LLM and AI Agents:** Para implementar la arquitectura RAG, se desplegó un flujo de orquestación a través del componente de orquestación para LLM

y agentes de IA, utilizando la herramienta Flowise, conforme a lo sugerido en el framework. Mediante esta herramienta de orquestación, se creó un “Chatflow”, que es el término que la herramienta utiliza para los flujos de trabajo generados. Dentro de estos Chatflows se realizan todos los procesos y conexiones necesarios entre los diferentes componentes para alcanzar el atributo de calidad de interoperabilidad y permitir la interacción entre distintos componentes de IA y la operatividad del chatbot de Inteligencia Artificial. Durante el desarrollo del caso de estudio, se configuró un Chatflow que facilita la interacción de los diversos componentes con configuraciones específicas, asegurando la implementación de un flujo de componentes que cubría los requisitos establecidos para el caso de estudio. Además, esta herramienta de orquestación facilita la interoperabilidad entre el sistema de información de Seguridad y Salud en el Trabajo y el chat de IA, a través de la generación de un componente Frontend en tecnología React, el cual se puede integrar debido a que Flowise proporciona el código necesario para su incrustación, logrando así integrar el chatbot de Inteligencia Artificial dentro del sistema de información de Seguridad y Salud en el Trabajo.

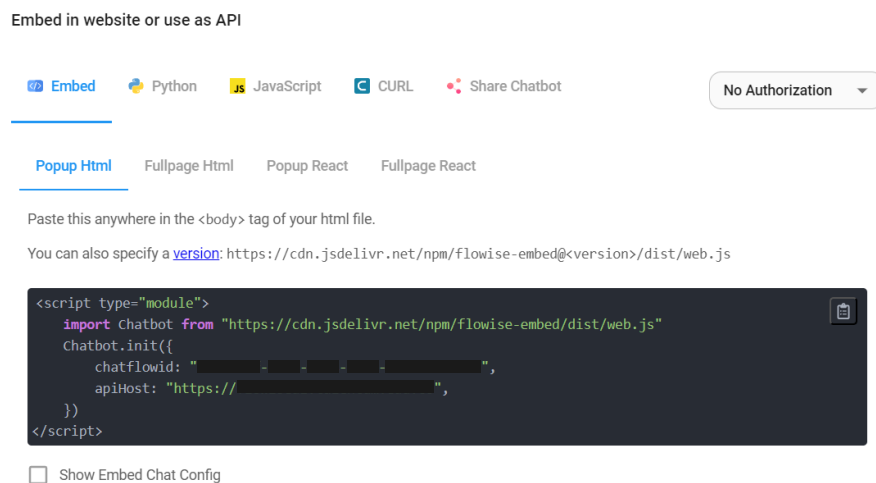


Figura 4.10: Código para embeber un chatbot de Inteligencia Artificial desarrollado en Flowise

- **File upload component:** El es el encargado de permitir la carga de archivos a través de una interfaz amigable para el usuario. Este componente hace referencia a la herramienta desplegada conocida como Filebrowser. Filebrowser es una herramienta que permite a los usuarios cargar diversos tipos de archivos mediante una interfaz.

Para la orquestación de los archivos cargados, el Chatflow incluye un componente denominado “Folder with Files”, a través del cual se especifica el lugar donde los archivos quedaron almacenados dentro de la máquina. El componente “Folder with Files” fue implementado con el objetivo de acceder a los archivos almacenados, que fueron previamente cargados mediante Filebrowser, permitiendo su utilización dentro de la ejecución del Chatflow.

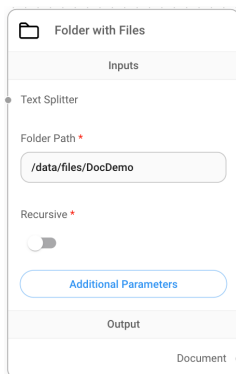


Figura 4.11: Componente “Folder with Files” desde Flowise

El componente “Folder with Files” dispone de tres parámetros: “Folder Path”, que es el campo donde se introduce la ruta de acceso a los archivos; “Recursive”, que es un botón tipo toggle que, al ser activado, habilita al componente para buscar archivos dentro de las carpetas en la ruta especificada; y “Additional Parameter”, que permite al desarrollador especificar metadatos de forma manual.

- **Data structure component:** Es utilizado para dividir texto en segmentos de tamaño especificado. Dentro de Flowise, se implementó el componente “Character Text Splitter”, que permite configurar visualmente los segmentos de la información que se procesará en los archivos almacenados. De esta manera, estos segmentos, al ser almacenados dentro de la base vectorial, no se guardan en su totalidad dentro de un vector único. Por medio de este componente, el contenido de los documentos se divide en varias partes, y así los datos quedan almacenados a través de diferentes vectores dentro de la base de datos vectorial.

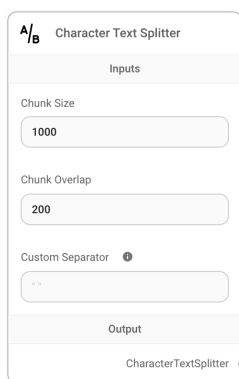


Figura 4.12: Componente “Character Text Splitter”

Este componente se implementó debido a que la base de datos vectorial tiene un límite máximo en cuanto al largo del documento que puede almacenar. Al realizar el proceso de fragmentación

de texto o “chunking”, cada parte se almacena por separado, evitando que se supere este límite. Además, aunque es posible ampliar la configuración de los vectores, un vector más grande puede resultar en una pérdida de precisión de la información en situaciones específicas. El componente incluye tres parámetros: “Chunk Size”, donde se especifica el número máximo de caracteres de cada segmento que se incluirá en un fragmento; “Chunk Overlap”, donde se define la cantidad de caracteres que pueden superponerse durante la partición; y “Custom Separator”, que permite al desarrollador especificar un carácter presente en el texto que se utilizará para realizar el “chunking”.

- Embeddings component:** Para la implementación del componente de embeddings, es importante mencionar que se utilizaron dos modelos de LLM durante el desarrollo del caso de estudio: el llama2-70b-chat-v1 y el gpt-3.5-turbo. Para cada modelo, se implementó un componente de embedding diferente, como se ilustra en la Figura 4.13, donde la caja izquierda ilustra el componente utilizado para el LLM de llama2-70b-chat-v1 y la caja derecha ilustra, el componente usado para el gpt-3.5-turbo. Es relevante destacar que estos dos componentes no operan simultáneamente; si uno está activo, el otro debe ser desactivado.

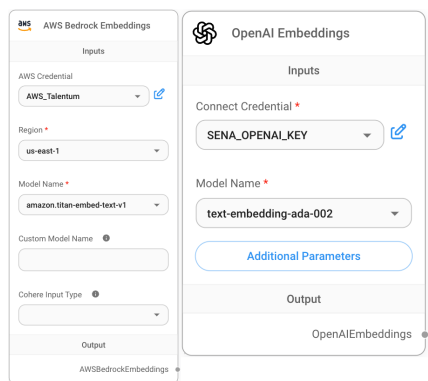


Figura 4.13: Componentes de embedding para gpt-3.5-turbo y llama2-70b-chat-v1

En el caso del modelo gpt-3.5-turbo, se utilizó el componente “OpenAI Embeddings”, que facilita el proceso de vectorización a través de un modelo de embedding desarrollado por la misma empresa que creó el modelo gpt-3.5-turbo, OpenAI. Este componente incluye tres parámetros: “Connect Credential”, donde se guardan las credenciales para conectarse a la API de OpenAI, para lo cual se creó previamente una cuenta en OpenAI de la cual se obtuvieron las credenciales de conexión para la API; “Model Name”, que permite seleccionar el modelo de embedding a implementar, que en este caso fue el modelo text-embedding-ada-002; y “Additional Parameters”, que permite al desarrollador configurar características adicionales para la ejecución del componente.

Por otro lado, en el caso del modelo llama2-70b-chat-v1, se utilizó el componente “AWS Bedrock Embeddings”, que también permite el proceso de vectorización mediante un modelo de

embedding desarrollado por Amazon. Este componente cuenta con cuatro parámetros: “AWS Credential”, que almacena las credenciales para conectarse a la API de AWS, operando con la cuenta de la Corporación Talentum, por lo que no se dejan registros de credenciales por razones de privacidad de la información, aunque se habilitaron temporalmente unos credenciales de conexión para la API; “Region”, que especifica la región en la cual se desplegó la API; “Model Name”, donde se selecciona el modelo de embedding a implementar a partir de una lista desplegable, siendo el modelo elegido amazon.titan-embed-text-v1; y “Custom Model Name”, donde el desarrollador puede digitar manualmente el nombre del modelo de embedding de AWS a implementar.

- **Knowledge Database Component:** El componente se encarga de almacenar información de las fuentes de datos seleccionadas para el caso de estudio, que en este caso están asociadas a Seguridad y Salud en el Trabajo. Esta información, conocida como conocimiento, proviene de especialistas de SST. Para su almacenamiento, se utiliza Qdrant como base de datos vectorial, y dentro del Chatflow, existe un componente llamado “Qdrant” para orquestar esta base de datos desde el flujo de orquestación, en la Figura 4.14 se muestra este componente.

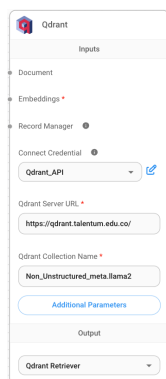


Figura 4.14: Componente de “Qdrant” dentro del Chatflow

Qdrant es la base de datos vectorial elegida para el desarrollo del caso de estudio, por lo que fue provisionada en la infraestructura de la Corporación Talentum. El componente de “Qdrant” del Chatflow facilita la interacción de la base de datos vectorial con el modelo de embedding mediante las conexiones generadas a través de la herramienta de orquestación, incluye tres parámetros: “Connect Credentials”, donde se almacenan las credenciales para conectarse a la API de Qdrant, obtenidas mediante la aplicación de Qdrant previamente desplegada; “Qdrant Server URL”, donde se introduce la URL donde se desplegó el servicio de Qdrant; y “Qdrant Collection Name”, donde se introduce el nombre de la colección de Qdrant donde se almacenarán los documentos vectorizados por el componente de embedding. Es importante mencionar que para este último parámetro, se puede introducir el nombre de una colección ya existente en Qdrant o de una que aún no existe, ya que, al momento de cargar los archivos, si esta no existe, la base de datos creará la colección automáticamente.

- Large Language Model Component:** Para el desarrollo del caso de estudio se utilizaron dos modelos: el llama2-70b-chat-v1 de Meta y el GPT de OpenAI. Dentro del Chatflow del orquestador, se emplearon los componentes correspondientes a estos modelos, los cuales se pueden observar en la Figura 4.15; la caja de la izquierda corresponde al modelo de llama2-70b-chat-v1 y la de la derecha al modelo de gpt-3.5-turbo. Es importante señalar que para el caso de estudio se puede usar solo una IA a la vez, por lo que solo puede estar activo un componente para evitar conflictos durante la ejecución. Por tal motivo, si se está utilizando uno, se debe eliminar temporalmente el otro.

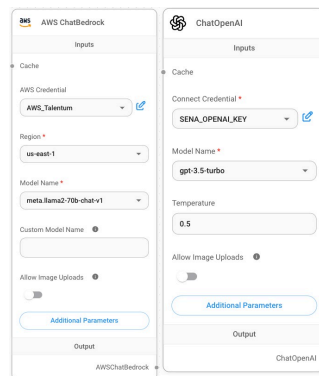


Figura 4.15: Componentes de LLM para llama2-70b-chat-v1 y gpt-3.5-turbo

En el caso de gpt-3.5-turbo, se implementó el componente “ChatOpenAI” dentro del flujo de orquestación, el cual permite el uso de diferentes modelos desarrollados por OpenAI. Este componente incluye cinco parámetros: “Connect Credential”, donde se utiliza la cuenta configurada en ese componente con los credenciales de la API de OpenAI; “Model Name”, donde se seleccionó el LLM a implementar, que para este caso fue el modelo gpt-3.5-turbo; “Temperature”, que permite al modelo definir qué tan precisas o creativas pueden ser las respuestas generadas; “Allow Image Uploads”, un botón tipo toggle que, al activarse, permite cargar imágenes al modelo para ciertos casos de uso; y “Additional Parameters”, que permite al desarrollador configurar características adicionales para la ejecución del componente.

Por otro lado, para el modelo llama2-70b-chat-v1, se implementó el componente “AWS ChatBedrock”, que facilita el uso de diferentes modelos desarrollados por Amazon y otros que son de carácter open source. Este componente se compone de seis parámetros: “AWS Credential”, donde, al igual que en el componente “AWS Bedrock Embeddings”, se han configurado los credenciales de la API de AWS en la cuenta de AWS de la Corporación Talentum para acceder a estos modelos desplegados de manera serverless; “Region”, que especifica la región en la que se desplegó la API; “Model Name”, donde se selecciona el modelo de embedding a implementar a partir de una lista desplegable, eligiendo el modelo meta.llama2-70b-chat-v1; “Custom Model Name”, donde el desarrollador puede digitar manualmente el nombre

del LLM de AWS a implementar; y “Additional Parameters”, que permite al desarrollador configurar características adicionales para la ejecución del componente.

- **Buffer Memory Component:** Para el desarrollo del caso de estudio, era necesario almacenar los chats generados por los usuarios. Para esto, se utilizó el componente “DynamoDB Chat Memory” dentro del Chatflow del orquestador, un componente de Flowise que facilitó la conexión con una base de datos DynamoDB de AWS, desplegada temporalmente para el caso de estudio. DynamoDB es una base de datos clave-valor que permite almacenar las conversaciones entre el chatbot y el usuario de manera eficiente, facilitando consultas rápidas.

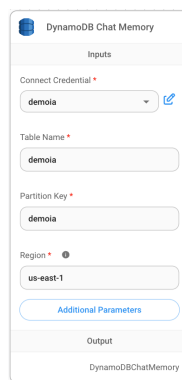


Figura 4.16: Componente del Chatflow para la base de datos DynamoDB

El componente “DynamoDB Chat Memory” gestiona la información que se almacena en la base de datos configurando ciertos parámetros. Este componente se compone de cuatro parámetros: “Connect Credential”, donde se almacenan las credenciales de la base de datos de AWS; “Table Name”, donde se especifica el nombre de la tabla en DynamoDB destinada al almacenamiento de las conversaciones; “Partition Key”, que es un campo configurado según la tabla de DynamoDB; “Region”, que identifica la región en la que se desplegó la base de datos; y “Additional Parameters”, que permite al desarrollador configurar características adicionales para la orquestación de la base de datos en el Chatflow.

- **LLMs Configuration and Execution Component:** El último componente implementado dentro del Chatflow es el “Conversational Retrieval QA Chain”, que configura la arquitectura RAG (Retrieval Augmented Generation). Este componente es esencial en el Chatflow, ya que inicia todo el proceso de ejecución del flujo de orquestación. A través de él, se gestiona la recuperación de información relevante de fuentes externas, en este caso, fuentes de conocimiento de Seguridad y Salud en el Trabajo, y facilita la generación de contenido basado en esa información.

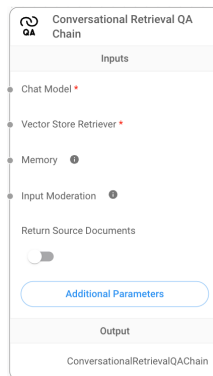


Figura 4.17: Componente “Conversational Retrieval QA Chain” del Chatflow para iniciar el flujo de orquestación

El “Conversational Retrieval QA Chain” incluye dos parámetros principales: “Return Source Documents”, un botón tipo toggle que, al activarse, permite que la respuesta generada incluya también la fuente de datos de la cual se extrajo la información; y “Additional Parameters”, donde se pueden establecer instrucciones adicionales que el modelo debe considerar para mejorar la precisión de las respuestas generadas. Estas instrucciones se configuraron durante el desarrollo del caso de estudio para optimizar el desempeño del modelo.

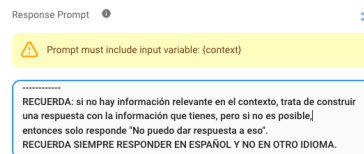


Figura 4.18: Instrucciones establecidas en los parámetros adicionales del componente “Conversational Retrieval QA Chain”

4.4.2.3. Flujo del prompt de SST

Antes de detallar los pasos del proceso del prompt, es relevante mencionar que en la sección anterior se introdujo el Chatflow configurado en el componente arquitectónico Flowise. A continuación, se describe cómo viaja el prompt a través de la arquitectura RAG que ha sido implementada. Se expone primero un flujo general del viaje del prompt para facilitar la comprensión de cada paso detallado posteriormente.

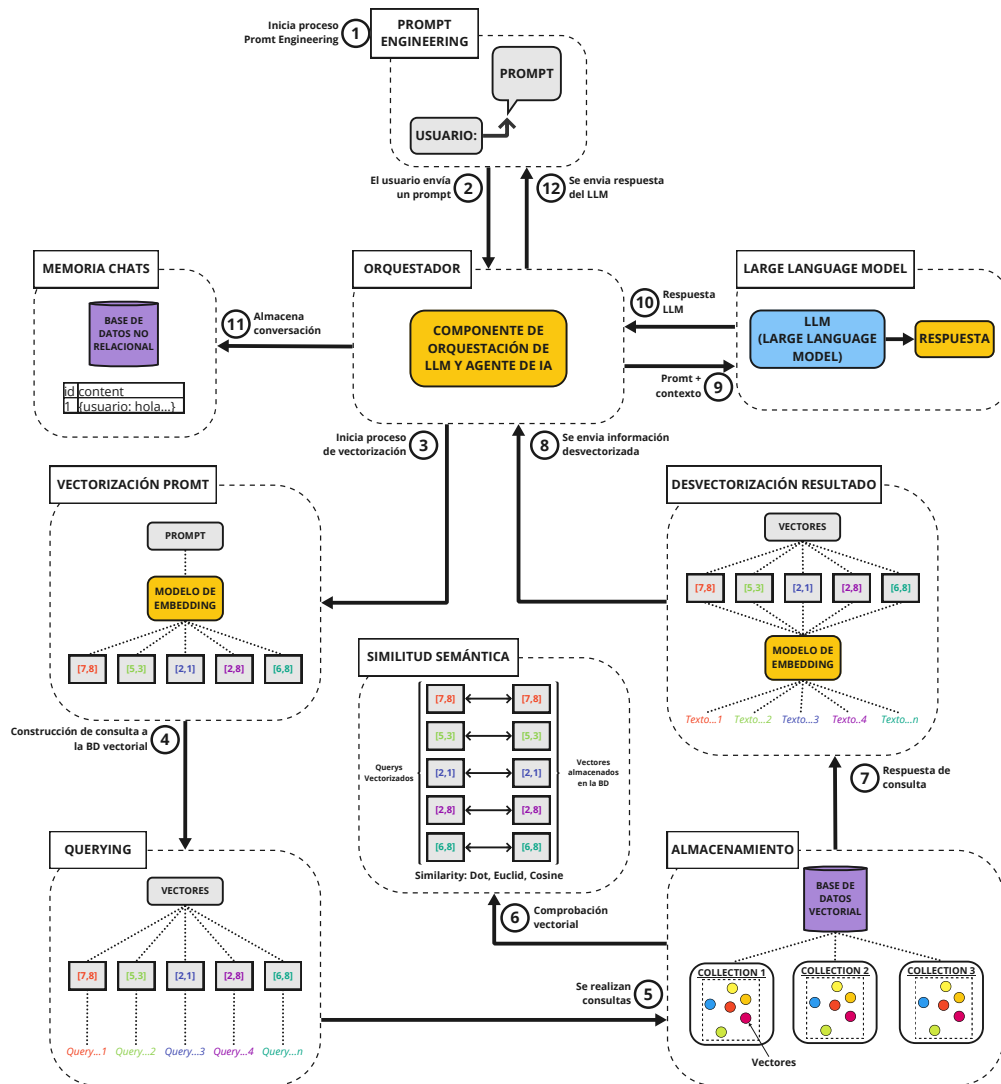


Figura 4.19: Diagrama de flujo de prompt de SST

1. Inicialmente, se establecen los criterios y las mejores prácticas del Framework en ingeniería de prompts, específicamente enfocados en Seguridad y Salud en el Trabajo. Esto se realiza para formular prompts más precisos para el flujo de orquestación, donde juega un papel vital el proceso de la arquitectura RAG. Al aplicar estas directrices, se garantiza que los prompts generados sean más adecuados y alineados con los objetivos específicos de SST requeridos por la Corporación Talentum.
2. Posteriormente, el usuario redacta un prompt y lo envía a través del chat de Inteligencia Artificial integrado en el sistema de información de Seguridad y Salud en el Trabajo. Este prompt es recibido por el componente orquestador, el cual es el núcleo del flujo de orquestación. Este

componente es esencial porque activa el proceso RAG, iniciando así la cadena de operaciones necesarias para el procesamiento y generación de respuestas adecuadas basadas en el prompt suministrado.

3. Una vez que el orquestador recibe el prompt, inicia el proceso de vectorización del mismo. Es importante destacar que el orquestador debe coordinar varios componentes, incluyendo la base de datos vectorial que para el caso de estudio se definió Qdrant. Las consultas a esta base de datos se realizan exclusivamente mediante vectores. Esta base de datos alberga una colección de datos relevantes proporcionados por especialistas en Seguridad y Salud en el Trabajo de la Corporación Talentum. Mediante vectores, se calcula la similitud entre el contenido del prompt introducido por el usuario y la información almacenada, con el objetivo de enriquecer el prompt con contexto adicional que mejore la precisión de las respuestas generadas por la Inteligencia Artificial. Este proceso de vectorización es esencial para realizar cálculos de similitud entre los vectores y optimizar la relevancia de la respuesta final.
4. Después de la vectorización del prompt, el siguiente paso implica la consulta a la base de datos para determinar qué vectores presentan una similitud significativa según ciertas métricas. Para este caso de estudio, se opta por utilizar la Similitud Coseno debido a su aptitud para manejar similitudes en el procesamiento de texto. Sin embargo, otras métricas también están disponibles y pueden utilizarse según las necesidades específicas:
 - **Dot Product (Producto Punto)**: Esta métrica calcula el producto escalar de dos vectores, siendo más alto cuando los vectores están más alineados en dirección.
 - **Euclidean Distance (Distancia Euclidiana)**: Esta mide la distancia convencional entre dos puntos en el espacio vectorial, donde distancias menores indican una mayor similitud.
 - **Cosine Similarity (Similitud Coseno)**: Mide el coseno del ángulo entre dos vectores, comúnmente utilizado en procesamiento de texto para evaluar cuán similares son dos documentos en cuanto a orientación, más allá de su magnitud.
5. Con la métrica de similitud que se menciona en el paso anterior, la base de datos de Qdrant prepara y ejecuta la consulta internamente. Este proceso implica ajustar la configuración de la consulta para utilizar específicamente la métrica definida, en este caso, la Similitud Coseno. La consulta se configura de manera que solo se consideren los vectores que correspondan a esta métrica de similitud, optimizando así la búsqueda para identificar aquellos vectores que mejor coincidan con el prompt vectorizado del usuario.
6. En este paso, la base de datos de Qdrant procesa la consulta de similitud y realiza el ranking de los resultados. Se enfoca en identificar y clasificar los cuatro vectores que presentan la mayor similitud con el prompt del usuario según la métrica de Similitud Coseno. Este ranking es crucial porque determina cuáles son los vectores más alineados con la consulta del usuario y, por tanto, los más aptos para ser utilizados en la formulación de la respuesta. Este proceso garantiza que solo los vectores más relevantes influyan en la generación de la respuesta final.

7. Con los vectores más similares, el siguiente paso es la desvectorización de esta información. Este proceso convierte los vectores en datos legibles y comprensibles que pueden ser incorporados como contexto adicional en el prompt que será enviado al modelo de lenguaje de gran escala (LLM). Aunque este proceso de desvectorización es automático y transparente en el Chatflow que se encuentra configurado en el Flowise, pero es importante resaltar que sin este paso no se puede incorporar la información como contexto al prompt.
8. Una vez que la información ha sido desvectorizada, se envía al orquestador. Este componente recibe los datos ya procesados y los prepara para incorporarlos como contexto adicional en el prompt original. Este paso es fundamental para enriquecer el contenido que será analizado y respondido por el modelo de lenguaje de gran escala (LLM), asegurando que la información relevante de Seguridad y Salud en el Trabajo sea considerada para generar respuestas más informadas y pertinentes.
9. El orquestador, al recibir la información enriquecida y desvectorizada, procede a añadir este contexto adicional al prompt. Este enriquecimiento del prompt es crucial para que el LLM tenga acceso a una base de datos contextual ampliada que le permita producir una respuesta más precisa y detallada. Una vez organizado el prompt con toda la información necesaria, el orquestador lo envía al LLM para su procesamiento.
10. El modelo de lenguaje de gran escala procesa el prompt enriquecido con el contexto adicional y genera una respuesta basada en esta información compuesta. Este paso culmina con el envío de la respuesta generada de vuelta al usuario a través del sistema de chat, cerrando el ciclo de interacción con una respuesta que busca ser lo más acertada y útil posible para las consultas realizadas.
11. Tras generar la respuesta el LLM, el orquestador no solo recibe esta información, sino que también se encarga de almacenarla junto con el prompt original en una base de datos DynamoDB. Este almacenamiento sirve múltiples propósitos: mantiene un registro histórico de las interacciones del chat, facilita el análisis de las respuestas previas, y permite al sistema recordar conversaciones, temas o respuestas anteriores en futuras sesiones con el mismo usuario. Este registro es fundamental para mejorar la continuidad y la personalización del servicio ofrecido.
12. Finalmente, después de que el chat se ha almacenado adecuadamente, el orquestador procede a enviar la respuesta al sistema de información de Seguridad y Salud en el Trabajo. Esta respuesta es devuelta específicamente al chat de Inteligencia Artificial, donde se muestra visualmente al usuario. Este último paso cierra el ciclo de interacción, asegurando que el usuario reciba una respuesta clara y visualmente accesible, lo cual completa el proceso de consulta y respuesta dentro de la arquitectura RAG.

4.4.2.4. Flujo de vectorización de fuentes de datos de SST

Es esencial destacar que el proceso de vectorización de las fuentes de datos, aunque se realice una sola vez, constituye un paso fundamental. En el contexto del caso de estudio, las fuentes de datos definidas deben someterse a vectorización. Aunque este procedimiento ocurre una sola vez tras la limpieza y estructuración de los datos, es vital recalcar que cualquier cambio o modificación en los datos requiere repetir la vectorización. Dentro del componente Flowise, esta actividad puede llevarse a cabo utilizando herramientas integradas, aunque también se puede gestionar externamente si el volumen de datos es considerablemente grande. A continuación, se ilustra un diagrama que muestra el flujo de este proceso, seguido por una explicación breve de los pasos involucrados.

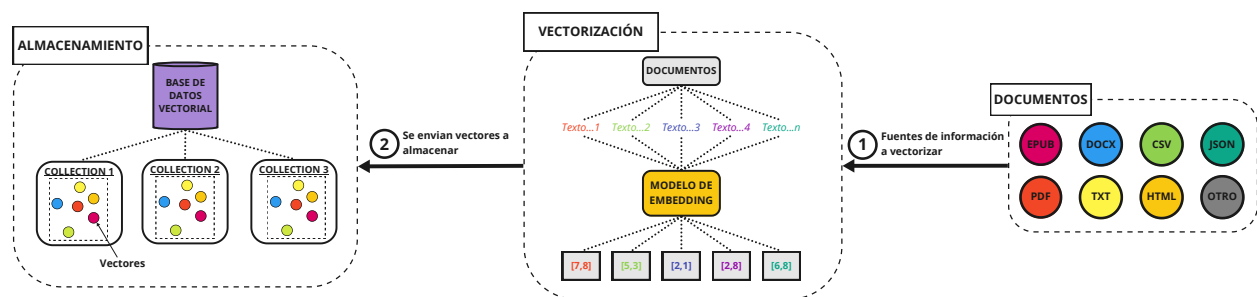


Figura 4.20: Diagrama de flujo de vectorización de fuentes de datos de SST

Antes de comenzar, es importante mencionar que los datos deben haber sido previamente filtrados en un proceso de limpieza, conservando únicamente la información pertinente para almacenar en la base de datos vectorial, que para el caso de estudio se utiliza Qdrant. Dado que se utiliza Flowise para cargar estos datos, es necesario emplear el componente Filebrowser para subir los archivos a una ruta específica configurada en Flowise, permitiendo el acceso a los mismos. Las configuraciones de estos componentes se explicaron anteriormente en el flujo de componentes. Para proporcionar un contexto más detallado sobre lo que ocurre durante la vectorización, se describen los pasos a seguir:

1. Organizar las fuentes de datos en una ubicación accesible que facilite la aplicación del proceso de vectorización. Es fundamental garantizar que los datos estén ordenados y disponibles para procesar eficientemente sin interrupciones.
2. Aplicar la vectorización utilizando modelos de embeddings para transformar la información. Según el tipo de formato de los datos, se ajusta la vectorización para asegurar que la información vectorial sea consistente y esté adecuadamente formateada para consultas futuras.
3. Una vez vectorizada la información, se almacena en la base de datos de Qdrant, específicamente en la colección o colecciones destinadas para su uso en el proceso de Arquitectura RAG. Este almacenamiento está diseñado para facilitar la aplicación de métricas de similitud, mejorando la precisión al extraer información.

4.4.3. Aplicación de ingeniería de prompt en SST

Según se detalla en el Framework propuesto, la ingeniería de prompt constituye el proceso esencial para diseñar y optimizar las consultas de entrada que se introducen en los modelos de lenguaje de gran escala (LLMs), con el objetivo de obtener respuestas más precisas y adecuadas. Para implementar la ingeniería de prompt de manera efectiva, se organizó un encuentro con especialistas en Salud y Seguridad en el Trabajo (SST). En dicho encuentro, se expuso el concepto de ingeniería de prompt, se explicaron los distintos tipos de prompts que se pueden utilizar y se ofrecieron criterios específicos para diseñar prompts de manera adecuada. Además, durante esta sesión, los especialistas de SST tuvieron la oportunidad de interactuar con una demostración del chatbot y aplicar los conocimientos adquiridos sobre ingeniería de prompt.

4.4.3.1. Proceso aplicado de Prompt Engineering en SST en la Corporación Talentum

El proceso aplicado de ingeniería de prompt en el ámbito de la Salud y Seguridad en el Trabajo (SST) en la Corporación Talentum se efectuó mediante la implementación de los conceptos y buenas prácticas descritas en el Framework.

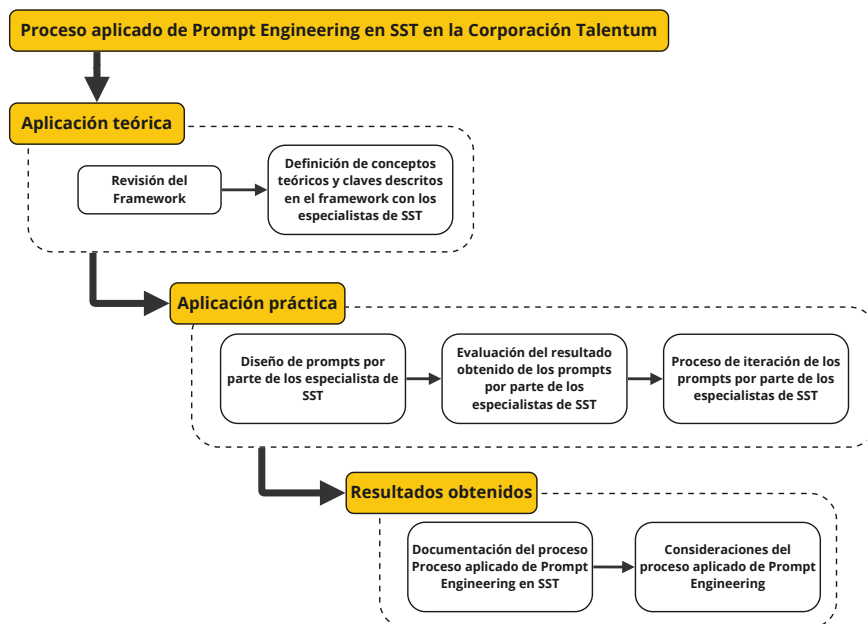


Figura 4.21: Diagrama del proceso aplicado de Prompt Engineering en SST en la Corporación Talentum

El proceso de ingeniería de prompt se estructuró en dos fases distintas con los especialistas de SST. La primera fase consistió en una parte teórica, necesaria para definir y comprender junto a los especialistas los conceptos y elementos clave del framework relacionados con la ingeniería de prompt. La segunda fase tuvo un carácter práctico. A continuación, se detallan los conceptos

teóricos de ingeniería de prompt descritos en el Framework que se aplicaron en la primera parte del proceso:

- Se organizó una sesión formativa en la cual se instruyó a los especialistas de SST sobre la definición y aplicabilidad de la ingeniería de prompt, destacando su importancia para lograr resultados superiores con los modelos de lenguaje de gran escala (LLM).
- Durante esta sesión, se trataron los pasos de definición de objetivos, diseño del prompt, evaluación de la respuesta e iteración, descritos en el framework como las etapas esenciales del proceso de ingeniería de prompt.
- También se explicaron los criterios para el diseño de prompts, tales como la claridad y la concisión, la relevancia del contexto, la especificidad, la inclusión de palabras clave, la longitud y complejidad adecuadas, instrucciones claramente definidas, prompts segmentados, la consistencia, la adaptabilidad a tareas concretas, la facilidad para evaluar respuestas, la variación y experimentación, y por último, el reconocimiento de las limitaciones inherentes al modelo.

Para la segunda parte, como ya se mencionó anteriormente, se realizó una sesión práctica. En esta sesión, los especialistas de SST pudieron poner a prueba los conocimientos adquiridos sobre ingeniería de prompt mediante su interacción con el prototipo de chatbot de IA de SST. A continuación, se presenta un listado de algunos de los diferentes tipos de prompts descritos en el framework que fueron aplicados por los especialistas, junto con las respuestas obtenidas por parte del prototipo de chatbot de IA de SST:

- **Prompts cerrados y/o abiertos:** Recordando que un prompt cerrado es aquel que define un límite exacto para la longitud de la respuesta, se verificó que el prototipo de chatbot de IA está en la capacidad de responder a este tipo de prompts. Por ejemplo, cuando uno de los especialistas preguntó por la fecha exacta de un informe, el chatbot respondió sin problemas.

USUARIO: En qué fecha se redactó el informe de SST?
MODELO: El informe de SST fue redactado el 5 de diciembre de 2022.

Tabla 4.9: Prompt “Closed” generado por especialista de SST

De manera similar, también se confirmó la capacidad del prototipo de chatbot de IA para manejar los prompts abiertos, dado que cuando uno de los especialistas de SST formuló una pregunta sin una delimitación específica para la respuesta, el chatbot fue capaz de generar una respuesta del largo que consideró pertinente.

USUARIO: Qué me puedes contar sobre el tema de SST?
Continúa en la siguiente página

<p>MODELO: Hola, me llamo Future y estoy aquí para ayudarte. En cuanto a tu pregunta, puedo decirte que la Seguridad y Salud en el Trabajo es un tema muy importante en cualquier entorno laboral. Se refiere a la implementación de medidas y políticas orientadas a prevenir accidentes laborales, enfermedades ocupacionales y promover un entorno laboral seguro y saludable. Esto abarca una amplia gama de aspectos, desde la identificación y evaluación de riesgos hasta la capacitación del personal en prácticas seguras y el acceso a servicios médicos adecuados. ¿Quieres saber más sobre algún aspecto específico de SST?</p>
--

Tabla 4.10: Prompt “Opened” generado por especialista de SST

- **Flipped Interaction (Interacción invertida):** Este prompt, que propone que sea el chatbot de IA quien formule las preguntas al usuario, también fue evaluado durante el proceso de ingeniería de prompt por los especialistas de SST. Para esto, uno de los especialistas introdujo un prompt solicitando al modelo que sugiriera tres preguntas sobre el tema de SST.

<p>USUARIO: Sugiereme tres preguntas sobre el tema de SST</p>
--

<p>MODELO:</p>

- | |
|---|
| <ol style="list-style-type: none"> 1. ¿Cómo puede determinar si una empresa cumple con las normas de seguridad y salud en el trabajo? 2. ¿Cómo se pueden prevenir accidentes laborales y enfermedades ocupacionales en una organización? 3. ¿Cuál es el papel de la alta dirección en la implementación del Sistema de Gestión de Seguridad y Salud en el Trabajo? |
|---|

Tabla 4.11: Prompt “Flipped Interaction” generado por especialista de SST

Se observó que el chatbot de IA generó las preguntas solicitadas por el especialista sin inconvenientes, aunque es importante mencionar que estas preguntas se basaron en el contexto proporcionado por los especialistas de SST durante la sesión, lo cual le dio suficiente información para formular preguntas relacionadas con el tema, aunque no necesariamente indicaba que poseía el conocimiento para responder a dichas preguntas.

- **Meta Language Creation (Creación del lenguaje meta):** Este tipo de prompt se utiliza para especificar algunos términos dentro del lenguaje del prompt que pueden ser confusos o ambiguos. En su aplicación, los especialistas optaron por clarificar el significado de las siglas “SST”, las cuales pueden tener más de un significado.

USUARIO: Cuando yo diga SST me refiero a seguridad y salud en el trabajo.
MODELO: Entendido, ¿qué puedo ayudarte con SST?

Tabla 4.12: Prompt “Meta Language Creation” generado por especialista de SST

- **Output Automater (Automatización de salida):** Mediante este prompt se pueden automatizar procesos manuales de generación de texto. Para verificar la respuesta del prototipo de chatbot de IA frente a este tipo de prompts, uno de los especialistas solicitó al chatbot que generara una lista de instituciones que aparecían en la matriz de riesgos y que finalizara cada elemento con los caracteres: “- Informe1”.

USUARIO: Necesito una lista de instituciones que aparecen en la matriz de riesgos y terminar cada elemento con los siguientes caracteres: “- informe1”
MODELO: Instituciones ANÓNIMA 1- Informe1 ANÓNIMA 2- Informe1 ANÓNIMA 3- Informe1 ANÓNIMA 4- Informe1 ANÓNIMA 5- Informe1 ANÓNIMA 6- Informe1 ANÓNIMA 7- Informe1

Tabla 4.13: Prompt “Output Automater” generado por especialista de SST

- **Persona:** El prompt de persona permite al modelo asumir una identidad; en este caso, un especialista planteó al modelo que fingiera ser un coordinador de SST.

USUARIO: Quiero que finjas que eres un coordinador de SST y me hagas una pregunta que él haría.
MODELO: ¿Cómo se puede medir el nivel de riesgo en una institución educativa y cuáles son las acciones preventivas y de control que se pueden implementar para disminuir el riesgo y mejorar la seguridad y salud en el trabajo?

Tabla 4.14: Prompt “Persona” generado por especialista de SST

El resultado fue que el prototipo de chatbot de IA asumió el rol sin problemas, generando preguntas que estaban relacionadas con el tema de SST y que parecían adecuadas para un coordinador de dicha área.

- **Alternative Approaches (Enfoque alternativo):** Este tipo de prompt permite solicitar al modelo sugerencias sobre cómo mejorar ciertos procesos. Para verificar la funcionalidad de este prompt, uno de los especialistas de SST pidió al modelo que sugiriera una manera de organizar mejor la matriz de riesgos.

<p>USUARIO: Sugíereme los pasos que hay que seguir para de alguna forma mejorar la organización de la matriz de riesgos.</p>
<p>MODELO:</p> <ol style="list-style-type: none"> 1. Adoptar las recomendaciones con respecto a la matriz de peligros y riesgos en el plan de trabajo anual del SG-SST. 2. Establecer y disponer desde la alta dirección, los recursos humanos, financieros, materiales, técnicos, tecnológicos necesarios para el desarrollo de los controles dados a conocer. 3. Socializar los resultados del informe matriz de peligros y riesgos a la alta dirección y el vigía ocupacional o el comité paritario de seguridad y salud en el trabajo (COPASST), según aplique. 4. Implementar medidas de control y corrección para los peligros identificados en el Nivel de Riesgo Tipo I y Tipo II. 5. Realizar un seguimiento y monitoreo de los riesgos identificados para evaluar la efectividad de las medidas implementadas. 6. Revisar y actualizar la matriz de riesgos periódicamente para asegurar que se estén abordando los riesgos más importantes y actualizados. 7. Capacitar y concienciar a los trabajadores sobre los riesgos identificados y las medidas de control y corrección implementadas. 8. Establecer un sistema de reporte y seguimiento de incidentes y accidentes para identificar oportunidades de mejora. 9. Realizar un análisis de costos y beneficios para justificar la intervención y su rentabilidad en los peligros identificados en el Nivel de Riesgo Tipo III. 10. Implementar un sistema de gestión de seguridad y salud en el trabajo que abarque todos los procesos y actividades de la organización. <p>Espero que estos pasos te ayuden a mejorar la organización y la matriz de riesgos.</p>

Tabla 4.15: Prompt “Alternative Approaches” generado por especialista de SST

Como resultado, el chatbot de IA propuso una serie de parámetros que podrían seguirse para optimizar el proceso de organización de la matriz de riesgos de SST, demostrando la capacidad del modelo para ofrecer respuestas útiles a este tipo de prompts.

- **Fact Check List (Verificación de hechos):** Este tipo de prompt permite validar la respuesta generada por el modelo solicitando los hechos que respaldan dicha respuesta. Para verificar la funcionalidad de este prompt, uno de los especialistas solicitó al modelo que evaluara la situación de SST de una institución y que fundamentara dicha evaluación.

USUARIO: En una escala del 1 al 5, siendo 1 la nota más baja y 5 la nota más alta, cómo calificarías la situación de SST para la IE ANÓNIMA 1 y con qué hechos respaldas tu respuesta?

MODELO: Calificaría la situación de SST para la IE ANÓNIMA 1 con un 2, ya que se han identificado peligros y riesgos en los procesos administrativo y operativo, pero no se han implementado controles efectivos para mitigarlos.

En el proceso administrativo, se ha identificado un peligro físico relacionado con el ruido del timbre de cambio de clase, que puede generar daños leves. Aunque se ha evaluado el nivel de riesgo como III, lo que significa que es necesario mejorar la situación, no se han implementado controles efectivos para reducir el ruido.

En el proceso operativo, se ha identificado un peligro biomecánico relacionado con el movimiento repetitivo al manejar la puerta reja, que puede generar daños graves. Aunque se ha evaluado el nivel de riesgo como II, lo que significa que es necesario corregir y adoptar medidas de control de inmediato, no se han implementado controles efectivos para reducir el riesgo.

En general, la situación de SST en la IE ANÓNIMA 1 es deficiente, ya que no se han implementado controles efectivos para mitigar los peligros y riesgos identificados. Es necesario que la institución educativa implemente medidas de prevención y control para reducir los riesgos y mejorar la situación de SST.

Tabla 4.16: Prompt “Fact Check List” generado por especialista de SST

Como resultado, se observó que el prototipo de chatbot de IA de SST fue capaz de proporcionar los hechos encontrados en el informe que le permitieron formular una evaluación sobre la situación de SST de la institución.

- **Refusal Breaker (Rompedor de rechazos):** A través de este prompt, se establece al modelo una acción a seguir en caso de que no sea capaz de responder a una pregunta. Para confirmar la capacidad del prototipo de chatbot de IA frente a este tipo de prompts, uno de los especialistas de SST introdujo un prompt preguntando por un tema no relacionado con el campo de SST.

USUARIO: Cómo puedo preparar una crema de champiñones? Si no puedes responder esta pregunta, entonces sugiéreme una pregunta que si puedas responder.

MODELO: Hola, me llamo Future y estoy aquí para ayudarte. Lo siento, pero no puedo responder a tu pregunta sobre cómo preparar una crema de champiñones, ya que mi área de especialización es el análisis de salud y seguridad en el trabajo para instituciones educativas. Sin embargo, puedo ayudarte con preguntas relacionadas con la seguridad y salud en el trabajo, como por ejemplo, “¿Cuáles son los peligros más comunes en un lugar de trabajo?” o “¿Cómo puedo identificar y evaluar los riesgos en mi lugar de trabajo?”. ¿Te gustaría hacerme una pregunta en este sentido?

Tabla 4.17: Prompt “Refusal Breaker” generado por especialista de SST

Se observó que el chatbot de IA sugirió varias preguntas que sí era capaz de responder, siguiendo las indicaciones propuestas por el usuario al introducir el prompt, ya que no pudo responder la pregunta original planteada por el usuario.

4.4.3.2. Consideraciones del proceso aplicado de Prompt Engineering

A partir de los diferentes resultados obtenidos mediante las interacciones de los especialistas de SST con el prototipo del chatbot de IA, desarrollado a partir del caso de estudio, se observó que el prototipo respondía adecuadamente a los diversos tipos de prompts establecidos, gracias a varios lineamientos del Framework. Es esencial señalar que, para alcanzar estos resultados, fue primordial la capacitación inicial de los especialistas de SST sobre el proceso de ingeniería de prompt. Esta formación se basó en la información propuesta por el Framework sobre ingeniería de prompt de SST, donde los especialistas comenzaban por definir un objetivo, luego diseñaban un prompt y, conforme el chatbot generaba una respuesta, procedían a evaluar los resultados obtenidos.

Cabe señalar que en varias ocasiones, los resultados no fueron los esperados por los especialistas de SST, dado que, aunque se mostró que el modelo lograba responder satisfactoriamente a los prompts, esto se debió a la necesidad de un ejercicio de iteración por parte de los especialistas, quienes tuvieron que rediseñar el prompt varias veces hasta obtener el resultado deseado. Este proceso de iteración se aplicó conforme a las buenas prácticas de la ingeniería de prompt y según lo mencionado en el Framework, ya que forma parte de los pasos esenciales dentro de la ingeniería de prompt para alcanzar los mejores resultados.

Es relevante destacar que, para alcanzar resultados superiores en el proceso de aplicación de ingeniería de prompt durante la sesión con los especialistas de SST, se consideró también la aplicación de varios criterios para el diseño de prompts, como la claridad y concisión al redactar el prompt, la especificación y delimitación adecuada del contexto dentro del prompt y la definición precisa de las instrucciones. Al implementar estos criterios, se facilitó el proceso de iteración permitiendo una

mejor redacción de los prompts.

Finalmente, como último aspecto a considerar dentro de los criterios y buenas prácticas aplicados al proceso de ingeniería de prompt durante la sesión con los especialistas de SST, fue relevante tener en cuenta las limitaciones del modelo y de los datos cargados. Además, se destacó que el prototipo de chatbot de IA desarrollado para esta demostración era simplemente una herramienta para facilitar el trabajo en ciertas tareas dentro del campo de SST, y no significaba que tuviera la capacidad de responder a todos los tipos de prompts ingresados ni que pudiera realizar una infinidad de tareas de forma automática.

4.5. Pruebas y Validación

Para el desarrollo de esta sección, se emplearon dos artefactos de evaluación de vital importancia: la Evaluación de Arquitectura Ligera (LAE) y la evaluación realizada por especialistas en Seguridad y Salud en el Trabajo. Estos métodos son cruciales para garantizar la calidad y el cumplimiento de los objetivos y requisitos planteados en el caso de estudio y el diseño arquitectónico. Cada uno ofrece una perspectiva única y valiosa, facilitando la identificación y comprensión de diversas opiniones que enriquecen y fortalecen el desarrollo del proyecto piloto derivado del caso de estudio.

4.5.1. Evaluación de Arquitectura Ligera (LAE)

La Evaluación de Arquitectura Ligera (LAE, por sus siglas en inglés) es un método diseñado para implementarse en un contexto interno de un proyecto, donde la revisión se lleva a cabo regularmente por pares. Esta técnica se basa en los mismos principios que el método ATAM (Architecture tradeoff analysis method), pero está adaptada para ser más ágil y menos formal, permitiendo evaluaciones frecuentes y focalizadas en cambios específicos o áreas no examinadas previamente de la arquitectura. La flexibilidad de LAE la convierte en una herramienta ideal para proyectos que requieren revisiones continuas sin la sobrecarga de procesos más formales y extensos.

En este contexto, el método LAE se aplicó al caso de estudio de la Corporación Talentum, una organización que enfrenta el desafío de integrar soluciones de Inteligencia Artificial en sus sistemas de Seguridad y Salud en el Trabajo. Posteriormente, se describe cómo se adaptó y ejecutó el proceso de LAE para evaluar y mejorar la arquitectura de software en la Corporación Talentum, destacando la importancia de la interoperabilidad y el rendimiento en el contexto de sus objetivos empresariales y tecnológicos específicos. Seguidamente, se explica detalladamente el proceso de aplicación de LAE en este caso particular, cubriendo los pasos críticos y las decisiones tomadas durante la evaluación.

4.5.1.1. Pasos Aplicados de LAE

A continuación, se describen los pasos seguidos en la Evaluación de Arquitectura Ligera (LAE), aplicados para examinar la arquitectura sugerida después de implementar el Attribute-Driven Design (ADD). Esta evaluación permitió revisar junto al equipo de desarrolladores los registros de

decisiones arquitectónicas (ADR). Es esencial detallar cómo se organizó el proceso antes de proceder a explicar cada paso en particular.

Para este ejercicio, se convocó a tres desarrolladores con experiencia intermedia, quienes han sido formados por la Corporación Talentum y han participado en los proyectos más significativos de la organización. Aunque no son seniors, estos desarrolladores aportan un gran valor gracias a su experiencia y aprendizaje continuo. La invitación para participar en este proceso se realizó con una semana de antelación, proporcionándoles el caso de estudio y la documentación completa del Attribute-Driven Design, incluyendo todos los ADR. Esto les permitió prepararse adecuadamente, revisando la documentación y explorando herramientas, librerías, estilos arquitectónicos y tecnologías que podrían contribuir al proceso de LAE.

La reunión se llevó a cabo en una sala y se extendió por aproximadamente cuatro horas, durante las cuales se aplicó la metodología LAE, utilizando el Framework como guía para evaluar el diseño arquitectónico. Con este contexto establecido, se procede a presentar cada paso y cómo se implementó durante la sesión.

1. Paso 1: Convocatoria y Preparación del Equipo de Desarrolladores

La sesión de LAE fue iniciada mediante la convocatoria de un equipo selecto de desarrolladores dentro de la organización. Antes de la reunión, se compartió con ellos documentación importante que incluía detalles sobre la aplicación del Diseño Guiado por Atributos (Attribute-Driven Design, ADD) y los Registros de Decisiones Arquitectónicas (Architecture Decision Records, ADR). Este material fue distribuido con el objetivo de que los desarrolladores llegaran preparados y contextualizados sobre el caso de uso específico a tratar.

Aunque el equipo no está especializado en estos temas, para la Corporación Talentum es fundamental ofrecer estos espacios de revisión. Participar en estos ejercicios permite a los desarrolladores aprender y familiarizarse con diversas problemáticas, fomentando su crecimiento profesional dentro de la organización. Para esta sesión, se seleccionaron tres desarrolladores considerados los más experimentados y capaces de aportar mayor valor a la revisión.

Además, se les explicó previamente el proceso de LAE para asegurarse de que todos los participantes comprendieran cómo funcionaría el ejercicio y cuál sería su rol durante la sesión. Esta preparación inicial busca maximizar la eficacia del encuentro y garantizar que todos los asistentes puedan contribuir de manera significativa al análisis y discusión de la arquitectura.

2. Paso 2: Revisión de los Objetivos de Negocio

En el contexto de la sesión de Evaluación de Arquitectura Ligera (LAE) en la Corporación Talentum, el segundo paso se centra en revisar los objetivos de negocio cruciales para el éxito del proyecto. Esta revisión asegura que todos los participantes, incluidos arquitectos y desarrolladores, estén alineados y comprendan las prioridades comerciales que guían el desarrollo del sistema de Seguridad y Salud en el Trabajo integrado con Inteligencia Artificial.

- **Costos:** Se enfatiza la importancia de la eficiencia económica, priorizando el uso de tecnologías de código abierto siempre que sea posible. Esta estrategia no solo reduce los costos directos asociados con el desarrollo del software, sino que también aumenta la flexibilidad y adaptabilidad del proyecto. Al optar por soluciones de código abierto, el equipo puede aprovechar un ecosistema amplio y dinámico de herramientas y librerías, lo que facilita la personalización y escalabilidad del sistema conforme a las necesidades cambiantes del proyecto.
- **Rendimiento:** El rendimiento es un objetivo crítico, particularmente en las funcionalidades que integran componentes de Inteligencia Artificial, como el Chat IA. Se establece como prioritario que el sistema maneje las solicitudes de los usuarios con baja latencia, garantizando una interacción rápida y fluida. Durante la revisión, se discute cómo las decisiones arquitectónicas pueden influir directamente en los tiempos de respuesta del sistema, buscando optimizar tanto el rendimiento como la experiencia general del usuario.
- **Interoperabilidad:** La interoperabilidad se identifica como el desafío principal, especialmente en la integración entre el sistema existente de SST y los nuevos componentes de Inteligencia Artificial. Se analiza la necesidad de implementar soluciones que interfieran mínimamente con la arquitectura preexistente, manteniendo la estabilidad y funcionalidad del entorno operativo. Los participantes examinan cómo diferentes enfoques arquitectónicos pueden facilitar o dificultar la integración, asegurando que el sistema resultante sea cohesivo y capaz de comunicarse eficientemente entre sus distintos módulos y servicios.

Este paso es fundamental para que todos los miembros del equipo tengan claridad sobre las metas comerciales que deben guiar las decisiones técnicas durante el desarrollo y la integración de la solución de IA en el ámbito de la SST. Al finalizar esta revisión, el equipo se encuentra mejor preparado para abordar los siguientes pasos de la evaluación con una comprensión compartida de lo que se necesita lograr para alinear la arquitectura del software con los objetivos estratégicos de la Corporación Talentum.

3. Paso 3: Revisión de la Arquitectura

Dado que la arquitectura fue entregada previamente y se socializó el proceso de diseño orientado por atributos (Attribute-Driven Design) con los desarrolladores, únicamente se realizó una breve presentación de los diagramas C4, específicamente los de contexto y contenedores. Este repaso rápido se llevó a cabo para recordar a los participantes los aspectos clave de la discusión, permitiendo así ahorrar tiempo y enfocarse más en la fase de ideación. La presentación facilitó que los desarrolladores, quienes ya habían analizado la arquitectura previamente, pudieran concentrarse en aportar ideas y evaluar los cambios recientes de manera eficiente.

4. Paso 4: Revisar los enfoques arquitectónicos

Durante la sesión de Evaluación de Arquitectura Ligeras (LAE) en la Corporación Talentum, el cuarto paso se centra en la revisión de los enfoques arquitectónicos empleados para abordar

las preocupaciones específicas relacionadas con la calidad y la funcionalidad del sistema. En este paso, se examina detenidamente la adopción de la Arquitectura RAG y la arquitectura basada en servicios, ambas importantes para el desarrollo del caso de estudio de integración de Inteligencia Artificial en el sector de Seguridad y Salud en el Trabajo.

- **Presentación de la Arquitectura RAG:** Un punto destacado en la reunión es la presentación del ADR (Architectural Decision Record) titulado “ADR-INT-03 Arquitectura de referencia para extracción de información”. Esta arquitectura permite a los modelos de lenguaje de máquina (LLM) buscar información relevante en bases de datos o conjuntos de datos externos y generar texto nuevo a partir de esta, empleando técnicas de aprendizaje automático o plantillas. Se decide implementar la Arquitectura RAG dado que mejora significativamente la precisión, relevancia e informatividad del texto generado por los LLM. Las consecuencias de esta decisión incluyen una generación de texto más precisa y relevante, así como modelos de LLM más creativos, capaces de ofrecer nuevas ideas y perspectivas en la generación de respuestas.
- **Evaluación de la Arquitectura Basada en Servicios:** Además, se analiza el estilo arquitectónico basado en servicios, que es un híbrido del estilo de microservicios y es valorado por su flexibilidad arquitectónica. Esta arquitectura se describe como menos compleja y costosa que otras arquitecturas distribuidas, lo que la hace popular en muchas aplicaciones empresariales. La topología de esta arquitectura sigue una estructura en capas macro distribuida, con interfaces de usuario, servicios de granularidad gruesa (servicios de dominio) y una base de datos monolítica, todos implementados de manera independiente. Se discute cómo la escalabilidad, la tolerancia a fallos y el rendimiento pueden gestionarse mediante múltiples instancias de un servicio de dominio, con equilibrio de carga para garantizar que la interfaz de usuario se conecte a una instancia saludable y disponible.

Esta revisión de los enfoques arquitectónicos no solo ayuda a los participantes a comprender las opciones disponibles y sus implicaciones, sino que también facilita la toma de decisiones informadas sobre cómo mejorar la integración de la IA en los sistemas existentes de SST, garantizando así que la arquitectura propuesta alineada con los objetivos empresariales y tecnológicos de la Corporación Talentum.

5. Paso 5: Revisión del Árbol de Utilidad de Atributos de Calidad

En la Corporación Talentum, el paso de revisión del árbol de utilidad de atributos de calidad es importante para alinear al equipo en torno a los objetivos arquitectónicos significativos del proyecto. Antes de esta sesión, se espera que todos los participantes hayan revisado detalladamente la documentación del diseño impulsado por atributos (Attribute Driven Design, ADD), incluyendo los escenarios ya documentados. Este conocimiento previo es esencial para facilitar una discusión informada y productiva durante la sesión de Evaluación de Arquitectura Ligera (LAE).

Durante la sesión, se presenta un árbol de utilidad de atributos de calidad, que destaca los atributos clave y pone especial énfasis en los Requisitos Arquitectónicamente Significativos (ASR, por sus siglas en inglés). Este árbol es una herramienta visual que ayuda a los participantes a comprender cómo los diversos componentes de la arquitectura contribuyen a los objetivos generales de calidad del sistema.

El árbol también sirve para verificar que los atributos priorizados estén alineados con las metas y expectativas del negocio, facilitando así la revisión y, si es necesario, la actualización de los escenarios de calidad. La discusión se centra en validar o modificar estos escenarios en función de las prioridades y riesgos identificados, asegurando que la arquitectura propuesta pueda satisfacer las exigencias del entorno operativo actual y futuro.

La efectividad de este paso depende en gran medida de la preparación previa de los participantes y de su familiaridad con los principios del diseño impulsado por atributos. Al asegurar que todos los miembros del equipo estén adecuadamente preparados y tengan un entendimiento profundo de la arquitectura existente, la sesión de LAE en la Corporación Talentum se convierte en un foro eficaz para idear y priorizar mejoras arquitectónicas significativas.

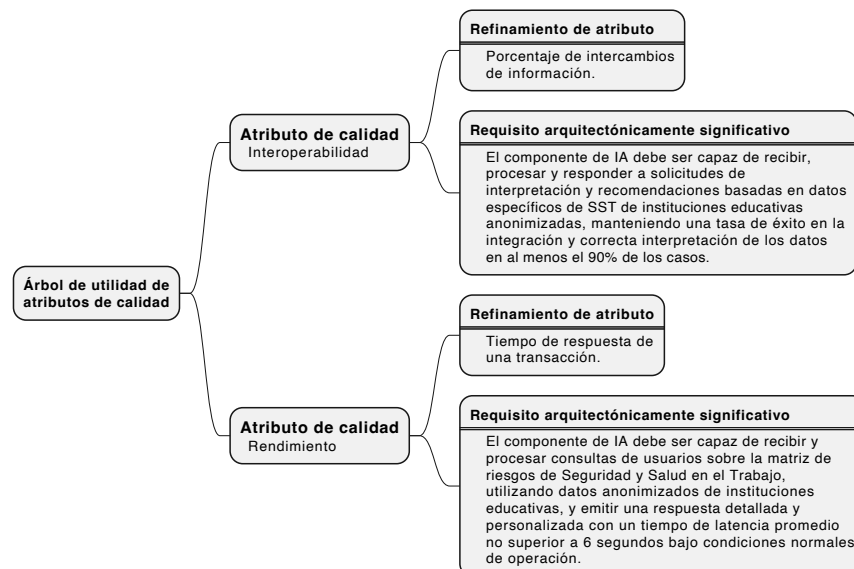


Figura 4.22: Árbol de utilidad de los atributos de calidad

6. **Paso 6: Lluvia de ideas y priorización de escenarios** En el paso 6 del proceso de Evaluación de Arquitectura Ligera (LAE) en la Corporación Talentum, los participantes dedicaron unos 40 minutos a discutir sobre los registros de decisiones arquitectónicas (Architectural Decision Records, ADR). Aunque los desarrolladores no son seniors, mostraron un gran interés

en comprender las decisiones tomadas, generando un ambiente enriquecedor para el intercambio de ideas y propuestas. Durante esta sesión, además de presentar propuestas, se llevó a cabo la priorización de atributos de calidad, enfocándose principalmente en la interoperabilidad, seguido por el rendimiento, como los atributos prioritarios a cumplir. Esto orientó la discusión hacia la búsqueda de mejoras en la interoperabilidad.

En este contexto, surgieron discusiones sobre posibles componentes orquestadores alternativos a Flowise, incluyendo Langflow y LLMStack, presentados por los desarrolladores que los habían investigado previamente. Langflow se destacó por su alta configurabilidad, permitiendo una adaptación detallada a las necesidades específicas de cada proyecto. Sin embargo, esta misma característica aumenta la complejidad de su manejo, requiriendo que los usuarios posean un nivel más avanzado de experticia técnica para manipular adecuadamente la herramienta y sus componentes, y crear flujos de orquestación eficientes. Aunque Langflow ofrece ventajas significativas en términos de personalización, su complejidad puede representar una barrera para equipos menos experimentados.

Por otro lado, LLMStack se presentó como una alternativa más sencilla y fácil de usar en comparación con Langflow. Aunque su simplicidad viene con limitaciones, especialmente en términos de integraciones disponibles, lo que puede dificultar la creación de flujos de orquestación que interactúen con otras tecnologías que podrían añadir valor al sistema. Además, LLMStack presenta desafíos en cuanto a su despliegue, lo que puede complicar su implementación en entornos que requieren una integración fluida y eficiente.

La discusión sobre estas herramientas no solo permitió compartir conocimiento entre los miembros del equipo, sino que también ayudó a evaluar si las nuevas propuestas podrían generar valor en el contexto actual del proyecto, especialmente en términos de interoperabilidad. La Corporación Talentum valora estos espacios de debate, ya que fomentan que los desarrolladores investiguen, prueben y socialicen tecnologías e información, contribuyendo significativamente al enriquecimiento del conocimiento colectivo del equipo. Estos encuentros son fundamentales para mantener una mentalidad abierta y estar continuamente al tanto de las nuevas tecnologías que podrían integrarse en futuros proyectos.

7. Paso 7: Análisis de Enfoques Arquitectónicos en Corporación Talentum

Durante el séptimo paso de la Evaluación de Arquitectura Ligera en la Corporación Talentum, se realizó un análisis de los enfoques arquitectónicos, con especial atención en el estilo RAG y la arquitectura basada en servicios. Este paso fue importante para los desarrolladores, quienes inicialmente se mostraron inseguros debido a su falta de familiaridad con el estilo arquitectónico RAG, detallado en el Framework. La sesión facilitó un espacio esencial para resolver dudas y llenar los vacíos de conocimiento sobre la aplicación práctica de este estilo.

Aspectos importantes del proceso LAE sobre el paso 7:

- **Estilo de Arquitectura RAG:** El estilo arquitectónico RAG, o Arquitectura de Respuesta Generativa, es una metodología emergente diseñada para sistemas que requieren una alta adaptabilidad y respuesta dinámica a las condiciones cambiantes del entorno operativo. Este estilo se caracteriza por su capacidad para integrar componentes de IA, como el aprendizaje automático y el procesamiento de lenguaje natural, facilitando así sistemas altamente responsivos y adaptativos. Los beneficios de adoptar la arquitectura RAG en el proyecto incluyen una mejor capacidad de respuesta a las necesidades del usuario final, una integración más fluida de nuevas funcionalidades de IA, y una mayor escalabilidad al permitir que el sistema evolucione con menos intervenciones manuales.
- **Estilo de Arquitectura Basada en Servicios:** Por otro lado, la arquitectura basada en servicios se discutió como un pilar fundamental para el desarrollo del caso de estudio. Este estilo arquitectónico, que es un híbrido entre la arquitectura monolítica y la de microservicios, es valorado por su flexibilidad. Ofrece menos complejidad y costos en comparación con otras arquitecturas distribuidas, lo que la convierte en una opción popular para aplicaciones empresariales que necesitan escalabilidad y fácil mantenimiento. La topología de esta arquitectura se organiza en capas macro distribuidas, que incluyen interfaces de usuario, servicios de granularidad gruesa (servicios de dominio) y una base de datos monolítica, implementados de manera independiente. Durante el paso 7, se exploró cómo la escalabilidad, la tolerancia a fallos y el rendimiento pueden ser administrados a través de múltiples instancias de un servicio de dominio, con técnicas de balanceo de carga para asegurar que la interfaz de usuario siempre se conecte a una instancia saludable y disponible.

El debate y análisis en este paso permitieron a los desarrolladores comprender mejor cómo los estilos arquitectónicos RAG y basados en servicios pueden coexistir y complementarse para mejorar la interoperabilidad, un aspecto importante que resalta el Framework con integración de IA. Al final de esta sesión, se alcanzó un consenso sobre la viabilidad y el valor agregado de estos enfoques arquitectónicos, asegurando que la arquitectura propuesta no solo cumple con los requisitos técnicos sino que también facilita una integración efectiva y eficiente de soluciones de IA, posicionando al sistema para una evolución continua y adaptativa.

8. Paso 8: Conclusiones y Captura de Resultados del Proceso de Evaluación de Arquitectura Ligera en Corporación Talentum

Al finalizar el proceso de Evaluación de Arquitectura Ligera (LAE) en la Corporación Talentum, se presentaron diversas conclusiones importantes que reflejan el valor y la eficacia del proceso:

- **Enriquecimiento del Equipo:** La ejecución de LAE ha sido un proceso enriquecedor para el equipo de desarrollo. Ha fomentado la investigación y búsqueda de nuevas tecnologías que pueden aportar un valor significativo a la Corporación Talentum, destacando la importancia de la actualización continua y el aprendizaje activo entre los profesionales involucrados.

- **Validación del Orquestador Flowise:** Durante la evaluación, se destacó la herramienta de orquestación Flowise, que se ha demostrado ser una opción robusta para la integración y operación de soluciones de IA dentro de la organización. Flowise se caracteriza por su amplia gama de integraciones y su simplicidad tanto en la operación como en el despliegue, apoyada por una documentación extensa y una comunidad activa de usuarios. Estas características hacen que Flowise sea particularmente adecuado para la Corporación Talentum, ya que facilita la interoperabilidad y mejora la eficiencia operativa al permitir una integración fluida con múltiples componentes y plataformas, incluidas soluciones en la nube como AWS Bedrock, lo que optimiza los costos y acelera los tiempos de implementación.
- **Adopción de Estilos Arquitectónicos Beneficiosos:** El equipo reconoce y valora los estilos arquitectónicos promovidos por el Framework, en particular el estilo arquitectónico RAG y el estilo basado en servicios. Este último se aprecia por su flexibilidad y menor complejidad en comparación con otras arquitecturas distribuidas, lo que resulta en menores costos y simplificación en la gestión. La arquitectura basada en servicios permite una escalabilidad efectiva y una tolerancia a fallos robusta, gestionando la carga a través de múltiples instancias de servicios de dominio. Esta estructura facilita la integración de IA al permitir una expansión y adaptación ágil dentro de los sistemas de SST, asegurando que las soluciones sean escalables y puedan manejar adecuadamente incrementos en la demanda y en la complejidad de las operaciones.
- **Manejo de Riesgos Asociados con Tecnologías Emergentes:** Aunque la adopción de tecnologías de vanguardia ofrece numerosas ventajas, también introduce riesgos, especialmente debido a su novedad en el mercado. Para mitigar estos riesgos, el equipo se compromete a mantener una vigilancia constante sobre las vulnerabilidades, las actualizaciones de seguridad y las mejoras en las configuraciones de las tecnologías emergentes. Esta práctica asegura que cualquier riesgo potencial se maneje proactivamente, manteniendo la seguridad y la integridad del sistema.
- **Confirmación de la Arquitectura Propuesta:** Todos los participantes acordaron que la arquitectura del Attribute Driven Design propuesta y revisada durante la sesión de LAE cumple con las expectativas y los requisitos del caso de estudio. Sin embargo, el proceso de aplicación de LAE ha sido valioso para conocer otras herramientas de orquestación que están emergiendo en la industria y que podrían ser adecuadas para diferentes casos de uso. Esta visión ampliada ayuda a la Corporación Talentum a permanecer informada sobre las tendencias del mercado y a considerar futuras adaptaciones o mejoras.

Estas conclusiones resaltan el éxito del proceso de LAE en fomentar un ambiente colaborativo y proactivo dentro de la Corporación Talentum, permitiendo una mejora continua en la integración de la IA en los sistemas de SST. Esto no solo fortalece la infraestructura tecnológica actual, sino que también prepara a la organización para futuros desafíos y oportunidades.

4.5.2. Evaluación con especialistas de SST

Es importante validar si los requerimientos acordados con los especialistas de SST de la Corporación Talentum fueron efectivamente satisfechos tras la implementación del Framework propuesto al caso de estudio. Aunque se completaron todos los requerimientos acordados, la verdadera medida de éxito radica en la satisfacción de estos especialistas. En consecuencia, se elaboró un artefacto específicamente diseñado para evaluar su grado de satisfacción, asegurando que no solo se cumplieron los requisitos técnicos, sino que también se alinearon con sus expectativas y necesidades. Sin embargo, al ser una integración con Inteligencia Artificial, es esencial recoger la retroalimentación de estas personas, por lo que este artefacto se aplicó después de la implementación del Framework al caso de estudio. Se organizó una sesión de pruebas con el equipo de especialistas de SST. Esta sesión consistió en crear un espacio con 10 profesionales del área de SST, quienes interactuaron con el chatbot de IA integrado al sistema de información de Seguridad y Salud en el Trabajo mediante diversos prompts y preguntas para evaluar su funcionamiento según sus conocimientos, necesidades y experiencia.

En esta sesión, se realizó la aplicación de un artefacto de evaluación, donde los participantes respondieron a una encuesta de satisfacción sobre el prototipo del Chatbot de IA para el sistema de información de SST. Dicho artefacto incluía 12 preguntas, donde las primeras 11 solicitaban asignar una calificación en una escala del 1 al 5, siendo 1 la nota más baja y 5 la más alta, según el enunciado de cada pregunta. La última pregunta del artefacto permitía una respuesta abierta y consistía en dejar una sugerencia para mejorar la experiencia del usuario respecto a las interacciones que habían tenido con el prototipo. Es esencial destacar que las encuestas realizadas son completamente anónimas. La organización ha decidido no revelar la identidad de los especialistas involucrados en este proceso para mantener la confidencialidad de sus productos y prácticas de Seguridad y Salud en el Trabajo. Esta medida asegura la integridad de la información y fomenta una participación sincera y abierta por parte de los especialistas.

En la siguiente tabla se presentan las preguntas que componían el artefacto de evaluación que los participantes debían diligenciar después de interactuar con el prototipo.

Num.	Pregunta
1	¿Qué tan fluida y natural considera que fue la conversación con el chatbot?
2	¿Qué tan bien considera que se desempeña el chatbot a la hora de seguir instrucciones?
3	¿Considera que la información proporcionada por el chatbot es completa y da recomendaciones relevantes respecto a sus preguntas?
4	¿Qué tan bien cree que el prototipo cumple con las necesidades específicas de SST?
5	¿Qué tan confiables fueron los resultados o salidas generadas por el prototipo?
Continúa en la siguiente página	

Num.	Pregunta
6	¿Qué tan útil le resultó el chatbot para resolver sus dudas con base a la fuente de datos de matriz de riesgos de SST?
7	¿Cuál fue su experiencia al trabajar con las funcionalidades integradas de IA en el prototipo?
8	¿Cómo calificaría la interfaz de usuario del prototipo en términos de facilidad de uso?
9	En cuanto a la velocidad y el rendimiento del prototipo, ¿cómo evaluaría su experiencia?
10	Basado en su experiencia, ¿qué tan escalable le parece el prototipo para futuras expansiones o modificaciones en el área de SST?
11	¿Recomendaría este prototipo a otros profesionales del área de SST?
12	¿Tiene alguna sugerencia para mejorar la experiencia del usuario con este prototipo? (Respuesta abierta)

Tabla 4.18: Preguntas del artefacto de evaluación para los especialistas de SST

Una vez que los profesionales especializados en el área de Seguridad y Salud en el Trabajo terminaron de diligenciar el artefacto de evaluación tras participar en la sesión de pruebas, se procedió a realizar el análisis respectivo de este para validar la experiencia y satisfacción que habían tenido con respecto a su interacción con el prototipo de SST.

A continuación, se presenta un análisis detallado del artefacto de evaluación, apoyándose en un conjunto de gráficas que permiten comprender y analizar la información proporcionada por los especialistas de Seguridad y Salud en el Trabajo. Se resaltarán los aspectos más relevantes donde se evidencia el impacto de la prueba, ofreciendo una visión clara de los resultados obtenidos.

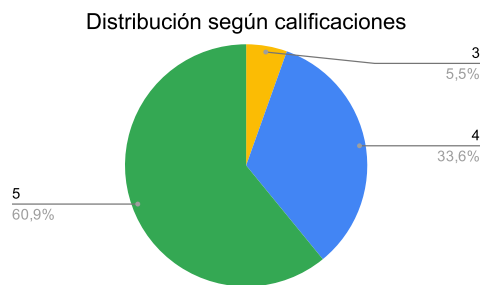


Figura 4.23: Gráfico de distribución según calificaciones obtenidas

El gráfico de la Figura 4.23 ilustra la distribución de las calificaciones otorgadas por los 10 profesionales especialistas en el área de Seguridad y Salud en el Trabajo al prototipo del caso de estudio de SST. Las calificaciones, basadas en las respuestas a las preguntas del artefacto de eva-

luación, variaban en una escala de 1 a 5, donde 1 era la calificación más baja y 5 la más alta.

El análisis del gráfico revela que la mayoría de los participantes, equivalentes al 60.9%, otorgaron la máxima puntuación de 5 al chatbot en las diferentes preguntas contestadas. Además, un 33.6% de los participantes dieron calificaciones de 4, y solo un 5.5% asignó puntuaciones de 3 al chatbot. Cabe destacar que, según las respuestas, ningún participante otorgó calificaciones de 1 o 2 al chatbot de IA en ninguna de las preguntas del artefacto de evaluación.

El hecho de que la gran mayoría de las calificaciones se concentraran en los puntajes de 4 o 5 indica que los resultados fueron generalmente muy positivos y sugieren que los especialistas estuvieron satisfechos con el prototipo del chatbot de SST. La mayoría destacó que el chatbot fue útil, informativo y fácil de usar. Además, resaltaron que el proceso de ingeniería de prompts realizado con la guía del Framework fue de gran ayuda y las iteraciones contribuyeron significativamente a su comprensión del proceso de uso.

Otro gráfico que aporta valor al análisis es uno de distribución según calificaciones por pregunta, que se presenta a continuación, para entender mejor el comportamiento de cada pregunta y comprender mejor la perspectiva de los especialistas de SST.

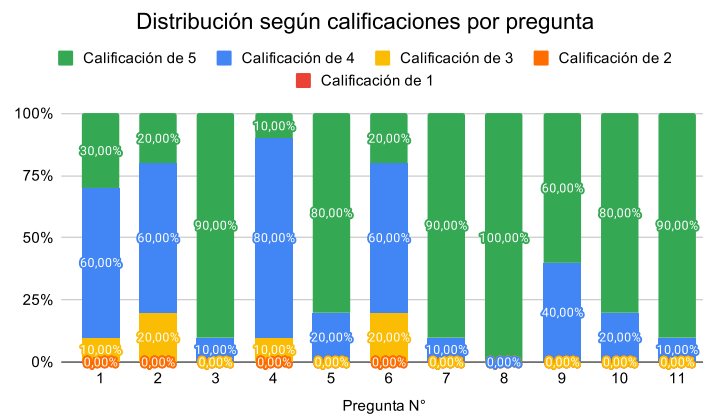


Figura 4.24: Gráfico de distribución según calificaciones por pregunta

El gráfico de la Figura 4.24 muestra la distribución de las calificaciones por pregunta otorgadas por los 10 profesionales especialistas del área de SST del prototipo del caso de estudio. Las respuestas están calificadas en una escala del 1 al 5, donde 1 es la nota más baja y 5 la más alta.

Al analizar el gráfico, se observa que la mayoría de las calificaciones se concentran en los puntajes de 4 y 5. Dentro de estos porcentajes, es notable que en las preguntas 3, 7 y 11, el 90% de las calificaciones fue un puntaje de 5. Estas preguntas respectivamente evalúan qué tan completa

y relevante es la información suministrada por el chatbot, el nivel de satisfacción de los usuarios al trabajar con funcionalidades integradas de IA, y si recomendarían el uso del prototipo de Chatbot de IA a otros profesionales de SST en la Corporación Talentum.

Estos resultados indican que los usuarios quedaron altamente satisfechos con la capacidad de respuesta del chatbot. A pesar de que consideran que hay aspectos que podrían mejorarse, la experiencia de interactuar con el prototipo de Chatbot por primera vez captó significativamente su atención, motivándolos a recomendar su implementación a otros profesionales en el área de SST.

Como mención especial, cabe destacar el resultado de la pregunta 8, la cual evalúa la facilidad de uso de la interfaz del prototipo de chatbot. Esta fue la única pregunta que alcanzó un resultado del 100% con un puntaje de 5 de parte de todos los usuarios. Esto permite concluir que la interfaz del chatbot es intuitiva y fácil de manejar para todos los usuarios en general.

El último gráfico que se presenta es el de las calificaciones promedio por pregunta, el cual es importante para ilustrar la tendencia central de las respuestas. Este gráfico proporciona una estimación del valor típico de los datos, ofreciendo una visión general de cómo los especialistas valoraron cada aspecto del prototipo. Dicho gráfico se expone a continuación.

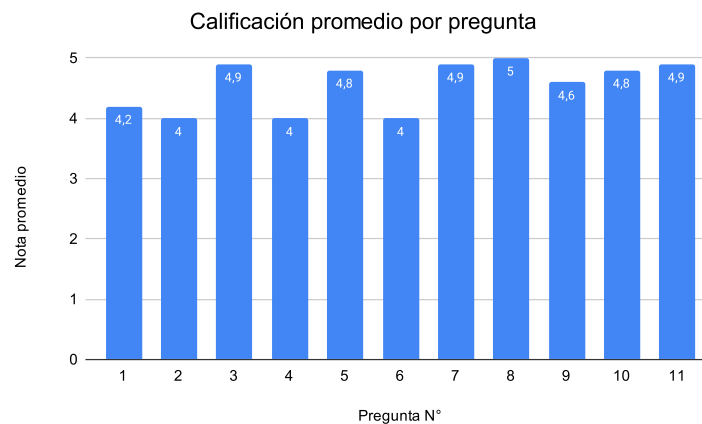


Figura 4.25: Gráfico de calificación promedio por pregunta

El gráfico en la Figura 4.25 muestra el valor promedio de las calificaciones asignadas por los 10 profesionales especialistas del área de SST. Este gráfico busca evidenciar la tendencia de las respuestas de los especialistas al prototipo del caso de estudio, basándose en sus calificaciones para cada una de las preguntas del artefacto de evaluación, que seguía una escala del 1 al 5, donde 1 era la nota más baja y 5 la más alta.

Al analizar el gráfico, se observa que el valor promedio de calificación en cada pregunta fue

igual o mayor que 4. Los resultados indican que las preguntas 2, 4 y 6 recibieron el valor promedio de calificación más bajo con un puntaje de 4, que, dentro de la escala propuesta, representa una buena calificación, evidenciando que ninguna pregunta obtuvo un resultado negativo en cuanto al valor promedio de las calificaciones.

Además, al igual que en la gráfica de la Figura 4.24, se destaca que las preguntas 3, 7, 8 y 11 alcanzaron los mejores resultados en términos de las calificaciones promedio más altas.

Al considerar estos promedios de manera global y calcular un promedio general para todo el artefacto de evaluación, basado en los puntajes asignados por los 10 participantes a cada una de las 11 preguntas, se concluye que la evaluación logró una calificación promedio total de 4.55. Este resultado es bastante favorable, ya que 4.55 en una escala del 1 al 5 equivale a un 91 % del valor total del rango de calificación, lo que se traduce en que el nivel porcentual de satisfacción de los especialistas de SST con respecto a su experiencia interactuando con el prototipo del chatbot de SST es del 91 %.

Este análisis detallado no solo subraya la utilidad del chatbot dentro del contexto de Seguridad y Salud en el Trabajo, sino que también resalta las áreas de potencial mejora y adaptación para futuras iteraciones del prototipo, asegurando así una integración más efectiva y una aceptación más amplia entre los especialistas.

Conclusiones

Contents

5.1. Conclusiones generales	115
5.2. Trabajo futuro	119
5.2.1. Generalización del Framework a Otros Sectores	119
5.2.2. Propuesta Espacio Colaborativo para la Mejora Continua del Framework	120
5.2.3. Conclusiones	121

En el presente proyecto se ha desarrollado un Framework diseñado para orientar la integración de componentes de Inteligencia Artificial en soluciones de software destinadas al sector de Seguridad y Salud en el Trabajo. La colaboración con la Corporación Talentum permitió aplicar este Framework en un caso de estudio, evidenciando su aplicabilidad en contextos reales. Este estudio ha abordado desafíos significativos relacionados con la interoperabilidad, la exactitud en la extracción de datos y la modificación de arquitecturas de software preexistentes. El Framework propuesto ofrece una serie de directrices, buenas prácticas y recomendaciones arquitectónicas que facilitan la incorporación de la IA, asegurando una integración ajustada a las necesidades específicas del sector de la seguridad y salud laboral. Este capítulo final no solo recopila los hallazgos y contribuciones derivados de la aplicación del Framework al caso de estudio, sino que también explora posibles trabajos futuros. Se destaca el potencial del Framework para ser adaptado y aplicado en otros sectores, demostrando su capacidad para generar valor más allá del ámbito de la Seguridad y Salud en el Trabajo, fomentando así la innovación y la mejora continua en la integración de tecnologías de IA en diferentes infraestructuras de software.

5.1. Conclusiones generales

La implementación del Framework en el caso de estudio con la Corporación Talentum, junto con el análisis de diversas fuentes, ha facilitado la identificación de elementos clave para el desarrollo de aplicaciones y herramientas de Inteligencia Artificial. Estos componentes han validado la utilidad y la aplicabilidad de la IA en el contexto de la Seguridad y Salud en el Trabajo, destacando su capacidad para integrarse y generar valor a este sector.

Este logro se alcanzó a lo largo del desarrollo del proyecto que, a pesar de enfrentar varios obstáculos, se culminó exitosamente, conduciendo a las siguientes conclusiones:

- La integración de bases de datos vectoriales en los Modelos de Lenguaje Grande (LLMs) ha significado un avance notable en este campo, mejorando considerablemente su rendimiento y precisión. Aunque el desarrollo de estas bases de datos ha sido un proceso de varios años, su reciente optimización permite un uso más eficiente con diversos LLMs. Es vital reconocer las restricciones específicas de estas bases de datos en comparación con otras tecnologías, para garantizar que se adapten adecuadamente a los casos de uso previstos y proporcionen resultados óptimos.
- Durante la implementación del proceso de embedding para almacenar el contenido de los documentos en una base de datos vectorial, resulta esencial considerar el modelo de embedding seleccionado para la vectorización. A medida que se inician las pruebas con Modelos de Lenguaje Grande (LLMs) y su uso conjunto con embeddings, es recomendable seleccionar un modelo de embedding que esté específicamente alineado con el LLM utilizado. Antes de optar por un modelo de embedding alternativo, es importante que se revisen sus especificaciones para evaluar si realmente ofrece mejoras significativas. Una elección precisa es fundamental, pues una selección inadecuada podría resultar en discrepancias significativas en la precisión de la información almacenada.
- En la implementación de sistemas de generación de texto asistida por recuperación, utilizando la arquitectura Retrieval-Augmented Generation (RAG), con documentos extensos, es importante optimizar cómo se almacenan y recuperan los datos. Fragmentar los documentos en unidades más pequeñas, como párrafos o secciones, y emplear técnicas avanzadas de indexación mediante embeddings semánticos, facilita la recuperación eficaz y precisa de la información. Herramientas como Qdrant, una base de datos vectorial, resultan ideales para estas operaciones, ya que soportan búsquedas basadas en la similitud semántica y están optimizadas para manejar grandes volúmenes de datos. La elección de la tecnología correcta, como Qdrant, que soporte operaciones vectoriales y que cumpla con las necesidades de escalabilidad y rendimiento del sistema, es fundamental. La capacidad de la base de datos para interactuar de manera fluida con el modelo de lenguaje, posiblemente a través de APIs diseñadas para facilitar esta comunicación, es esencial para maximizar la efectividad del modelo en la generación de respuestas precisas y contextualmente relevantes.
- Un desafío significativo al almacenar documentos en bases de datos vectoriales es la gestión de elementos gráficos o formatos de texto complejos, como textos en columnas dobles, tablas e imágenes de texto. Estos formatos requieren un procesamiento especial para optimizar su almacenamiento y asegurar un acceso eficiente por parte del LLM. Implementar estrategias adecuadas para manejar estos elementos gráficos y formatos de texto no convencionales es esencial para aprovechar al máximo las capacidades de las bases de datos vectoriales en escenarios de búsqueda y recuperación de información.
- La arquitectura Retrieval-Augmented Generation (RAG) ha mejorado significativamente el desempeño de los modelos de lenguaje grandes (LLMs) en tareas de extracción de información. Esta arquitectura permite la integración de información externa, lo que ayuda a superar las

barreras de conocimiento limitado y la interpretación de contextos complejos. No obstante, se debe considerar que cada LLM tiene restricciones en cuanto al volumen de contexto que puede procesar, lo cual limita la cantidad de información que puede ser efectivamente incorporada en un solo prompt.

- La selección de un LLM adecuado requiere una revisión de sus limitaciones y ventajas, así como de la documentación asociada. La variabilidad en el rendimiento de los distintos LLMs según los casos de uso que aborda, el idioma, la fuente de datos y el contexto específico hace esencial elegir un modelo que se alinee con los requisitos del proyecto. Esta decisión es crítica para garantizar que el modelo seleccionado ofrezca el desempeño esperado y se adecúe al contexto de uso.
- La adopción de tecnologías serverless, como AWS Bedrock, Google Cloud Functions y Azure Functions, ha demostrado ser una estrategia efectiva para reducir los costos iniciales en proyectos de Inteligencia Artificial. Estas tecnologías permiten la implementación de soluciones de IA sin grandes inversiones en infraestructura física y minimizan los costos operativos, ya que solo se incurre en gastos por el tiempo de cómputo utilizado. Además, facilitan la escalabilidad automática de las aplicaciones según la demanda, permitiendo a los desarrolladores centrarse en la mejora de las capacidades de sus soluciones de IA.
- Aunque los LLM han sido fundamentales en aplicaciones como generación de texto, traducción de idiomas y creación de contenido creativo, no deben considerarse soluciones universales. Estos modelos tienen limitaciones significativas y no deben sustituir el juicio humano en la toma de decisiones críticas. Es importante que se utilicen como herramientas de apoyo y no como reemplazos de las capacidades humanas, como se ha enfatizado a los especialistas en Seguridad y Salud en el Trabajo de la Corporación Talentum.
- La continua evolución de las tecnologías de IA y los LLMs, con actualizaciones frecuentes y mejoras en las capacidades, requiere que las organizaciones mantengan una vigilancia constante sobre las últimas innovaciones para optimizar sus aplicaciones. Establecer procedimientos para adaptarse a estos cambios permite a las organizaciones mantenerse competitivas, mejorar sus procesos y explorar nuevas oportunidades en el campo de la IA.
- La integridad y la calidad de los datos son esenciales para el éxito de las arquitecturas de generación aumentada por recuperación (RAG). La precisión de los datos extraídos y utilizados para enriquecer las respuestas del modelo afecta directamente la calidad y la utilidad de las soluciones generadas. Invertir en la curación y el etiquetado adecuado de los datos no solo mejora los resultados, sino que también maximiza el potencial del sistema para ofrecer soluciones innovadoras y valiosas. En este contexto, los datos representan el fundamento sobre el cual se construyen y validan todas las iniciativas de IA.
- Identificar claramente los casos de uso específicos y los datos o inputs necesarios para aplicar el diseño basado en atributos (Attribute-Driven Design, ADD) resulta esencial en el desarrollo de productos con Inteligencia Artificial. Analizar cuidadosamente los insumos facilita una

comprensión adecuada del entorno y la estrategia del producto, asegurando que los registros de decisiones arquitectónicas (Architecture Decision Records, ADR) estén sincronizados con los objetivos y desafíos específicos del negocio. Por lo tanto, un diseño arquitectónico bien estructurado y orientado a datos es imprescindible para maximizar la funcionalidad y el impacto de las soluciones de IA, garantizando que se alcanzan los objetivos estratégicos y operacionales de la empresa.

- Es fundamental reconocer que el Framework propuesto no sustituye la función del arquitecto de software en el diseño arquitectónico, sino que proporciona un apoyo en este proceso. Actuando como un repositorio de prácticas, recomendaciones, criterios y componentes de software, este Framework se revela como un recurso invaluable para la integración de soluciones de Inteligencia Artificial con arquitecturas preexistentes en productos de Seguridad y Salud en el Trabajo. Proporciona al arquitecto las herramientas necesarias para tomar decisiones informadas y optimizar la implementación de nuevas tecnologías, asegurando que complementen y enriquezcan los sistemas existentes, facilitando así la interoperabilidad y fomentando la innovación, a la vez que fortalecen el papel del arquitecto en la implementación de soluciones tecnológicas avanzadas y adaptadas al sector.
- La vectorización de documentos en formato de texto plano (.txt) es comparativamente más directa que otros formatos que incluyen elementos complejos como tablas e imágenes. La simplicidad del texto plano elimina las complicaciones durante el proceso de fragmentación y manejo del contenido diverso. Para asegurar que la vectorización sea efectiva y permita aplicar adecuadamente métricas de similitud como la distancia coseno, es importante gestionar de manera precisa el tamaño de los fragmentos y su solapamiento. Ajustar estos parámetros mejora significativamente la detección de similitudes y la eficiencia de las búsquedas de texto, asegurando que los resultados sean precisos y pertinentes.
- La ingeniería de prompts es fundamental para optimizar la interacción de los usuarios con el Chatbot de IA del caso de estudio, mejorando la comprensión del contexto y las intenciones del usuario y produciendo respuestas más naturales y adecuadas. Al diseñar cuidadosamente los prompts, se disminuyen las ambigüedades y se facilita la escalabilidad hacia distintos contextos, mejorando significativamente la experiencia del usuario y la eficiencia del sistema.
- En el ámbito de la Seguridad y Salud en el Trabajo, la ingeniería de prompts se muestra como una herramienta extremadamente eficaz cuando se emplea adecuadamente. La creación de espacios de trabajo colaborativos con usuarios de SST permite fomentar el conocimiento sobre las directrices que proporciona el Framework. Este enfoque colaborativo no solo facilita la implementación del sistema, sino que asegura que los prompts estén alineados con las necesidades reales y las prácticas óptimas del sector, mejorando así la efectividad de las medidas de Seguridad y Salud en el Trabajo.
- Según los resultados de las evaluaciones realizadas por los usuarios, el prototipo del Chatbot de IA en Seguridad y Salud en el Trabajo ha sido validado exitosamente. Los especialistas del

sector han expresado un alto grado de satisfacción, asignando al prototipo una calificación promedio de 4.55 sobre 5, lo que representa un 91 % de satisfacción. Este resultado subraya el potencial del chatbot como una herramienta eficaz para la gestión de la Seguridad y Salud en el Trabajo. Sin embargo, se recomienda continuar el desarrollo del producto, añadiendo nuevas funcionalidades que aporten un valor incrementado, convirtiéndolo en una herramienta más completa y precisa para el mejoramiento continuo de los procesos de seguridad y salud laboral.

5.2. Trabajo futuro

En esta sección se pretenden identificar y proponer líneas de investigación futuras que se derivan del trabajo realizado en este proyecto de grado. A continuación, se explorarán diversas áreas en las que se puede continuar el desarrollo e investigación para enriquecer el valor del framework a nivel de industria. El objetivo es ampliar su aplicabilidad y utilidad, permitiendo que una mayor cantidad de organizaciones puedan implementarlo eficazmente en sus operaciones.

5.2.1. Generalización del Framework a Otros Sectores

El framework inicialmente desarrollado para la integración de componentes de Inteligencia Artificial en soluciones de software específicamente en el sector de Seguridad y Salud en el Trabajo, posee un potencial significativo para adaptarse y aplicarse en otros sectores industriales. Esta capacidad de adaptación se propone explorar mediante una metodología sistemática que aborda los retos y oportunidades en un espectro más amplio de aplicaciones. A continuación, se detallan los aspectos clave de esta metodología:

1. Abstracción de Prácticas de Manejo de Datos:

- **Identificación de patrones comunes:** La identificación de patrones comunes en el manejo de datos entre el sector de SST y otros sectores como por ejemplo manufactura, servicios financieros y salud. Esto se logra a través de estudios comparativos que mapean y analizan cómo cada sector gestiona sus datos, revelando similitudes que pueden ser abordadas por el framework y diferencias que requieren adaptaciones específicas. Estos estudios ayudarán a crear una base de conocimiento que sustente el desarrollo de módulos de datos genéricos y flexibles.
- **Desarrollo de módulos de datos genéricos:** Se propone la creación de módulos que puedan ser configurados y adaptados a los diferentes tipos de datos y requerimientos de procesamiento que cada sector demanda. Estos módulos permitirán una personalización según las necesidades específicas de cada sector, manteniendo la integridad y la utilidad de los datos en diferentes contextos de aplicación.

2. Estructuración y Normalización de Datos para la IA:

- **Especificación de estructuras de datos:** Definir y estandarizar estructuras de datos es importante para que puedan ser universalmente aplicables y eficaces en el trabajo con diferentes modelos de IA. Esto asegura que la información se presenta de manera que facilita el análisis y la generación de insights relevantes y precisos, independientemente del sector industrial en el que se implemente.
- **Desarrollo de un conjunto de herramientas para la transformación de datos:** Se recomienda desarrollar un conjunto de herramientas dedicadas a la transformación de datos sectoriales a un formato estándar. Estas herramientas deberán ser capaces de procesar eficazmente los datos para que sean compatibles con diversos modelos de IA, facilitando así la interoperabilidad y la efectividad de las soluciones basadas en IA.

3. Pruebas de Interoperabilidad en Diversos Contextos:

- **Implementación de pruebas piloto:** Para validar la adaptabilidad y efectividad del framework en diferentes sectores, se sugiere la realización de pruebas piloto en una variedad de contextos operativos. Estas pruebas permitirán identificar y ajustar cualquier desajuste o ineficiencia antes de una implementación más amplia.
- **Recopilación y análisis de feedback:** La recopilación y análisis de feedback de los usuarios durante las pruebas piloto son fundamentales. Esta retroalimentación será esencial para iterar y mejorar el framework, basándose en datos empíricos y experiencias reales de usuarios en distintos sectores. Esto asegurará que el framework no solo cumple con los requisitos técnicos sino que también responde a las necesidades prácticas de los usuarios finales.

5.2.2. Propuesta Espacio Colaborativo para la Mejora Continua del Framework

La colaboración intersectorial es fundamental para la adaptación y generalización efectiva del framework propuesto. Ante la necesidad de una plataforma que permita una interacción dinámica y continua entre diferentes expertos, se sugiere la creación de un espacio colaborativo en línea, diseñado para facilitar la comunicación y el intercambio de conocimientos. Este espacio, operando bajo el modelo de una wiki, permitirá a académicos, profesionales de la industria y expertos en tecnología contribuir activamente al desarrollo y refinamiento del framework.

1. Creación de un Repositorio Dinámico de Buenas Prácticas:

- **Implementación de una Plataforma Wiki:** Se propone el desarrollo de una plataforma estilo wiki que sea accesible de manera pública o mediante registro, según se determine necesario para mantener la calidad y relevancia de las contribuciones. Esta plataforma permitirá a los usuarios documentar de manera continua las buenas prácticas, las modificaciones al framework, y las lecciones aprendidas en la aplicación del mismo en distintos sectores.
- **Contribuciones en Tiempo Real:** La plataforma permitirá que los usuarios aporten en tiempo real, ofreciendo un entorno colaborativo en el que las contribuciones pueden

incluir ejemplos de implementación, ajustes basados en experiencias prácticas y discusiones sobre posibles mejoras. Este enfoque orgánico y comunitario facilitará la evolución continua del framework, adaptándose rápidamente a los desafíos y oportunidades que surgen en diversos contextos operativos.

2. Desarrollo de Estándares Flexibles para la Integración de IA:

- **Foro para la Estandarización de Prácticas:** La plataforma también servirá como un foro para el debate y consenso sobre prácticas estándar de integración de IA. La meta es desarrollar y refinar APIs comunes y protocolos de intercambio de datos que sean ampliamente aplicables, facilitando así la interoperabilidad entre diferentes sistemas y sectores.
- **Adaptación a Cambios y Tendencias Tecnológicas:** Dado que la tecnología y los requisitos del mercado evolucionan rápidamente, la plataforma estará diseñada para adaptarse a estos cambios. Mantener la relevancia y efectividad del framework será posible gracias a una base de conocimiento que se actualiza constantemente, reflejando las últimas tendencias y avances tecnológicos.

5.2.3. Conclusiones

- La propuesta de generalizar el framework a otros sectores no solo aumentará su relevancia y aplicabilidad, sino que también facilitará la creación de soluciones de software basadas en IA más robustas, flexibles y eficientes. Este enfoque sistemático garantiza que el framework pueda adaptarse y evolucionar conforme a las necesidades cambiantes de diversos sectores industriales, promoviendo la innovación y mejorando la competitividad en el panorama tecnológico global.
- El establecimiento de este espacio colaborativo no solo simplificará el proceso de colaboración y mejora continua del framework, sino que también promoverá la transparencia y la adaptabilidad. Al permitir que múltiples stakeholders participen en un diálogo abierto y constructivo, esta plataforma garantiza que el framework se mantenga a la vanguardia de la tecnología y las mejores prácticas. Este enfoque democratizado y participativo es esencial para cultivar un ecosistema donde la innovación en la integración de la IA sea tanto inclusiva como efectiva, maximizando así los beneficios para todos los sectores implicados.

Bibliografía

- Adetayo, A. J., Aborisade, M. O., and Sanni, B. A. (2024). Microsoft Copilot and Anthropic Claude AI in education and library service. *Library Hi Tech News*.
- Agra, O. (2020). Redis — In-memory data store.
- Alay Coqueña, E. G. (2023). INFLUENCIA DE UNA APLICACION POR MEDIO DE CLUSTER EN LA OPTIMIZACION DE LA GESTION ACADEMICA DE LOS INSTITUTOS TECNOLOGICOS DE LA REGION PUNO 2019.
- Amazon (2012). AWS — Servicio de base de datos gestionada NoSQL (DynamoDB).
- Andrickson, J., Gabriel, E., Chacón, E., Puello, A., Rondon, F., Emeterio, W., and Reyes, M. (2021). Estándar OPC UA como solución a la integración industrial en una empresa. *Ciencia, Ingenierías y Aplicaciones*, 4(2):41–65.
- Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., Chu, E., Clark, J. H., Shafey, L. E., Huang, Y., Meier-Hellstern, K., Mishra, G., Moreira, E., Omernick, M., Robinson, K., Ruder, S., Tay, Y., Xiao, K., Xu, Y., Zhang, Y., Abrego, G. H., Ahn, J., Austin, J., Barham, P., Botha, J., Bradbury, J., Brahma, S., Brooks, K., Catasta, M., Cheng, Y., Cherry, C., Choquette-Choo, C. A., Chowdhery, A., Crepy, C., Dave, S., Dehghani, M., Dev, S., Devlin, J., Díaz, M., Du, N., Dyer, E., Feinberg, V., Feng, F., Fienber, V., Freitag, M., Garcia, X., Gehrmann, S., Gonzalez, L., Gur-Ari, G., Hand, S., Hashemi, H., Hou, L., Howland, J., Hu, A., Hui, J., Hurwitz, J., Isard, M., Ittycheriah, A., Jagielski, M., Jia, W., Kenealy, K., Krikun, M., Kudugunta, S., Lan, C., Lee, K., Lee, B., Li, E., Li, M., Li, W., Li, Y., Li, J., Lim, H., Lin, H., Liu, Z., Liu, F., Maggioni, M., Mahendru, A., Maynez, J., Misra, V., Moussalem, M., Nado, Z., Nham, J., Ni, E., Nystrom, A., Parrish, A., Pellat, M., Polacek, M., Polozov, A., Pope, R., Qiao, S., Reif, E., Richter, B., Riley, P., Ros, A. C., Roy, A., Saeta, B., Samuel, R., Shelby, R., Slone, A., Smilkov, D., So, D. R., Sohn, D., Tokumine, S., Valter, D., Vasudevan, V., Vodrahalli, K., Wang, X., Wang, P., Wang, Z., Wang, T., Wieting, J., Wu, Y., Xu, K., Xu, Y., Xue, L., Yin, P., Yu, J., Zhang, Q., Zheng, S., Zheng, C., Zhou, W., Zhou, D., Petrov, S., and Wu, Y. (2023). PaLM 2 Technical Report.
- Arcidiacono, J. (2020). Arquitectura de microservicios distribuidos para una plataforma que orquesta actividades orientadas a la recolección de datos con intervención humana.
- Baktash, J. A. and Dawodi, M. (2023). GPT-4: A Review on Advancements and Opportunities in Natural Language Pro-cessing. *J Electrical Electron Eng*, 2(4):548–549.
- Bass, L., Clements, P., Kazman, R., and Safari, a. O. M. C. (2021). Software Architecture in Practice, 4th Edition. page 464.

- Beese, J., Aier, S., Haki, K., and Winter, R. (2023). The impact of enterprise architecture management on information systems architecture complexity. *European Journal of Information Systems*, 32(6):1070–1090.
- Blinowski, G., Ojdowska, A., and Przybyłek, A. (2022). Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation. *IEEE Access*, 10:20357–20374.
- Caceres Claussen, J. D. and Carlosviza Collanqui, B. Y. (2022). Arquitectura cloud escalable y de alta disponibilidad que soporte una aplicación web de importaciones. Technical report.
- Cambarieri, M., Vivas, L., Garcia Martinez, N., Guimerans, G., and Viadana, A. (2020). Un marco de trabajo para el desarrollo de software de dominio específico en el contexto de Gobierno Digital.
- Carlos, J. and Pérez, R. (2022). Selección y adaptación de métricas para Microservicios. Technical report.
- Castro Estrada, A. R. (2022). Definición de una API para mejorar la interoperabilidad entre las plataformas de sistemas informáticos de los departamentos de estadística y geografía del INEGI.
- ChatGPT 4 and Sabit Ekin (2023). Prompt Engineering For ChatGPT. Technical report.
- Chen, B., Zhang, Z., Langrené, N., and Zhu, S. (2023). Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review.
- de Toledo, S. S., Martini, A., and Sjøberg, D. I. (2021). Identifying architectural technical debt, principal, and interest in microservices: A multiple-case study. *Journal of Systems and Software*, 177.
- DESHPANDE, N., ORTIZ, J., SARAKOGLU, I., SEMINI, C., TSAGARAKIS, N., BRYGO, A., FERNANDEZ, J., FRIGERIO, M., SACCARES, L., TOXIRI, S., and CALDWELL, D. G. (2017). NEXT-GENERATION COLLABORATIVE ROBOTIC SYSTEMS FOR INDUSTRIAL SAFETY AND HEALTH. pages 187–200.
- Dias, H. (2019). File Browser.
- Eberendu, A. C. (2016). Unstructured Data: an overview of the data of Big Data. *International Journal of Computer Trends and Technology*, 38(1):46–50.
- Eduardo Ruano Altamirano, B. (2024). Balanceo De Carga Para Mejorar La Eficiencia En Redes De Comunicaciones Entrantes De La Empresa Clicknet S.A. Technical report.
- Esther, T. and Tella, A. (2023). A Review Of The Ethics Of Artificial Intelligence And Its Applications In The United States. *International Journal on Cybernetics & Informatics (IJCI)*, 12(6):122–137.
- Fernandez (2022). Diseño y desarrollo de un microservicio para la gestión de información de monitorización y predicciones de tráfico en red.

- Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., and Wang, H. (2023). Retrieval-Augmented Generation for Large Language Models: A Survey.
- García Benzal, M. (2023). Mecanismo de registro de transacciones de información sanitaria basado en blockchain.
- García Clavijo, H. and Betancur Gil, V. (2023). Diseño De Un Prototipo De Metodología Integral Para La Automatización De Aseguramiento Y Control De Calidad De Software: Optimización De Tiempo Y Rentabilidad Empresarial. Technical report.
- García López, C. D. (2024). Diseño y desarrollo de un modelo de reingeniería de software para sistemas legados en una institución de educación superior.
- Geller, C., Haas, B., Kloeker, A., Hermens, J., Lampe, B., and Eckstein, L. (2024). CARLOS: An Open, Modular, and Scalable Simulation Framework for the Development and Testing of Software for C-ITS.
- Giménez, J. A. (2019). Buenas prácticas en el diseño de bases de datos.
- Gong, H., Shen, Y., Yu, D., Chen, J., and Yu, D. (2020). Recurrent Chunking Mechanisms for Long-Text Machine Reading Comprehension. Technical report.
- Guimarey, A. (2020). Beneficios y riesgos de migrar una arquitectura monolítica a microservicios. Technical report.
- Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M.-W. (2020). REALM: Retrieval-Augmented Language Model Pre-Training.
- Han, Y., Liu, C., and Wang, P. (2023). A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge.
- Heng, H. (2023). Flowise - Low code LLM Apps Builder.
- Izacard, G. and Grave, E. (2020). Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering.
- Jagtap, S., Garcia-Garcia, G., and Rahimifard, S. (2021). Optimisation of the resource efficiency of food manufacturing via the Internet of Things. *Computers in Industry*, 127.
- Jana, R., Chowdhury, A. R., and Islam, M. (2014). Optical Character Recognition from Text Image. *International Journal of Computer Applications Technology and Research*, 3(4):240-244.
- Jeong, C. (2024). A Study on the Implementation of Generative AI Services Using an Enterprise Data-Based LLM Application Architecture. Technical report.
- Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. (2020). Dense Passage Retrieval for Open-Domain Question Answering.

- Khurana, D., Koli, A., Khatter, K., and Singh, S. (2023). Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*, 82(3):3713–3744.
- Kokkonen, H., Lovén, L., Motlagh, N. H., Kumar, A., Partala, J., Nguyen, T., Pujol, V. C., Kostakos, P., Leppänen, T., González-Gil, A., Sola, E., Angulo, I., Liyanage, M., Bennis, M., Tarkoma, S., Dustdar, S., Pirttikangas, S., and Rieki, J. (2022). Autonomy and Intelligence in the Computing Continuum: Challenges, Enablers, and Future Directions for Orchestration.
- Kumar, P. and Brown, J. D. (2023). Anchoring Global Security: Autonomous Shipping with Mind Reading AI, GPT-core and MAMBA-core Agents, RAG-Fusion, AI Communities, Hive-AI, and the Human Psyche.
- Kumar Gupta, M., Kr, M., and Bodol, D. (2073). A Study on Indexes and Index Structures. Technical Report 1.
- Leonel Suarez, S. (2022). Análisis de patrones de resiliencia en una arquitectura basada en micro-servicios. Technical report, 11.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., and Kiela, D. (2005). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.
- Linares, B. M., German, J., Urrutia, M. A., and William, C. (2022). Metodología De Conversión De Aplicaciones Monolíticas A Microservicios Desplegable En La Nube Para Pequeñas Empresas. Technical report.
- Liu, Y. (2023). Implications of generative artificial intelligence for the development of the media industry. *Advances in Engineering Innovation*, 1(1):29–36.
- Lyu, C., Wu, M., Wang, L., Huang, X., Liu, B., Du, Z., Shi, S., and Tu, Z. (2023). Macaw-LLM: Multi-Modal Language Modeling with Image, Audio, Video, and Text Integration.
- Mainas, N., Bouraimis, F., Karavisileiou, A., and Petrakis, E. G. M. (2023). Annotated OpenAPI Descriptions and Ontology for REST Services. *International Journal on Artificial Intelligence Tools*, 32(06).
- Malenfer, M., Sarrey, M., Clerté, J., Hery, M., Bieri, M., Braunschweig, B., Chatellier, R., Fates, N., Halluin, S., de Jouvenel, F., Mandinaud, V., Munoz, J., Olympio, A., Silvestre, T., and Soupizet, J.-F. (2023). Artificial intelligence in the service of health and safety at work: Perspectives and challenges from now to 2035 - A prospective study. *Qeios*.
- Mani, N., Singh, A., and Nimmagadda, S. L. (2020). An IoT Guided Healthcare Monitoring System for Managing Real-Time Notifications by Fog Computing Services. In *Procedia Computer Science*, volume 167, pages 850–859. Elsevier B.V.

- McIvor, A. (2020). Guardians of Workers' Bodies? Trade Unions and the History of Occupational Health and Safety. *Labour History*, 119(1):1–30.
- Meta (2023). Introducing LLaMA: A foundational, 65-billion-parameter language model.
- Morgan, J. (2023). Ollama — Get up and running with large language models locally.
- Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., and Mian, A. (2023). A Comprehensive Overview of Large Language Models.
- Nexla (2024). Vector Databases: Tutorial, Best Practices & Examples — Nexla.
- Ospina, J. P. and Aristizábal, E. (2021). Aplicación de inteligencia artificial y técnicas de aprendizaje automático para la evaluación de la susceptibilidad por movimientos en masa. *Revista Mexicana de Ciencias Geológicas*, 38:43–54.
- Pan, J. J., Wang, J., and Li, G. (2023). Survey of Vector Database Management Systems.
- Pesantes Robles, C. A. (2023). Automatización De Pruebas De Aceptación En El Proceso De Desarrollo De Software. Technical report.
- Rathi, V. K., Rajput, N. K., Mishra, S., Grover, B. A., Tiwari, P., Jaiswal, A. K., and Hossain, M. S. (2021). An edge AI-enabled IoT healthcare monitoring system for smart cities. *Computers and Electrical Engineering*, 96.
- Rebolledo Ruiz, S. (2024). Performance Analysis of a Bicameral Cache in Vector Architectures. Technical report.
- Richards, M. W. M., Lange, B., and Ford, N. (2021). Fundamentals of software architecture: an engineering approach.
- Rikhotso, O., Morodi, T. J., and Masekamani, D. M. (2022). Occupational Health and Safety Statistics as an Indicator of Worker Physical Health in South African Industry.
- Rosner, D. and Markowitz, G. (2020). A Short History of Occupational Safety and Health in the United States. *American Journal of Public Health*, 110(5):622–628.
- Ruiz Barea, R., Del, P., Rodrigo, C., and Font, G. (2023). Protección de APIs REST. Technical report.
- Saito, K., Sohn, K., Lee, C.-Y., and Ushiku, Y. (2024). Unsupervised LLM Adaptation for Question Answering.
- Sarker, I. H. (2024). LLM Potentiality and Awareness: A Position Paper from the Perspective of Trustworthy and Responsible AI Modeling.
- Saygılı, Y. S. and Çakmak, B. (2022). Occupational Health and Safety in Agricultural Production. *Turkish Journal of Agriculture - Food Science and Technology*, 9(sp):2418–2426.

- Taipalus, T. (2024). Vector database management systems: Fundamental concepts, use-cases, and current challenges. *Cognitive Systems Research*, 85:101216.
- Taleb, I., Serhani, M. A., and Dssouli, R. (2018). Big Data Quality Assessment Model for Unstructured Data. In *Proceedings of the 2018 13th International Conference on Innovations in Information Technology, IIT 2018*, pages 69–74. Institute of Electrical and Electronics Engineers Inc.
- Tomás López-Solaz, José A. Troyano, F. Javier Ortega, and Fernando Enríquez (2016). *Procesamiento del Lenguaje Natural*.
- Törnberg, P. (2023). How to use LLMs for Text Analysis.
- Törnberg, P. (2024). Best Practices for Text Annotation with Large Language Models. Technical report.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023a). LLaMA: Open and Efficient Foundation Language Models.
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Michael, E., Ranjan, S., Xiaoqing, S., Tan, E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. (2023b). Llama 2: Open Foundation and Fine-Tuned Chat Models. Technical report.
- Trebejo Loayza, W. J. (2023). Herramienta para el modelado y generación de código de Arquitecturas de Software basadas en Microservicios.
- Unstructured (2024). Understanding What Matters for LLM Ingestion and Preprocessing – Unstructured.
- Vallvé, A. (2023). Integración de datos de procesos de negocio y organizacionales de bases de datos NoSQL.
- Vargas, J., Augusto, P., Guerrero, F., Victoria, M., and Brando, J. (2023). Solución de automatización de pruebas de calidad en empresa bancaria para reducir tiempos de certificación con herramientas tecnológicas de software. Technical report.
- Vasnetsov, A. (2021). Qdrant - Vector Database.

- Vasquez Figueroa, M. T. (2023). Análisis Comparativo De Los Algoritmos De Balanceo De Cargas Round Robin Y Least Connection De Ha-Proxy En Los Sistemas De Gestion De Base De Datos. Technical report.
- Velasco Navas, J. P. (2024). Construcción de una plataforma de comercio electrónico basada en microservicios para la gestión, entrega y rastreo de productos a domicilio. Technical report.
- Velásquez-Henao, J. D., Franco-Cardona, C. J., and Cadavid-Higuita, L. (2023). Prompt Engineering: a methodology for optimizing interactions with AI-Language Models in the field of engineering. *DYNA*, 90(230):9–17.
- White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., and Schmidt, D. C. (2023). A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT.
- Yusuf Alan, A., Karaarslan, E., and Aydin, O. (2024). A RAG-based Question Answering System Proposal for Understanding Islam: MufassirQAS LLM. Technical report.

Apéndice A: Fuente de Datos de SST

Para el caso de estudio que se va a desarrollar, es necesario primero explicar la fuente de datos con la que se trabajará y justificar su uso. Los especialistas en Seguridad y Salud en el Trabajo han seleccionado como material de trabajo el documento correspondiente a la matriz de riesgos de SST. Esta matriz se considera una herramienta fundamental para la gestión de riesgos laborales y consiste en una tabla que permite identificar, evaluar y priorizar los riesgos a los que están expuestos los trabajadores en un lugar de trabajo.

La matriz de riesgos en SST ha sido seleccionada como la fuente de datos para el desarrollo del caso de estudio porque facilita a los trabajadores de SST la identificación de peligros y factores de riesgo, la evaluación del nivel de riesgo de cada peligro y factor, la priorización de los riesgos según su nivel de gravedad, el establecimiento de medidas de control para eliminar o reducir los riesgos y el monitoreo de la eficacia de estas medidas.

Además, la implementación de la matriz de riesgo en el campo de SST ofrece beneficios significativos, como mejorar la Seguridad y Salud en el Trabajo al identificar y controlar los riesgos laborales, reducir los accidentes de trabajo y las enfermedades profesionales, aumentar la productividad de los trabajadores, disminuir los costos asociados a accidentes de trabajo y enfermedades profesionales y cumplir con los requisitos legales en materia de SST.

En resumen, la matriz de riesgos de SST es una herramienta esencial para la gestión de riesgos laborales, ya que su uso permite identificar, evaluar y priorizar los riesgos a los que están expuestos los trabajadores y establecer medidas de control para eliminar o reducir dichos riesgos.

Para el caso de estudio, la fuente de datos es un archivo de tipo .XLSX en el cual se almacena información correspondiente al área de Seguridad y Salud en el Trabajo para diferentes instituciones educativas del Valle del Cauca. Cada registro dentro del archivo contiene información sobre los aspectos de SST que se evaluaron en diferentes roles laborales dentro de cada institución. A continuación, se presenta una tabla [A.1](#) donde se listan las 58 columnas que componen el archivo, el tipo de datos que almacenan y una descripción de cada una.

COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ID	STRING	Identificador único del registro.
INFORMACIÓN GENERAL	STRING	Columna vacía para delimitar la categoría a la que pertenecen las columnas siguientes.
DEPARTAMENTO	STRING	Departamento donde se ubica la institución.
MUNICIPIO	STRING	Municipio donde se ubica la institución.
NOMBRE INSTITUCIÓN EDUCATIVA	STRING	Nombre de la institución educativa a la que se le aplicó la evaluación de riesgos.
CÓDIGO DE INSTITUCIÓN	STRING	Identificador único de la institución educativa.
NIT	INTEGER	Número de Identificación Tributaria (NIT) de la institución educativa.
NOMBRE Y APELLIDOS RECTOR	STRING	Nombre y apellidos del rector a cargo de la institución educativa.
ZONA	STRING	Zona en la que se ubica la institución. Puede ser RURAL o URBANA.
FECHA ELABORACIÓN	DATE	Fecha en la que se registró la evaluación de riesgo a la institución. El formato de la fecha es dd/mm/yyyy.
HORA ELABORACIÓN	TIME	Hora en la que se registró la evaluación de riesgo a la institución. El formato de la hora es hh:mm:ss.
PROCESO	STRING	El tipo de proceso al que pertenecen las actividades que realiza la persona en el cargo laboral al que se le evalúa el riesgo dentro de la institución educativa. Este puede ser ADMINISTRATIVO u OPERATIVO.
CARGO	STRING	Cargo laboral dentro de la institución al que se le realiza la evaluación de riesgo.
ACTIVIDADES	STRING	Una o más actividades que se realizan dentro del cargo laboral.
PELIGROS	STRING	Columna vacía para delimitar la categoría a la que pertenecen las columnas siguientes.
CLASIFICACIÓN	STRING	Categoría en la que se clasifica el tipo de peligro al que se está expuesto según el cargo y las actividades que se realizan. Este puede ser BIOLÓGICO, FÍSICO, QUÍMICO, PSICOSOCIAL, BIOMECÁNICOS, CONDICIONES DE SEGURIDAD o FENÓMENOS NATURALES.

Continúa en la siguiente página

COLUMNA	TIPO DE DATO	DESCRIPCIÓN
DESCRIPCIÓN	STRING	Descripción del peligro al que se está expuesto según según la clasificación de la columna anterior. Estos pueden ser Virus, Bacterias, Hongos, Rickettsias, Parásitos, Picaduras, Mordeduras, Fluidos o excrementos, Ruido, Iluminación, Vibración, Temperaturas extremas, Presión atmosférica, Radiaciones ionizantes, Radiaciones no ionizantes, Polvos orgánicos / inorgánicos, Fibras, Líquidos, Gases y vapores, Humos metálicos / no metálicos, Material particulado, Gestión organizacional, Características de la organización, Características del grupo social del trabajo, Condiciones de la tarea, Interfase persona tarea, Jornada de trabajo, Postura, Esfuerzo, Movimiento repetitivo, Manipulación manual de cargas, Mecánico, Eléctrico, Locativo, Tecnológico, Accidentes de tránsito, Públicos, Trabajo en alturas, Espacios confinados, Sismo, Terremoto, Vendaval, Inundación, Derrumbe, Precipitaciones.
EFFECTOS POSIBLES	STRING	Clasificación del nivel de daño que puede ocasionar el peligro identificado. Este puede ser Daño leve, Daño moderado o Daño extremo.
FUENTE GENERADORA	STRING	Razón o razones que describen la fuente generadora del peligro identificado.
RUTINARIO	STRING	Establece si la fuente generadora del peligro se presenta de forma constante o esporádicamente. Esta columna sólo toma dos valores: SÍ o NO.
TIEMPO DE EXPOSICIÓN (Hr)	INTEGER	Cantidad de tiempo en horas que se está expuesto al peligro y fuente identificados.
CONTROLES EXISTENTES	STRING	Columna vacía para delimitar la categoría a la que pertenecen las columnas siguientes.
FUENTE	STRING	Establece si la institución cuenta con controles para la fuente de los peligros identificados. Esta columna sólo toma dos valores: SÍ o NO.

Continúa en la siguiente página

COLUMNA	TIPO DE DATO	DESCRIPCIÓN
MEDIO	STRING	Establece si la institución cuenta con controles para el medio en el que suceden los peligros identificados. Esta columna sólo toma dos valores: SÍ o NO.
TRABAJADOR	STRING	Establece si la institución cuenta con controles para el trabajador relacionado a los peligros identificados. Esta columna sólo toma dos valores: SÍ o NO.
DESCRIPCIÓN DE LOS MEDIDAS DE PREVENCIÓN Y CONTROL EXISTENTES	STRING	Describe la medida o medidas existentes que posee la institución frente a los peligros identificados.
EVALUACIÓN DEL RIESGO	STRING	Columna vacía para delimitar la categoría a la que pertenecen las columnas siguientes.
NIVEL DEFICIENCIA	STRING	Calificación cualitativa del nivel de deficiencia del riesgo identificado. Este puede ser BAJO (B), MEDIO (M), ALTO (A) o MUY ALTO (MA).
ND VALOR	INTEGER	Calificación cuantitativa del nivel de deficiencia del riesgo identificado. Este puede ser 0, 2, 6 o 10, y corresponden respectivamente a las calificaciones cualitativas de BAJO (B), MEDIO (M), ALTO (A) o MUY ALTO (MA).
ND INTERPRETACIÓN	STRING	Corresponde a la descripción equivalente a la calificación tanto cualitativa como cuantitativa que se le asignó al nivel de deficiencia.
NIVEL DE EXPOSICIÓN	STRING	Calificación cualitativa del nivel de exposición del riesgo identificado. Este puede ser ESPORADICA (EE), OCASIONAL (EO), FRECUENTE (EF) o CONTINUA (EC).
NE VALOR	INTEGER	Calificación cuantitativa del nivel de exposición del riesgo identificado. Este puede ser 1, 2, 3 o 4 y corresponden respectivamente a las calificaciones cualitativas de ESPORADICA (EE), OCASIONAL (EO), FRECUENTE (EF) o CONTINUA (EC).

Continúa en la siguiente página

COLUMNA	TIPO DE DATO	DESCRIPCIÓN
NE INTERPRETACIÓN	STRING	Corresponde a la descripción equivalente a la calificación tanto cualitativa como cuantitativa que se le asignó al nivel de exposición.
NIVEL DE PROBABILIDAD	STRING	Calificación cualitativa del nivel de probabilidad del riesgo identificado. Este puede ser BAJO (B), MEDIO (M), ALTO (A) o MUY ALTO (MA).
NP VALOR	INTEGER	Calificación cuantitativa del nivel de probabilidad del riesgo identificado. Este puede ser 2, 4, 6, 8, 10, 12, 18, 20, 24, 30 o 40.
INTERPRETACIÓN NIVEL PROBABILIDAD	STRING	Corresponde a la descripción equivalente a la calificación tanto cualitativa como cuantitativa que se le asignó al nivel de probabilidad.
NIVEL DE CONSECUENCIA	STRING	Calificación cualitativa del nivel de consecuencia del riesgo identificado. Este puede ser LEVE (L), GRAVE (G), MUY GRAVE (MG), MORTAL o CATASTROFICO (M).
NC VALOR	INTEGER	Calificación cuantitativa del nivel de consecuencia del riesgo identificado. Este puede ser 10, 25, 60 o 100 y corresponden respectivamente a las calificaciones cualitativas de LEVE (L), GRAVE (G), MUY GRAVE (MG), MORTAL o CATASTROFICO (M).
NC INTERPRETACIÓN	STRING	Corresponde a la descripción equivalente a la calificación tanto cualitativa como cuantitativa que se le asignó al nivel de consecuencia.
NIVEL DE RIESGO	INTEGER	Calificación cuantitativa del nivel del riesgo identificado. Este puede ser 20, 40, 50, 60, 80, 100, 150, 200, 240, 250, 360, 400, 480, 500, 600, 800, 1200, 1440, 2000, 2400 o 4000.
INTERPRETACIÓN DEL NR	STRING	Corresponde al nivel equivalente a la calificación tanto cualitativa como cuantitativa que se le asignó al riesgo identificado. Este puede ser I, II, III, o IV.
SIGNIFICADO NR	STRING	Corresponde a la descripción equivalente al nivel de riesgo que se identificó.
VALORACIÓN RIESGO	STRING	Columna vacía para delimitar la categoría a la que pertenecen las columnas siguientes.

Continúa en la siguiente página

COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ACEPTABILIDAD DEL RIESGO	STRING	Calificación cualitativa de la aceptabilidad del riesgo identificado. Este puede ser NO ACEPTABLE, ACEPTABLE CON CONTROL ESPECÍFICO, MEJORABLE o ACEPTABLE y corresponden respectivamente a los niveles de riesgo de I, II, III o IV.
SIGNIFICADO DE ACEPTABILIDAD DEL RIESGO	STRING	Corresponde a la descripción equivalente a la calificación le asignó a la aceptabilidad del riesgo identificado.
CRITERIOS PARA ESTABLECER CONTROLES	STRING	Columna vacía para delimitar la categoría a la que pertenecen las columnas siguientes.
NÚMERO DE EXPUESTOS	INTEGER	Cantidad de individuos que se pueden ver expuestos por el riesgo identificado.
PEOR CONSECUENCIA	STRING	Califica de manera cualitativa la peor consecuencia a la que puede llegar el riesgo identificado. Este puede ser LEVE, GRAVE, MUY GRAVE, MORTAL o CATASTROFICO.
EXISTENCIA REQUISITO LEGAL ESPECIFICO ASOCIADO	STRING	Establece si la institución existe o no un requisito legal específico a la tarea que se está evaluando para tener parámetros de priorización en la implementación de las medidas de intervención. Esta columna sólo toma dos valores: SÍ o NO.
MEDIDAS DE PREVENCIÓN Y CONTROL	STRING	Columna vacía para delimitar la categoría a la que pertenecen las columnas siguientes.
1. ELIMINACIÓN	STRING	Establece si en la institución es necesarios realizar medidas de prevención y control de tipo de eliminación. Esta columna sólo toma dos valores: SÍ o NO.
2. SUSTITUCIÓN	STRING	Establece si en la institución es necesarios realizar medidas de prevención y control de tipo de sustitución. Esta columna sólo toma dos valores: SÍ o NO.
3. CONTROLES DE INGENIERÍA	STRING	Establece si en la institución es necesarios realizar medidas de prevención y control de tipo de controles de ingeniería. Esta columna sólo toma dos valores: SÍ o NO.

Continúa en la siguiente página

COLUMNA	TIPO DE DATO	DESCRIPCIÓN
4. CONTROLES ADMINISTRATIVOS	STRING	Establece si en la institución es necesarios realizar medidas de prevención y control de tipo de controles administrativos. Esta columna sólo toma dos valores: SÍ o NO.
5. EQUIPOS / ELEMENTOS DE PROTECCION PERSONAL Y COLECTIVO	STRING	Establece si en la institución es necesarios realizar medidas de prevención y control de tipo de equipos/elementos de protección personal y colectivo. Esta columna sólo toma dos valores: SÍ o NO.
DESCRIPCIÓN DE LAS MEDIDAS A IMPLEMENTAR	STRING	Descripción de las medidas que se deben implementar frente al riesgo identificado en la institución.
Informe	STRING	Columna vacía para delimitar la categoría a la que pertenecen las columnas siguientes.
Coordenadas	STRING	Coordenadas de geolocalización de la institución educativa a la que se le aplicó la evaluación de riesgos.

Tabla A.1: Descripción de las columnas que conforman la matriz de riesgos de SST

Es importante mencionar que la Corporación Talentum se reserva los datos sensibles presentes en la matriz de riesgo de SST, debido a que esta información se considera de carácter confidencial. Por tanto, algunos de los campos fueron modificados para conservar el anonimato de los datos. Las columnas que fueron alteradas para preservar la anonimidad de los datos incluyen:

- **Nombre de la Institución Educativa:** Para mantener el anonimato de las instituciones educativas que participan en la evaluación de riesgos, los nombres de las instituciones fueron cambiados por “IE ANÓNIMA #”, donde “#” hace referencia al número de la institución. Por ejemplo: IE ANÓNIMA 1, IE ANÓNIMA 2, IE ANÓNIMA 3, y así sucesivamente.
- **Código de Institución:** Para mantener el anonimato de las instituciones educativas que participan en la evaluación de riesgos, todos los códigos de las instituciones fueron cambiados por “A#”, donde “#” hace referencia al número de la institución. Ejemplos incluyen A1, A2, A3, y así sucesivamente.
- **NIT:** Para preservar el anonimato de las instituciones educativas participantes, todos los Números de Identificación Tributaria (NIT) fueron cambiados por un numero entero consecutivo “1, 2, 3”.
- **Nombre y Apellidos del Rector:** Para mantener el anonimato de los rectores de las instituciones educativas que participan en la evaluación de riesgos, todos los nombres y apellidos

de los rectores fueron cambiados por “RECTOR ANÓNIMO #”, donde “#” hace referencia al número de la institución. Ejemplos incluyen RECTOR ANÓNIMO 1, RECTOR ANÓNIMO 2, RECTOR ANÓNIMO 3, y así sucesivamente.

- **Coordenadas:** Para mantener el anonimato de las instituciones educativas que participan en la evaluación de riesgos, todas las coordenadas de geolocalización fueron cambiadas por “0.000000, 0.000000”.

Apéndice B: Architecture Decision Records

Título	ADR-INT-01 Administrador de documentos
Estado	Aceptado
Contexto	Es necesario poder tener un control sobre los archivos que se cargan a la base de datos vectorial de manera flexible y fácil para el usuario, además que debe ser capaz de acceder a directorios dentro de un servidor desplegado en la nube.
Decisión	Se decide implementar Filebrowser ya que es un administrador de archivos web, open source y ligero que permite administrar fácilmente archivos desde la nube o en un servidor local.
Consecuencias	<ul style="list-style-type: none"> ▪ Interfaz de usuario para el manejo de archivos, intuitiva, fácil de usar y navegar. ▪ Capacidad para subir, eliminar, descargar archivos, crear carpetas, mover archivos y renombrarlos con facilidad. ▪ Creación de varios usuarios donde cada usuario puede tener su propio directorio. ▪ Aplicación ligera que no consume muchos recursos del sistema. ▪ Seguridad y protección de los archivos frente a accesos no autorizados.

Tabla B.1: ADR de interoperabilidad Núm. 01: Administrador de documentos

Título	ADR-INT-02 Orquestador LLM
Estado	Aceptado
Contexto	Se necesita poder articular por medio de un orquestador los diferentes componentes que componen el flujo de trabajo de los LLM y los agentes de IA, además que que estos puedan desplegarse de manera automatizada y se pueda embeber la solución dentro de una aplicación web.
Decisión	Se decide usar Flowise debido a que es una herramienta open source que permite a los desarrolladores construir aplicaciones de modelos de lenguaje (LLM) de forma rápida y sencilla por medio de flujos de trabajo automatizados.
Consecuencias	<ul style="list-style-type: none"> ▪ Interfaz de usuario visual que hace que sea fácil de usar incluso para aquellos con poca experiencia en programación. ▪ Permite a los desarrolladores crear aplicaciones LLM rápidamente sin necesidad de escribir código complejo. ▪ Amplia gama de plantillas para aplicaciones LLM comunes, como chatbots, asistentes virtuales y generadores de texto. ▪ Creación de flujos de trabajos que permiten la integración de diferentes componentes para el desarrollo de aplicación de LLM con una alto catálogo de componentes compatibles.

Tabla B.2: ADR de interoperabilidad Núm. 02: Orquestador de flujos LLM

Título	ADR-REN-01 Base de datos vectorial
Estado	Aceptado
Contexto	Es necesario contar con una base de datos vectorial donde se puedan almacenar documentos convertidos en vectores para permitir a los LLM poder acceder a nuevos conocimientos por medio de los documentos cargados.
Decisión	Se decide utilizar Qdrant, el cual es un motor de bases de datos vectorial open source, porque permite almacenar fácilmente datos vectoriales. Es una solución rápida, escalable y rentable para buscar datos similares en grandes conjuntos de datos.
Continúa en la siguiente página	

Tabla B.3 – Continúa de la página anterior

Título	ADR-REN-01 Base de datos vectorial
Consecuencias	<ul style="list-style-type: none"> ▪ Búsqueda de datos similares de manera rápida y eficiente, incluso en grandes conjuntos de datos. ▪ Escalabilidad horizontal para manejar grandes cargas de trabajo. ▪ Interfaz gráfica que permite explorar los vectores almacenados en el espacio vectorial. ▪ Capacidad para exportar e importar copias de seguridad de los conjuntos de datos vectoriales almacenados. ▪ Mejora rendimiento de la IA al tener la información vectorizada

Tabla B.3: ADR de rendimiento Núm. 01: Base de datos vectorial

Título	ADR-REN-02 Base de datos para almacenar Chats
Estado	Aceptado
Contexto	El LLM debe estar en la capacidad de poder recordar elementos de conversaciones previas con los usuarios con el fin de poder utilizar estos conocimientos para mejorar la calidad y precisión de las respuestas.
Decisión	Se decide implementar Amazon DynamoDB porque es una base de datos NoSQL totalmente administrada y escalable que ofrece un rendimiento rápido y predecible con costos bajos. Es una opción popular para aplicaciones web y móviles que requieren un acceso rápido y confiable a grandes conjuntos de datos.
Continúa en la siguiente página	

Tabla B.4 – Continúa de la página anterior

Título	ADR-REN-02 Base de datos para almacenar Chats
Consecuencias	<ul style="list-style-type: none"> ▪ Almacenamiento de datos en documentos JSON o estructuras de clave-valor, lo que la hace flexible y escalable. ▪ Escalabilidad automática para manejar cualquier carga de trabajo, sin necesidad de aprovisionar o administrar servidores. ▪ Rendimiento de milisegundos de un solo dígito para lecturas y escrituras. ▪ Alta disponibilidad y durabilidad de los datos con replicación en varias regiones. ▪ Mejora el rendimiento al consultar el chat previamente guardado ▪ Capacidad del LLM para recordar conversaciones que haya tenido con el usuario en sesiones anteriores. ▪ Bases de datos NoSQL pero de clave-valor.

Tabla B.4: ADR de rendimiento Núm. 02: Base de datos para almacenar chat en memoria

Título	ADR-REN-03 Herramienta de monitoreo Langfuse
Estado	Aceptado
Contexto	Monitorear los prompts y las conversaciones de los usuarios con el modelo de LLM desplegado es importante ya que permite identificar y prevenir problemas, así como mejorar la experiencia del usuario, asegurar la seguridad y obtener información valiosa.
Decisión	Se decide usar Langfuse ya que esta es una herramienta open source que permite almacenar, observar y analizar diferentes métricas sobre los prompts y conversaciones de los usuarios con los modelos de LLM que se encuentren desplegados en diferentes aplicaciones.

Continúa en la siguiente página

Tabla B.5 – Continúa de la página anterior

Título	ADR-REN-03 Herramienta de monitoreo Langfuse
Consecuencias	<ul style="list-style-type: none"> ▪ Supervisión de los prompts y las conversaciones de los usuarios con las aplicaciones de LLM con un conjunto de métricas. ▪ Obtención del contexto completo de la ejecución del LLM, incluyendo llamadas a la API, contexto, indicaciones, conversaciones y más. ▪ Monitoreo de latencia del flujo del orquestador, permite ver que parte del flujo tiene mas latencia. ▪ La herramienta open source permite monitorear el flujo y las sesiones que el orquestador LLM esta procesando.

Tabla B.5: ADR de rendimiento Núm. 03: Monitoreo de rendimiento de flujos de orquestación LLM

Título	ADR-COS-01 Servicio serverless para desplegar LLM
Estado	Aceptado
Contexto	Se requiere desplegar un LLM sin impactar a la Corporación Talentum en costos elevados, que facilite el despliegue y el procesamiento de diferentes modelos.
Decisión	Se decide usar AWS Bedrock ya que permite acceder a modelos de IA de última generación a través de una única API, además de experimentar, personalizar y desarrollar aplicaciones de IA de forma segura y responsable sin impactar el costo y gestionar infraestructura dado que es un servicio serverless.
Continúa en la siguiente página	

Tabla B.6 – Continúa de la página anterior

Título	ADR-COS-01 Servicio serverless para desplegar LLM
Consecuencias	<ul style="list-style-type: none"> ▪ Ejecución de inferencias de modelos enviando peticiones utilizando diferentes configuraciones y modelos fundacionales para generar respuestas. ▪ Rendimiento aprovisionado para un modelo fundacional con el fin de realizar inferencias sobre los modelos de manera más eficiente y a precios reducidos. ▪ Evaluación de los resultados de los distintos modelos con conjuntos de datos de peticiones integrados o personalizados para determinar el modelo que mejor se adapte a la aplicación.

Tabla B.6: ADR de costos Núm. 01: Servicio serverless para desplegar LLM

Título	ADR-COS-02 Servicio serverless para desplegar Embeddings
Estado	Aceptado
Contexto	Se necesita implementar embeddings que transforman las palabras y frases en representaciones numéricas que las máquinas pueden entender y manipular. Esto permite que los modelos de IA realicen tareas como el análisis de sentimientos, la clasificación de texto y la traducción automática.
Decisión	Se toma la decisión de utilizar AWS Bedrock embeddings debido a que este permite generar representaciones numéricas de texto que se utilizan en el procesamiento del lenguaje natural (PNL) y el aprendizaje automático (ML). En otras palabras, transforma el texto en un código que las computadoras pueden entender y manipular más fácilmente. Este servicio de AWS es serverless lo cual no impacta en gestión de infraestructura y también reduce los costos por el uso del servicio.
Continúa en la siguiente página	

Tabla B.7 – Continúa de la página anterior

Título	ADR-COS-02 Servicio serverless para desplegar Embeddings
Consecuencias	<ul style="list-style-type: none"> ▪ Encontrar texto con significado similar. ▪ Clasificación de texto. ▪ Agrupamiento de texto similar. ▪ Almacenamiento de documentos en bases de datos vectoriales. ▪ Costos reducidos en infraestructura

Tabla B.7: ADR de costos Núm. 02: Servicio serverless para desplegar Embeddings

Título	ADR-COS-03 Herramienta de monitoreo Lunary
Estado	Aceptado
Contexto	Se necesita monitorear el uso y los costos de los diferentes modelos de LLM desplegados en la aplicación, es importante este monitoreo puesto que permite a la empresa establecer cuotas de cobro dependiendo de la cantidad de uso e interacción que tienen los diferentes LLM desplegados.
Decisión	Se decide usar Lunary porque su versión open source ofrece una amplia gama de métricas para monitorear la utilización de recursos de LLM además de identificar la cantidad de usos y costos que tienen cada uno de estos modelos desplegados.
Consecuencias	<ul style="list-style-type: none"> ▪ Establecimiento de cuotas de cobro por modelo y uso. ▪ Escalabilidad para monitorear aplicaciones de LLM de cualquier tamaño. ▪ Facilidad de uso que no requiere experiencia previa en monitoreo de sistemas. ▪ Análisis financiero sobre el estado de cobros de los modelos. ▪ Permite tener un mejor control del gasto por modelo que se esta usando.
Continúa en la siguiente página	

Tabla B.8 – Continúa de la página anterior

Título	ADR-COS-03 Herramienta de monitoreo Lunary
	Tabla B.8: ADR de costos Núm. 03: Monitoreo de costos por LLM

Apéndice C: Tácticas de arquitectura de software

Atributo de calidad	Táctica	Descripción	Justificación	Drivers
Rendimiento	Manage Work Requests	Esta táctica se centra en reducir el número de solicitudes que ingresan al sistema para realizar trabajos. Esto incluye gestionar la llegada de eventos mediante un Acuerdo de Nivel de Servicio (SLA) que especifica la tasa máxima de eventos.	Los componentes del sistema permiten la configuración de un límite de tasa, crucial para controlar y prevenir el exceso de solicitudes. Configurando un máximo de 20 mensajes por 60 segundos y un mensaje de “Cuota Excedid”, al exceder el límite, se mantiene la estabilidad del sistema y se evita la sobrecarga, asegurando un funcionamiento eficiente y continuo.	Rendimiento
Continúa en la página siguiente				

Tabla C.1 – continuación de la página anterior

Atributo de calidad	Táctica	Descripción	Justificación	Drivers
	Maintain Multiple Copies of Computations	Esta táctica implica la replicación de servicios para reducir la contención en las solicitudes de servicio. Usualmente se utiliza en arquitecturas de microservicios y servidores web replicados, con un balanceador de carga distribuyendo las solicitudes.	El uso de Kubernetes, alineado con las políticas de la Corporación Talentum, facilita la gestión de servicios replicados, mejorando la disponibilidad y escalabilidad del sistema. Esta capacidad de replicación asegura que las solicitudes de los usuarios se procesen de manera eficiente, incluso bajo condiciones de alta demanda, manteniendo así un rendimiento óptimo.	Rendimiento, costos

Continúa en la página siguiente

Tabla C.1 – continuación de la página anterior

Atributo de calidad	Táctica	Descripción	Justificación	Drivers
Interoperabilidad	Adhere to Standards	Esta táctica se centra en la estandarización de las implementaciones de sistemas como un facilitador primordial de la integrabilidad y la interoperabilidad, abarcando plataformas y proveedores. Los estándares pueden variar significativamente en términos del alcance de lo que prescriben, algunos se enfocan en definir sintaxis y semánticas de datos, mientras que otros incluyen descripciones más ricas, como aquellas que describen protocolos que contienen semánticas de comportamiento y temporales.	Para la Corporación Talentum, se utilizan estándares de industria para el uso de Inteligencia Artificial, como los formatos GPT-Generated Model Language (GGML) y GPT-Generated Unified Format (GGUF). La adopción de estos estándares tiene como objetivo limitar el espectro de interoperabilidad, asegurando así una integración más fluida y eficiente al incorporar componentes externos y facilitando la comunicación entre distintos sistemas y plataformas tecnológicas.	Interoperabilidad, costos

Continúa en la página siguiente

Tabla C.1 – continuación de la página anterior

Atributo de calidad	Táctica	Descripción	Justificación	Drivers
	Discover	Esta implica el uso de un servicio de descubrimiento, que es un catálogo de direcciones relevantes. Esto es útil cuando es necesario traducir de una forma de dirección a otra, cuando la dirección de destino puede haber sido vinculada dinámicamente, o cuando hay múltiples objetivos. Este mecanismo permite que aplicaciones y servicios se localicen entre sí. Las entradas en el servicio de descubrimiento deben registrarse y desregistrarse según su relevancia, lo cual puede ser estático o dinámico.	En el contexto de la Corporación Talentum, los servicios serán descubiertos mediante reglas o dominios especificados en la nube, facilitando que sean encontrados sin utilizar directamente direcciones IP y puertos. Esta táctica permite una mayor visibilidad y flexibilidad en la localización de servicios, contribuyendo a una integración eficaz y reduciendo la dependencia entre servicios que deben cooperar sin conocimiento previo el uno del otro.	Interoperabilidad

Continúa en la página siguiente

Tabla C.1 – continuación de la página anterior

Atributo de calidad	Táctica	Descripción	Justificación	Drivers
	Orchestrate	Orchestrate es una táctica que utiliza un mecanismo de control para coordinar y gestionar la invocación de servicios particulares de modo que puedan permanecer inconscientes unos de otros. La orquestación facilita la integración de un conjunto de servicios reutilizables y poco acoplados para crear un sistema que satisfaga una nueva necesidad. Esto reduce los costos de integración y fomenta la reutilización y la agilidad, conduciendo a procesos empresariales más flexibles que pueden ser gestionados bajo la filosofía del BPM (Business Process Management).	Para lograr una interoperabilidad no complicada, la Corporación Talentum emplea componentes orquestadores que permiten crear flujos para facilitar las integraciones. Esta táctica centraliza las dependencias en el mecanismo de orquestación, eliminando las dependencias explícitas entre los componentes y reduciendo las distancias semánticas y sintácticas, lo que resulta en una mayor eficiencia y flexibilidad en la gestión de los procesos empresariales.	Interoperabilidad, costos

Tabla C.1: Tácticas

Apéndice D: Patrones arquitectónicos de software

Atributo de calidad	Patrón	Descripción	Justificación	Drivers
Rendimiento	Load Balancer	Un intermediario que distribuye solicitudes de clientes entre un conjunto de servidores para equilibrar la carga Bass et al. (2021) .	La implementación de un patrón de Load Balancer permite distribuir las solicitudes entrantes entre varios servidores, lo que evita que cualquier servidor único se sobrecargue. Al equilibrar la carga, el sistema puede manejar un mayor volumen de solicitudes simultáneas, reduciendo así los tiempos de espera y mejorando la capacidad de respuesta del servicio. Este patrón es esencial para mantener un rendimiento óptimo sin sobrecargar los recursos.	Rendimiento

Continúa en la página siguiente

Tabla D.1 – continuación de la página anterior

Atributo de calidad	Patrón	Descripción	Justificación	Drivers
	Caching	Técnica que almacena datos temporalmente para acelerar el acceso a estos en futuras solicitudes.	Utilizando el caching a través de un CDN para almacenar datos accedidos frecuentemente, como información de chat de IA, se mejora significativamente el tiempo de respuesta. Esta estrategia reduce la necesidad de obtener datos de fuentes más lentas repetidamente, lo que disminuye la carga en los servidores de base de datos y mejora la experiencia del usuario final al proporcionar respuestas más rápidas y reducir la latencia.	Rendimiento, costos
Continúa en la página siguiente				

Tabla D.1 – continuación de la página anterior

Atributo de calidad	Patrón	Descripción	Justificación	Drivers
Interoperabilidad	Service-Oriented Architecture	Patrón que describe una colección de componentes distribuidos que ofrecen o consumen servicios a través de APIs Bass et al. (2021).	Aplicar un estilo de arquitectura basada en servicios facilita la interoperabilidad mediante el uso de APIs REST, permitiendo que distintos componentes de software, potencialmente desarrollados en diferentes plataformas, interactúen entre sí de manera efectiva. Al ofrecer interfaces claras y estandarizadas, promueve la integración y reutilización de servicios, lo que aporta flexibilidad y escalabilidad al sistema informático de la empresa.	Interoperabilidad

Tabla D.1: Patrones

Apéndice E: Lista de componetes de software del Framework

E.0.0.1. Data Structure Component

Componente encargado de organizar la estructura de la información contenida dentro de los documentos cargados. Cambia la forma en que se encuentra la disposición de los textos y demás contenidos para facilitar el proceso de vectorización y que los contenidos sean más legibles para el modelo.

- **Categoría:** Transformación de datos.
- **Entradas:**
 - Texto plano: Palabras, frases o documentos completos.
 - Texto en imágenes: Palabras, frases o documentos completos de tipo imagen cómo logos, tipografías, textos escaneados, entre otros.
 - Texto con formatos variados: Textos que tengan algún tipo de formato cómo títulos, negrilla, viñetas, hipervínculos, o demás formatos que se puedan generar al trabajar por ejemplo con archivos de tipo .DOCX, entre otros.
 - Tablas: Texto organizado en formato de tablas.
- **Salidas:**
 - Datos estructurados: Palabras, frases o documentos completos transformados de manera que el texto cambie su estructura, convirtiéndose en texto plano, fácil de entender y organizado.
 - Texto optimizado para embeddings: Los textos al haber sido transformados para organizar su estructura son más óptimos para garantizar mejores resultados durante el proceso de embedding.
 - Texto legible para LLMs: Los textos al haber sido transformados para organizar su estructura son más fáciles de entender para los LLMs.
- **Casos de uso:**

- Conversión de PDFs escaneados: En el caso de los documentos impresos y que en muchos casos son escaneados y guardados cómo archivos de tipo PDF, muchos modelos no reconocen el texto al tratarse de una imagen y no de texto digitado en si, la implementación de este componente permite convertir las imagenes con texto para que se pueda el contenido escrito de estás.
 - Organización de Tablas en texto plano: A la hora de realizar los embeddings estos pueden tener problemas con el texto organizado en tablas debido a la falta de una mayor estructura semántica, lo cual puede generar después resultados erroneos. Mediante el uso de este componente se puede organizar el contenido de las tablas en un formato de texto plano para que sea más optimo para el proceso de embedding y para los LLMs
 - Documentos extensos: Cuando los documentos son muy extensos no pueden ser almacenados de manera óptima en una base de datos vectorial. Para esto se puede implementar este componente para dividir el contenido del texto en diferentes partes de menor tamaño.
- **Consideraciones:**
 - Complejidad del documento: Dependiendo del contenido del documento o el formato, hay algunos que serán más complejos de transformar que otros.
 - Documentos extensos: En el caso que el documento sea muy extenso y se tenga que partir por partes, hay que tener en cuenta que algunos métodos para realizar este proceso no necesariamente van a cortar el contenido de manera coherente, ya que en algunos casos pueden quedar frases divididas sin terminar en diferentes vectores.
 - Es opcional: No siempre es necesario realizar un proceso de estructuración de los datos, hay muchos documentos cuyo contenido ya se encuentra organizado de manera óptima para los embeddings y el LLM.
 - **Ejemplos:**
 - Unstructured: Herramienta que facilita el proceso de estructuración de los datos para optimizar más los resultados de los modelos de IA [Unstructured \(2024\)](#).

E.0.0.2. Embeddings Component

Encargado de convertir palabras, frases o incluso documentos, en vectores de números reales, estos vectores se pueden utilizar para representar el significado semántico de las palabras o frases, dicho en otras palabras, es el encargado de representar información de forma matemática. Los embeddings comprimen información compleja en vectores de números reales que son más fáciles de procesar por las máquinas.

- **Categoría:** Transformación de datos.
- **Entradas:**

- Texto: Palabras, frases o documentos completos.
- Imágenes: Los píxeles de una imagen.
- Audio: Las señales de audio.
- Cualquier otro tipo de datos: Los embeddings se pueden usar para convertir cualquier tipo de datos en una representación matemática.

■ **Salidas:**

- Vectores de números reales: Es la salida principal de los embeddings, debido a que convierten la información de entrada en vectores de números reales.
- Similitudes: Los embeddings se pueden usar para calcular la similitud entre dos piezas de información.
- Clasificaciones: Los embeddings se pueden usar para clasificar piezas de información en diferentes categorías.
- Recomendaciones: Los embeddings se pueden usar para recomendar piezas de información a los usuarios.

■ **Casos de uso:** Teniendo en cuenta que la función principal de los embeddings es convertir información de alto nivel en una representación matemática de bajo nivel que sea más fácil de procesar por las máquinas, se indican a continuación algunos usos:

- Búsqueda de información: Buscar información en grandes cantidades de texto al encontrar la similitud entre los vectores de la consulta y los vectores de los documentos.
- Clasificación de texto: Clasificación de texto en diferentes categorías al encontrar la similitud entre el vector del texto y los vectores de las categorías.
- Traducción de texto: Traducir texto de un idioma a otro al encontrar la similitud entre el vector del texto en un idioma y los vectores del texto en el otro idioma.
- Motores de búsqueda: Usado en motores de búsqueda para encontrar las páginas web más relevantes para una consulta de búsqueda.
- Redes sociales: En las redes sociales se utilizan para recomendar contenido a los usuarios y para detectar comunidades.

■ **Consideraciones:**

- Variedad: Hay una variedad de algoritmos de embedding disponibles, el algoritmo que se elija depende del tipo de dato y la tarea que se desea realizar.
- Costo: Los embeddings pueden ser costosos de calcular y almacenar.
- Confidencialidad: Los embeddings pueden contener información confidencial, es importante considerar técnicas o mecanismos de seguridad para garantizar la confidencialidad.
- Sesgo: Los embeddings pueden ser sesgados, lo que significa que pueden reflejar los prejuicios de los datos con los que fueron entrenados.

- **Ejemplos:**

- Amazon Titan-embed-text-v1 y OpenAI Embedding: Herramientas que simplifican la conversión de texto a vectores, ampliando las capacidades de los modelos de IA para entender y procesar información.

E.0.0.3. Orchestration Component for LLM and AI Agents

Componente de gestión de flujo de trabajo para automatizar y optimizar procesos, permite crear una conexión entre Agentes de IA, LLM (Large Language Models), bases de datos y modelos de embeddings, para que los agentes de IA tengan la capacidad de interactuar con usuarios o entornos de manera autónoma y realizar tareas específicas.

- **Categoría:** Orquestación.

- **Entradas:**

- Solicitud del Agente de IA: Información que el agente de IA envía al LLM para que este la procese, en este caso particular es de tipo texto. En otras palabras, es la solicitud que hace el usuario al agente de IA.
- Contexto: El contexto proporciona información adicional al LLM que puede ser útil para comprender la solicitud del agente de IA.
- Metadatos: Los metadatos son información adicional sobre la solicitud del agente de IA que puede ser útil para el LLM. Esto puede incluir información sobre el usuario, el dispositivo, la ubicación y la hora.
- LLM: Se conecta a un LLM para enviar y recibir información a manera de prompts y respuestas hacia y desde el usuario.
- Base de datos: Las bases de datos proporcionan un espacio de almacenamiento para las fuentes externas de conocimiento que complementan la representación interna de la información del LLM.
- Embeddings: El modelo de embeddings proporciona la función matemática que permite convertir el texto de entrada y de salida en representaciones numéricas para realizar los cálculos de similitud semántica.

- **Salidas:**

- Respuesta del LLM: Es la salida principal y corresponde a la información que el LLM envía al agente de IA después de procesar la solicitud.
- Código para embeber: El orquestador permite generar un código con un ID para embeber todo el flujo implementado para poder generar una respuesta al usuario a través de una aplicación web.

- **Casos de uso:**

- Implementación de chatbots conversacionales o FAQ bots (Frequently Asked Questions): Chatbots capaces de apoyar al usuario, dando respuestas a preguntas frecuentes o inquietudes en un contexto determinado.
 - Implementación de chatbots de IA: Chatbots capaces de gestionar requerimientos de usuarios, dependiente del contexto.
- **Consideraciones:**
- La complejidad de las entradas y salidas: Esta dependerá de la tarea específica que se esté realizando.
 - Seguridad: Se debe tener en cuenta la seguridad y la privacidad al diseñar la conexión entre el agente de IA y el LLM.
 - Interacción con la IA: La conexión generada entre el agente de IA y el LLM permiten el acceso a múltiples funciones que le permiten interactuar con diferentes componentes relacionados a la configuración, el almacenamiento y el procesamiento de los prompts digitados por el usuario.
 - Flujo de trabajo: La integración completa del orquestador genera un flujo de trabajo que permite visualizar y entender la manera en que se conectan todos los componentes dentro de este, cómo interactúan entre sí y cómo se fluye la información desde el momento en que se carga hasta el momento en que se genera una respuesta.
- **Ejemplos:**
- Flowise AI: Permite elaborar flujos de orquestación a medida y agentes de IA [Heng \(2023\)](#).

E.0.0.4. Knowledge Database Component

Encargado de la persistencia de los datos, además de permitir un procesamiento rápido y eficiente de grandes cantidades de datos textuales, debido a que almacena y organiza la información en forma de vectores. Estos vectores son representaciones matemáticas multidimensionales que codifican información sobre entidades o conceptos. Este componente actúa como una fuente externa de conocimiento para complementar la representación interna de la información del LLM, garantizando que el modelo tenga acceso a los hechos más actuales y confiables.

- **Categoría:** Almacenamiento.
- **Entradas:**
- Vector: La entrada principal a una base de datos vectorial en el contexto de los chatbot es el vector que representa la consulta del usuario, puede ser una oración, un párrafo o cualquier otra forma de texto que sea ingresado por el usuario.

- Metadatos (Opcional): Metadatos sobre la consulta del usuario, estos metadatos pueden incluir información cómo la hora y la fecha de la consulta, el idioma de la consulta y el canal a través del cual se realizó la consulta.

▪ **Salidas:**

- Vectores: Es la salida principal de una base de datos vectorial, es un conjunto de vectores que representan el significado de la consulta del usuario, si en este vector se encuentra la respuesta al usuario, el chatbot generará la respuesta a partir de esta información, si la respuesta no se encuentra en esta salida, el chatbot podrá generar una respuesta de seguimiento para el usuario (solicitar información adicional) o puede con base en esa vector de salida volver a preguntar a la base de datos para obtener una respuesta y entregarla al usuario.
- Similitudes: La base de datos vectorial puede devolver una lista de similitudes que son relevantes para la consulta del usuario, estos pueden incluir documentos, entidades o cualquier otro tipo de información que sea relevante para la consulta.
- Metadatos: La base de datos vectorial también puede devolver metadatos sobre los resultados, los metadatos pueden incluir información cómo la puntuación de relevancia del resultado, la fuente del resultado y la fecha de creación del resultado, entre otros.

▪ **Casos de uso:**

- Persistencia de información: Persistencia de entidades, conceptos y relaciones que son relevantes para el dominio del chatbot.
- Representar el significado de las palabras y frases: Lo que permite que el chatbot comprenda las consultas de los usuarios y genere respuestas precisas y relevantes.

▪ **Consideraciones:**

- Mayor flexibilidad: Se pueden utilizar para una amplia gama de tareas, cómo la búsqueda de información, la clasificación de texto y la traducción automática.
- Mayor eficiencia: Pueden realizar búsquedas de similitud mucho más rápido que las técnicas tradicionales de PLN.
- Mayor precisión: Pueden comprender mejor el significado del texto y proporcionar respuestas más precisas.
- Similaridad Semántica: Facilitan el cálculo de la similitud semántica entre vectores, lo que permite encontrar elementos relacionados o similares a una consulta específica.
- Integración con Modelos de Aprendizaje Automático: Pueden integrarse fácilmente con modelos de aprendizaje automático para mejorar la precisión y la relevancia de las respuestas de los chatbots.
- Pueden ser costosas de mantener.

- Pueden ser sesgadas, lo que puede conducir a respuestas inexactas.

- **Ejemplos:**

- Qdrant: Motor de bases de datos vectorial y búsqueda de similitud de vectores [Vasnetsov \(2021\)](#).

E.0.0.5. Buffer Memory Component

Componente que permite al agente de IA y al LLM recordar conversaciones que ha tenido con los usuarios, ofreciendo así un servicio más personalizado y mejorando la experiencia del usuario.

- **Categoría:** Almacenamiento.

- **Entradas:**

- Texto de la conversación actual entre el LLM/Agente de IA y los clientes: El componente recibe conversaciones que tengan lugar en la sesión actual entre el LLM/Agente de IA y los clientes. Estas conversaciones pueden incluir texto de entrada del cliente, respuestas generadas por el LLM/Agente de IA.
- Metadatos: Metadatos sobre los datos, cómo el nombre de los datos, el tipo de datos y la fecha de creación, entre otros.

- **Salidas:**

- Texto de las conversaciones anteriores entre el LLM/Agente de IA y los clientes: Es la salida principal del componente teniendo en cuenta que su principal función es la capacidad de almacenar y recuperar estas conversaciones.
- Metadatos: Los metadatos sobre los datos que se almacenaron en el búfer.

- **Casos de uso:**

- Atención al cliente o usuario: Permite a los chatbot del servicio de atención al cliente recordar las conversaciones anteriores con los clientes y ofrecer un servicio más personalizado.
- Personalización de Respuestas: El componente puede utilizar las conversaciones anteriores para personalizar las respuestas futuras del LLM/Agente de IA, proporcionando respuestas más relevantes y coherentes a los clientes.
- Análisis de Interacciones: El componente puede analizar las conversaciones almacenadas para identificar patrones, tendencias y áreas de mejora en las interacciones entre el LLM/Agente de IA y los clientes.
- Seguimiento de Historial: El componente puede proporcionar a los usuarios la capacidad de revisar y recuperar conversaciones anteriores, lo que facilita el seguimiento del historial de interacciones y la resolución de problemas.

- **Consideraciones:**

- Seguridad: El componente debe ser seguro para proteger los datos que se almacenan en él de accesos no autorizados.
- Privacidad: El componente debe respetar la privacidad de los usuarios y proteger sus datos personales.
- Rendimiento: El componente debe ser eficiente para poder manejar grandes cantidades de datos.
- Optimización del Almacenamiento: Es importante optimizar el almacenamiento de las conversaciones para minimizar el uso de recursos y garantizar un rendimiento óptimo del sistema.

- **Ejemplos:**

- Redis y DynamoDB: Bases de datos NoSQL empleadas para la memoria caché, mejorando el acceso rápido a datos frecuentemente requeridos [Agra \(2020\)](#); [Amazon \(2012\)](#).

E.0.0.6. LLMs Configuration and Execution Component

Permite configurar y ejecutar los LLMs (Large Language Models) de manera local, haciendo uso de la CPU o GPU de la máquina donde se está ejecutando.

- **Categoría:** Configuración y ejecución.

- **Entradas:**

- LLM: El Large Language Model que se quiere ejecutar.
- Configuración: Configuración del LLM mediante parámetros o mediante un archivo de configuración.
- Base de datos: Las bases de datos proporcionan un espacio de almacenamiento para las fuentes externas de conocimiento que complementan la representación interna de la información del LLM.
- Confianza (Opcional): La confianza es una medida de la certeza del LLM en la precisión de su respuesta. Esta medida de confianza ayuda al agente de IA a evaluar la calidad y la fiabilidad de la respuesta generada por el LLM.

- **Salidas:**

- Texto: Es la salida principal y representa una respuesta a una pregunta, una traducción de un idioma a otro, un resumen de un texto o una creación de texto creativa, entre otros.
- Precisión: El LLM puede incluir marcadores de precisión en la respuesta para indicar áreas donde la confianza en la respuesta es baja. Estos marcadores pueden tomar la forma de palabras como "quizás", "posiblemente." o "no estoy seguro".

- Metadatos: El componente puede proporcionar metadatos sobre el texto generado, cómo por ejemplo el tema, el autor, la fecha de creación y otras variables relevantes.
 - Explicación (Opcional): La explicación es una descripción de cómo el LLM llegó a su respuesta. Esto puede ser útil para el agente de IA para comprender la respuesta del LLM y para decidir cómo usarla.
- **Casos de uso:**
 - Configuración y ejecución de LLMs: Configura y ejecuta los LLM de manera local, bien sea en un equipo físico o una máquina virtual, ya que proporciona la infraestructura y las herramientas necesarias para usar LLMs en proyectos propios.
- **Consideraciones:**
 - Rendimiento: El rendimiento del componente depende del hardware que se esté utilizando. Una GPU o CPU más potente dará un mejor desempeño.
 - Alto consumo de memoria RAM para la ejecución del LLM.
 - Almacenamiento: Se debe contar con una buena capacidad de almacenamiento, debido a que los LLM por lo regular ocupan bastante espacio en disco.
 - Compatibilidad: Durante la elección del componente se debe revisar que sea compatible con el LLM que se va a usar.
- **Ejemplos:**
 - Ollama: Facilita el despliegue eficiente de modelos de lenguaje extensos en servidores locales o servicios en la nube [Morgan \(2023\)](#).

E.0.0.7. Large Language Model Component

Componente que está previamente entrenado con grandes cantidades de datos para aprender y reproducir la estructura del lenguaje, puede realizar diversas tareas de procesamiento del lenguaje natural (NLP) cómo por ejemplo, generar y clasificar texto, responder a preguntas de forma conversacional y traducir texto de un idioma a otro.

- **Categoría:** Tareas de procesamiento del lenguaje natural (NLP).
- **Entradas:**
 - Fuente de información: Texto usado para el entrenamiento de este componente, pueden ser libros, artículos, sitios web, scripts, fragmentos de código entre otros. El texto puede ser en bruto o preprocesado, cómo la eliminación de stopwords o la lematización.
 - Texto digitado por el usuario: Prompts que pueden ser una pregunta, una solicitud, una afirmación u otra forma de entrada por parte del usuario.

- Imágenes (opcional): Usadas para el entrenamiento de este componente. Las imágenes pueden ser en bruto o preprocesadas, cómo la redimensión o la normalización.
 - Audio (opcional): Usado para el entrenamiento de este componente. El audio puede ser en bruto o preprocesado, cómo la extracción de características o la reducción de ruido.
 - Video (opcional): Usado para el entrenamiento de este componente. El video puede ser en bruto o preprocesado, cómo la extracción de fotogramas o la reducción de la tasa de bits.
 - Contexto Adicional: Usado para suministrar información contextual adicional para comprender mejor el contexto de la solicitud. Esto puede incluir metadatos cómo la fecha y hora, el idioma, la ubicación del usuario, el historial de interacciones previas, entre otros.
- **Salidas:**
- Texto: Es la salida principal de este componente, este texto puede ser la respuesta a preguntas realizadas a él, poemas, historias, guiones, código, traducciones, clasificación de texto o imágenes, entre otros. El texto puede variar dependiendo de la complejidad de la solicitud y la capacidad del modelo.
 - Imágenes: En el caso que el modelo esté entrenado y cuente con los complementos necesarios, este puede generar imágenes con base al prompt digitado.
 - Audio: En el caso que el modelo esté entrenado y cuente con los complementos necesarios, este puede generar audio con base al prompt digitado.
 - Vídeo: En el caso que el modelo esté entrenado y cuente con los complementos necesarios, este puede generar video con base al prompt digitado.
 - Metadatos de Salida: Este componente puede proporcionar metadatos adicionales asociados con la salida, cómo la longitud del texto, la puntuación de la respuesta, la relevancia, entre otros.
 - Embeddings: Por medio de un modelo de embedding se transforma el contenido ingresado por el usuario (que puede ser texto, imágenes, audio o video) para ser comparado por medio de cálculos de similitud semántica con los vectores almacenados en la base de datos vectorial. Los embeddings son vectores de números reales que representan la información semántica del contenido.
- **Casos de uso:**
- Generación de contenido: Este componente está en capacidad de generar contenido creativo cómo por ejemplo, poemas, historias, guiones, entre otros.
 - Análisis de datos: Este componente se puede usar para clasificación de textos en categorías, extracción de entidades de un texto, cómo por ejemplo, personas, lugares, organizaciones, entre otras.

- **Búsqueda de información:** Este componente puede ser usado para responder preguntas sobre un tema determinado, búsqueda de semántica de información, entre otros.
- **Interacción con el usuario:** Este componente puede usarse para crear chatbots que puedan interactuar con usuarios humanos, así cómo también la creación de asistentes virtuales que puedan ayudar a los usuarios con tareas cómo programar citas, hacer reservas, encontrar información, entre otros.

■ **Consideraciones:**

- **Prompt:** Es necesario saber cómo redactar un prompt y tener en cuenta todos los criterios que estos requieren. A este proceso se le conoce cómo ingeniería de prompt
- **Tamaño:** El tamaño de las entradas y salidas puede afectar el rendimiento de este componente.
- **Calidad:** La calidad de las entradas puede afectar la calidad de las salidas de este componente.
- **Seguridad:** La seguridad de este componente puede verse afectada por el contenido de las entradas.
- **Privacidad:** Es importante tener en cuenta la privacidad al usar este componente, especialmente para aplicaciones que procesan datos personales.

■ **Ejemplos:**

- **GPT-4 de OpenAI:** Modelo de lenguaje de gran tamaño (LLM) conocido por su habilidad para generar textos coherentes y contextualizados [Jeong \(2024\)](#).
- **Llama2 de Meta:** Dotado de capacidades similares para tareas de NLP, sobresale por su eficacia en el procesamiento y generación de lenguaje [Jeong \(2024\)](#).

E.0.0.8. Documents Storage Component

Este componente está diseñado para el almacenamiento de archivos no estructurados como documentos en formatos Excel, PDF, Word, entre otros. Posteriormente, estos documentos son vectorizados para su almacenamiento en una base de datos vectorial, permitiendo así que la Inteligencia Artificial acceda fácilmente a la información. Aunque este componente es opcional, ya que su funcionalidad podría ser integrada en el componente orquestador, incluyendo los documentos directamente en él, ofrece la posibilidad de separar la responsabilidad del almacenamiento de otros componentes de software sin impactar su rendimiento.

■ **Categoría:** Almacenamiento.

■ **Entradas:**

- Documentos en diversos formatos como .docx, .pdf, .csv, .txt, .json, imágenes, hojas de cálculo y otros tipos de archivos digitales.

■ Salidas:

- Documentos digitalizados almacenados de manera segura en espacios de almacenamiento tanto en la nube como en infraestructura local (On-Premise), con opciones de redundancia para garantizar su disponibilidad y durabilidad.
- Indexación de los documentos para facilitar búsquedas rápidas y eficientes dentro del espacio de almacenamiento.
- Metadatos asociados a cada documento almacenado, que pueden incluir información como el autor, fecha de creación, última modificación y cualquier dato relevante para su clasificación y recuperación.
- Sistemas de versionado de documentos que permiten llevar un seguimiento de los cambios realizados y posibilitan la restauración a versiones anteriores si es necesario.
- Integración con servicios de backup y disaster recovery para proteger los documentos contra pérdida de datos.
- Posibilidad de establecer políticas de retención de documentos y de automatización de la eliminación de archivos obsoletos o que ya no se necesiten, cumpliendo con las normativas de gestión de información.
- Interfaces de programación de aplicaciones (APIs) para la interacción con otros componentes o sistemas, que permiten la automatización de procesos como la carga y descarga de documentos, así como su vectorización.

■ Casos de uso:

- Gestión de Documentación Regulatoria: Almacenar documentación legal y regulatoria que requiere una recuperación rápida y auditabilidad completa para cumplir con normativas específicas del sector.
- Gestión de Contenido Empresarial: Como parte de un sistema de gestión de contenido empresarial (ECM), almacenar contratos, informes financieros y correspondencia empresarial de manera segura y estructurada.
- Plataformas de Colaboración en Línea: Permitir que los usuarios suban y compartan documentos en un entorno colaborativo, manteniendo el versionado y el control de acceso de los documentos.
- Backup y Recuperación de Desastres: Utilizar el almacenamiento para realizar copias de seguridad de documentos importantes y garantizar su disponibilidad en caso de incidentes que comprometan los datos locales.
- Integración con Servicios de Análisis de Datos: Facilitar el procesamiento y análisis de grandes conjuntos de documentos no estructurados a través de servicios de inteligencia de negocios o análisis de datos.

■ Consideraciones:

- **Estructura Organizativa:** La solución de almacenamiento debe permitir la creación y manejo de estructuras organizativas como carpetas, etiquetas o esquemas de clasificación que faciliten el acceso, la gestión y la recuperación de los datos.
 - **Escalabilidad:** El sistema de almacenamiento permite escalar, tanto vertical como horizontalmente, para adaptarse al crecimiento en volumen de los datos sin comprometer el rendimiento.
 - **Recuperación y Respaldo de Datos:** Deben existir políticas y mecanismos de recuperación ante desastres y respaldo de datos para asegurar la continuidad del negocio y la integridad de los datos en caso de fallas del sistema.
 - **Interoperabilidad:** Capacidad de integrarse con otros sistemas y tecnologías mediante APIs o interfaces de programación para automatizar flujos de trabajo y facilitar las operaciones de almacenamiento y recuperación.
- **Ejemplos:**
- **AWS S3:** ofrece una solución de almacenamiento escalable y segura. Permite almacenar grandes cantidades de información en un entorno distribuido, accesible desde cualquier lugar. AWS S3 proporciona opciones de organización a través de 'buckets' y dentro de ellos, permite estructurar los datos con un sistema de carpetas. Ofrece durabilidad, alta disponibilidad y servicios integrados para el manejo de ciclo de vida de los datos, cifrado y seguridad. Esta integración posibilita una eficiente vectorización y recuperación de documentos para su uso en aplicaciones de Inteligencia Artificial, garantizando así que los documentos estén disponibles para el componente orquestador o cualquier otro sistema que requiera acceso a ellos.

