



UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ

Facultad:

CIENCIAS DE LA VIDA Y TECNOLOGÍAS

Carrera:

TECNOLOGÍAS DE LA INFORMACIÓN

Tema:

"ACTUALIZACIÓN Y MEJORAMIENTO DEL SISTEMA DE SEGUIMIENTO A GRADUADOS DE LA ULEAM, MATRIZ MANTA."

Trabajo de titulación presentado en conformidad con los requisitos establecidos para optar por el título de Ingeniero en tecnologías de la información.

Autores

Calderón López Byron Gregorio

Zavala Reyes Fabricio Josué

Tutor

Ing. González López Oscar Armando, Mg.

Manta – Manabí – Ecuador

Período Académico 2025(2)

Declaración expresa de autoría

Yo, BYRON GREGORIO CALDERÓN LÓPEZ, declaramos bajo juramento que el trabajo de titulación presentado es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración, cedemos los derechos de propiedad intelectual correspondiente a este trabajo, a la universidad Laica Eloy Alfaro de Manabí, según lo establecido por la ley de propiedad intelectual, por su reglamento y por la normativa institucional vigente.

Manta, 04 de febrero de 2026



Byron Gregorio Calderón López

Declaración expresa de autoría

Yo, FABRICIO JOSUÉ ZAVALA REYES, declaramos bajo juramento que el trabajo de titulación presentado es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración, cedemos los derechos de propiedad intelectual correspondiente a este trabajo, a la universidad Laica Eloy Alfaro de Manabí, según lo establecido por la ley de propiedad intelectual, por su reglamento y por la normativa institucional vigente.

Manta, 04 de febrero de 2026



Fabricio Josué Zavala Reyes|

CERTIFICACIÓN

En calidad de docente tutor(a) de la Facultad Ciencias de la vida y tecnología matriz manta de la carrera de Tecnologías de la información de la Universidad Laica "Eloy Alfaro" de Manabí, CERTIFICO:

Haber dirigido, revisado y aprobado preliminarmente el Trabajo de Integración Curricular bajo la autoría de los estudiantes Calderón López Byron Gregorio legalmente matriculado/a en la carrera de tecnología de la información período académico 2025-2, cumpliendo el total de 400 horas, cuyo tema del proyecto es **"Actualización y mejoramiento del sistema de seguimiento a graduados de la Uleam, Matriz Manta"**.

La presente investigación ha sido desarrollada en apego al cumplimiento de los requisitos académicos exigidos por el Reglamento de Régimen Académico y en concordancia con los lineamientos internos de la opción de titulación en mención, reuniendo y cumpliendo con los méritos académicos, científicos y formales, y la originalidad del mismo, requisitos suficientes para ser sometida a la evaluación del tribunal de titulación que designe la autoridad competente.

Particular que certifico para los fines consiguientes, salvo disposición de Ley en contrario.

Manta, 05 de 02 del 2026.

Lo certifico,


Ing. González López Oscar Armando
Docente Tutor(a)
Área:

	NOMBRE DEL DOCUMENTO:	CÓDIGO: PAT-04-F-004
	CERTIFICADO DE TUTOR(A).	
	PROCEDIMIENTO: TITULACIÓN DE ESTUDIANTES DE GRADO BAJO LA UNIDAD DE INTEGRACIÓN CURRICULAR	REVISIÓN: 1 Página 1 de 1

CERTIFICACIÓN

En calidad de docente tutor(a) de la Facultad Ciencias de la vida y tecnología matriz manta de la carrera de Tecnologías de la información de la Universidad Laica "Eloy Alfaro" de Manabí, CERTIFICO:

Haber dirigido, revisado y aprobado preliminarmente el Trabajo de Integración Curricular bajo la autoría de los estudiantes Zavala Reyes Fabricio Josué legalmente matriculado/a en la carrera de tecnología de la información período académico 2025-2, cumpliendo el total de 400 horas, cuyo tema del proyecto es **"Actualización y mejoramiento del sistema de seguimiento a graduados de la Uleam, Matriz Manta"**.

La presente investigación ha sido desarrollada en apego al cumplimiento de los requisitos académicos exigidos por el Reglamento de Régimen Académico y en concordancia con los lineamientos internos de la opción de titulación en mención, reuniendo y cumpliendo con los méritos académicos, científicos y formales, y la originalidad del mismo, requisitos suficientes para ser sometida a la evaluación del tribunal de titulación que designe la autoridad competente.

Particular que certifico para los fines consiguientes, salvo disposición de Ley en contrario.

Manta, 05 de 02 del 2026.

Lo certifico,


 Ing. González López Oscar Armando
Docente Tutor(a)
Área:

Agradecimiento

En primer lugar, agradezco a mi familia por el apoyo incondicional y los sacrificios que hicieron para que yo pudiera concentrarme en mis estudios.

Un agradecimiento especial a mi laptop HP, mi compañera incansable de batallas. Agradezco su fidelidad en cada noche de desvelo, su resistencia ante las jornadas interminables de investigación y por ser la herramienta tecnológica que nunca me falló en los momentos más críticos de la redacción de esta tesis.

Para cerrar, quiero expresarme un profundo agradecimiento a mí mismo. Me agradezco por la resiliencia ante la frustración, por las horas de sueño sacrificadas y por la fuerza de voluntad para terminar lo que comencé. Este título es el fruto de mi dedicación personal, de mi esfuerzo constante y del compromiso inquebrantable con mi futuro.

Byron Gregorio Calderón López

Agradecimiento

La primera persona a quien quiero agradecer enormemente es a mi querida madre Irene, muchas gracias mamá por apoyarme siempre, por estar junto a mí incluso cuando no me daba cuenta. Mi madre es la persona que hizo todo esto posible y siempre recordaré con amor todo el sacrificio que hizo por mí.

A mi abuela Olalla por todo su cariño y bondad hacia a mí, ellas dos me ayudaron a forjar el carácter y a ser resiliente.

A José Israel por toda la ayuda que me brindó incondicionalmente.

A mi padre Lider que estuvo conmigo en momentos complicados.

A los amigos que hice a lo largo de mi carrera universitaria, cada uno de ustedes me enseñó a que no estamos solos en este proceso.

A mis docentes que me brindaron conocimientos y una mano amiga, hicieron mas de lo que debían y son excelentes personas.

A mis compañeros de trabajo por enseñarme y tenerme paciencia entendiendo que todos empezamos desde abajo.

Gracias a todos porque alguna vez, a su manera, fueron de mucha ayuda, porque al final somos el reflejo de las personas que nos rodean.

Fabricio Josué Zavala Reyes

Dedicatoria

Dedico este pequeño logro principalmente a la persona que fui al iniciar esta carrera y a la que soy hoy, por no rendirse nunca. Me dedico este título a mí mismo, como un recordatorio de que soy capaz de alcanzar lo que me proponga, reconociendo mi propia disciplina, mi paciencia y la madurez adquirida durante este proceso de formación profesional.

Finalmente, dedico este trabajo a mis padres por darme la vida y la oportunidad de vivir este momento, y a todos aquellos que, de palabra o acción, me dieron el impulso necesario para creer que llegar a la meta era posible.

Byron Gregorio Calderón López

Dedicatoria

Dedico este trabajo a mi madre y a mi familia, por el apoyo constante, la paciencia y el sacrificio que nunca faltaron a lo largo de este proceso.

Estuvieron presentes en los momentos más difíciles, acompañándome sin condiciones y confiando en mí incluso cuando las cosas no eran claras.

Este logro es también de ustedes, porque su presencia y su apoyo hicieron posible llegar hasta aquí.

Fabricio Josué Zavala Reyes

Índice de contenidos

1	Capítulo I	15
1.1	RESUMEN	15
1.2	ABSTRACT	16
1.3	Introducción	17
1.4	Planteamiento del problema.....	19
1.4.1	Análisis Macro (Contexto Internacional).....	19
1.4.2	Análisis Meso (Contexto Nacional y Regional).....	19
1.4.3	Análisis Micro (Contexto Institucional - ULEAM)	20
1.5	Formulación del problema	22
1.6	Objetivos	23
1.6.1	Objetivo general.....	23
1.6.2	Objetivos específicos.....	23
1.7	Justificación	23
1.7.1	Justificación académica.....	23
1.7.2	Justificación práctica	24
1.7.3	Justificación social	24
1.7.4	Justificación Técnica	24
1.8	Impactos esperados	25
2	CAPÍTULO II: MARCO TEÓRICO DE LA INVESTIGACIÓN.....	26
2.1	Antecedentes de la investigación	26
2.2	Fundamentación Teórica del Seguimiento a Graduados.....	31
2.3	Arquitectura de Software y Sistemas Distribuidos.....	34
2.4	Tecnologías de Desarrollo Frontend y Backend	36
2.5	Inteligencia Artificial y Ciencia de Datos	39
2.6	Seguridad Web y Protección de Datos	44
2.7	Experiencia de Usuario y Visualización de Datos	47
2.8	Metodologías de Desarrollo y Calidad de Software	48
2.9	Modelos de Computación en la Nube (Cloud Computing)	49

2.10	Ética en la Inteligencia Artificial: Sesgos y Transparencia.....	50
2.11	Soberanía Tecnológica e Independencia Digital	51
3	Capítulo III	53
3.1	Tipo de investigación	53
3.2	Métodos de investigación	53
3.2.1	Métodos Teóricos.....	53
3.2.2	Métodos Empíricos	55
3.3	Estrategia operacional para la recolección de datos.....	55
3.3.1	Población - Segmentación - Técnica de muestreo - Tamaño de la muestra	55
3.3.2	Análisis de las herramientas de recolección de datos a utilizar	56
3.3.3	Plan de recolección de datos	57
3.4	Análisis y presentación de resultados	59
3.4.1	Tabulación y análisis de los datos	59
4	Capítulo IV	60
4.1	Análisis de Requerimientos	60
4.1.1	Requerimientos Funcionales.....	60
4.1.2	Requerimientos No Funcionales	63
4.2	Arquitectura del Sistema	64
4.2.1	Arquitectura de Software.....	64
4.2.2	Tecnologías Utilizadas	67
4.3	Diseño de la Base de Datos	68
4.4	Descripción de Procesos Principales	72
4.4.1	Proceso de Postulación a Oferta	73
4.4.2	Proceso de Publicación de Oferta	73
4.4.3	Activación del Chat.....	73
4.4.4	Proceso de Configuración de Encuesta y Diseño de IA.....	74
4.4.5	Proceso de Participación de Graduados	74
4.4.6	Proceso de Generación de Reportes con IA.....	74
4.5	Interfaces y Capturas del Sistema	75

4.5.1	Interfaces del Módulo de Bolsa de Empleo	75
4.5.2	Interfaces del Módulo de Seguimiento a Graduados	78
5	Capítulo V.....	81
5.1	Conclusiones.....	81
5.2	Recomendaciones	83
5.3	REFERENCIAS BIBLIOGRÁFICAS	84
6	Anexos.....	86
6.1	Enlace al sistema de seguimiento a graduados:	86
6.2	Enlace al sistema de bolsa de empleos:	86

Índice de tablas

Tabla 1	61
----------------------	-----------

de Índice figuras

Figura 1	65
Figura 2	66
Figura 3	67
Figura 4	69
Figura 5	71
Figura 6	75
Figura 7	76
Figura 8	76
Figura 9	77
Figura 10	77
Figura 11	78
Figura 12	78
Figura 13	79
Figura 14	79
Figura 15	79

1 Capítulo I

1.1 RESUMEN

La gestión del seguimiento a graduados en la Universidad Laica Eloy Alfaro de Manabí (ULEAM) presentaba deficiencias operativas debido a una arquitectura de software obsoleta y monolítica basada en Laravel, la cual dificultaba la escalabilidad y obligaba al procesamiento manual de datos cualitativos.

El objetivo de este proyecto fue actualizar y mejorar dicho sistema mediante una reingeniería total, implementando una arquitectura modular distribuida. La metodología aplicada se basó en el desarrollo de múltiples backends, utilizando Angular 19 para el frontend, Node.js para soportar la alta concurrencia de la bolsa de empleo, y Python integrado con el modelo local de inteligencia artificial Llama 3.2 para la automatización de reportes. Como resultado, se logró aislar la concurrencia, reduciendo significativamente los tiempos de carga y automatizando la tabulación de respuestas abiertas mediante IA bajo un entorno de Servidor Privado Virtual (VPS). Se concluye que la transición de una arquitectura rígida a un ecosistema elástico mejora radicalmente la eficiencia administrativa, garantizando la escalabilidad del sistema y la privacidad absoluta de los datos institucionales.

1.2 ABSTRACT

The management of graduate tracking at the Universidad Laica Eloy Alfaro de Manabí (ULEAM) presented operational deficiencies due to an obsolete and monolithic software architecture based on Laravel, which hindered scalability and forced the manual processing of qualitative data.

The objective of this project was to update and improve this system through a complete reengineering, implementing a distributed modular architecture. The methodology applied was based on the development of multiple backends, using Angular 19 for the frontend, Node.js to support the high concurrency of the job board, and Python integrated with the local artificial intelligence model Llama 3.2 for the automation of reports. As a result, concurrency was successfully isolated, significantly reducing load times and automating the tabulation of open-ended responses using AI under a Virtual Private Server (VPS) environment. It is concluded that the transition from a rigid architecture to an elastic ecosystem radically improves administrative efficiency, ensuring system scalability and the absolute privacy of institutional data

1.3 Introducción

Para la Universidad Laica Eloy Alfaro de Manabí, el seguimiento a graduados dejó de ser un simple trámite administrativo; se convirtió en una barrera técnica. La calidad académica ya no se mide solo en el aula, sino en el éxito del profesional fuera de ella. Sin embargo, en la ULEAM, esta métrica crítica chocaba con una realidad operativa insostenible. No fallaban los datos, fallaba el software. La plataforma anterior, un monolito construido sobre versiones heredadas de Laravel, presentaba una arquitectura rígidamente acoplada. La base de datos, la lógica de negocio y la interfaz de usuario poseían un nivel de dependencia tan alto que cualquier intento de actualización o modernización en un componente comprometía la estabilidad operativa del resto del sistema.

Como respuesta a este escenario, el presente proyecto de titulación instrumenta una **reingeniería integral del ecosistema digital**. Se propone el abandono definitivo del esquema monolítico para transitar hacia un diseño modular y distribuido. Bajo esta premisa de alta disponibilidad, se orquestó una infraestructura técnica soportada por tres *backends* independientes, garantizando no solo la eficiencia en el procesamiento, sino la resiliencia del sistema ante demandas crecientes de carga. Los tres backends desarrollados son:

- Un servicio en Node.js dedicado exclusivamente a la lógica del Seguimiento a Graduados.
- Un segundo servicio en Node.js optimizado para manejar la concurrencia de la Bolsa de Empleo.
- Un tercer backend desarrollado en Python, especializado en el procesamiento de datos y la ejecución del modelo de lenguaje.

Esta lógica de negocio se consume a través de dos aplicaciones Frontend independientes desarrolladas en Angular 19, lo que permite desacoplar la experiencia de usuario: una interfaz está diseñada específicamente para las métricas y gestión del comité de seguimiento, mientras que la otra se enfoca en la usabilidad requerida para la bolsa de empleos.

Esta capacidad operativa se potencia, de manera transversal, con la integración de una capa de Inteligencia Artificial Local. Al ser ejecutada sobre una infraestructura propia de Servidores Privados Virtuales (VPS), esta unidad de procesamiento garantiza la soberanía de los datos y la eficiencia analítica, permitiendo que la interpretación de grandes volúmenes de información profesional se realice bajo un entorno controlado, escalable y libre de las latencias asociadas a servicios externos.

Esta arquitectura permite correr el modelo Llama 3.2 bajo control total del proyecto, facultando al sistema para interpretar respuestas abiertas y redactar informes técnicos en formato .docx de manera autónoma. De esta forma, se garantiza la soberanía de los datos y se transiciona de un flujo operativo manual de semanas a una gestión automatizada de minutos.

1.4 Planteamiento del problema

El análisis de la problemática se aborda desde tres dimensiones contextuales: macro (internacional), meso (nacional/regional) y micro (institucional), permitiendo visualizar la necesidad de actualización tecnológica como una respuesta a tendencias globales y necesidades locales.

1.4.1 *Análisis Macro (Contexto Internacional)*

La postura de organismos como la UNESCO y la Asociación Internacional de Universidades es clara: la calidad académica hoy se valida por el impacto del egresado en el sector productivo, no por los contenidos teóricos. No obstante, se percibe una disonancia crítica a escala global. En un escenario donde el sector industrial demanda niveles de analítica de datos sin precedentes, el entorno académico permanece, en gran medida, anclado a la gestión de los denominados **sistemas legado** (*legacy systems*). Estas estructuras no son más que repositorios estáticos de registros históricos; arquitecturas que han quedado obsoletas ante la necesidad de procesar la trayectoria laboral del estudiantado bajo una óptica de tiempo real.

Ante este panorama, la tendencia internacional se vuelve un mandato tecnológico: la transición necesaria de aplicaciones monolíticas —caracterizadas por su falta de flexibilidad— hacia **ecosistemas modulares de alta integración**. La apuesta actual no es solo el cambio de código, sino la incorporación estratégica de Modelos de Lenguaje de Gran Escala (LLMs) que permitan decodificar, desde una dimensión cualitativa, las dinámicas reales que enfrentan los profesionales en el tejido productivo actual.

1.4.2 *Análisis Meso (Contexto Nacional y Regional)*

En el Ecuador, el Consejo de Aseguramiento de la Calidad de la Educación Superior (CACES) establece en su modelo de evaluación indicadores estrictos sobre el seguimiento a

graduados. Las universidades ecuatorianas están obligadas a reportar tasas de empleabilidad, afinidad laboral y niveles de satisfacción. No obstante, en la región de Manabí, muchas instituciones aún gestionan estos procesos de manera semimanual o mediante sistemas web desarrollados con tecnologías que, si bien fueron estándares hace una década (como versiones antiguas de PHP/Laravel acopladas al frontend), hoy presentan limitaciones para escalar.

La fractura entre el sector productivo y la academia no es solo una desconexión teórica, sino una crisis de operatividad regional que se agudiza por la obsolescencia de las herramientas actuales. A día de hoy, los sistemas vigentes operan como meros depósitos de información estática; son estructuras pasivas que carecen de una lógica proactiva para la gestión de empleabilidad o el análisis avanzado de perfiles profesionales.

1.4.3 Análisis Micro (Contexto Institucional - ULEAM)

En la Universidad Laica Eloy Alfaro de Manabí (ULEAM), Matriz Manta, el sistema de seguimiento a graduados operaba bajo una arquitectura monolítica desarrollada en el framework Laravel. Aunque funcional en sus inicios, este sistema evidenció una deuda técnica significativa frente a los nuevos requerimientos institucionales y tecnológicos.

Se detectaron las siguientes falencias críticas en el sistema anterior:

1. **El lastre del procesamiento manual de datos:** Actualmente, el sistema se limita a ser un recolector de encuestas, pero deja el análisis de las respuestas abiertas en un limbo operativo. En la práctica, esto fuerza al personal administrativo a realizar una revisión manual exhaustiva, exportando registros a hojas de cálculo para una tabulación tradicional. Hablamos de una pérdida de

cientos de horas-hombre que, además de ser ineficiente, dispara la probabilidad de errores en la interpretación de los datos cualitativos.

2. **Rigidez arquitectónica y dependencia de un núcleo monolítico:** La plataforma arrastra un fuerte acoplamiento bajo Laravel que ha terminado por asfixiar cualquier intento de modernización. Al no estar separadas las capas de presentación y lógica, integrar librerías de Ciencia de Datos (que son, por naturaleza, el terreno de Python) o actualizar la interfaz de usuario representa un desafío técnico significativo. El costo y la complejidad de refactorizar un sistema tan rígido limitan drásticamente, cualquier posibilidad de escalabilidad real.
3. **Generación de reportes: un proceso rudimentario:** La producción de informes para los procesos de acreditación es, a día de hoy, un proceso ejecutado de forma completamente manual.

1.5 Formulación del problema

¿De qué manera el desarrollo de un nuevo sistema modular con inteligencia artificial optimizará la gestión de encuestas, la automatización de reportes y la inserción laboral de los egresados en la Universidad Laica Eloy Alfaro de Manabí, Matriz Manta?

1.6 Objetivos

1.6.1 *Objetivo general*

Actualizar y mejorar el sistema de seguimiento a graduados de la ULEAM, Matriz Manta, implementando una arquitectura modular que integre un backend administrativo en Node.js, un núcleo de inteligencia artificial en Python con el modelo Llama 3.2 y una interfaz en Angular, para automatizar la generación de reportes técnicos y dinamizar la inserción laboral.

1.6.2 *Objetivos específicos*

- Analizar los requerimientos funcionales y no funcionales del proceso de seguimiento a graduados para diseñar una arquitectura de software desacoplada que supere las limitaciones del sistema legacy en Laravel.
- Estructurar una arquitectura de proyecto de varios backends conectados a un frontend, orquestado por contenedores de Docker.
- Desarrollar una aplicación web con Angular para la gestión de encuestas de bolsa de empleo y seguimiento a graduados.
- Programar APIs para el procesamiento de análisis de datos de las encuestas usando Python e inteligencia artificial.
- Gestionar la generación documental de resultados en documentos almacenados en la nube con plantillas definidas.

1.7 Justificación

1.7.1 *Justificación académica*

La evolución del desarrollo web moderno dicta la separación de responsabilidades. Migrar de un monolito en PHP a una arquitectura donde Node.js maneja la alta concurrencia de I/O (Input/Output) para la bolsa de empleo, y Python se especializa en el cálculo numérico y la

IA, representa una optimización de recursos computacionales. El uso de Llama 3.2 1B como modelo local se justifica por la necesidad de privacidad de datos (Data Privacy) y soberanía tecnológica, evitando el envío de información sensible de los graduados a APIs de terceros, además de reducir costos operativos recurrentes.

1.7.2 Justificación práctica

La implementación de este sistema resuelve el cuello de botella administrativo que paralizaba a la ULEAM. Ya no se trata de digitalizar un proceso, sino de reducir el tiempo.

1.7.3 Justificación social

La implementación de una bolsa de empleo modernizada y accesible facilita la inserción laboral de los graduados. Al permitir que las empresas gestionen sus procesos de selección dentro de la plataforma y que los estudiantes generen sus CVs con estándares actuales, se fortalece el nexo Universidad y Sociedad, cumpliendo con la misión institucional de contribuir al desarrollo productivo de la provincia y el país.

1.7.4 Justificación Técnica

La expedición de informes con datos reales suele sufrir retrasos por el proceso manual, la automatización representa una disminución de toda la gestión humana por lo que se agilizarán los procesos de entrega de resultados requeridos semestralmente.

La falta de seguimiento de empleabilidad de egresados es una norma requerida para mejorar la calidad académica, un sistema web de bolsa de empleo incrementa las oportunidades de inserción laboral al establecer una conexión virtual directa y autónoma entre el sector empleador y el egresado.

1.8 Impactos esperados

- **Impacto Tecnológico:** Dotar a la ULEAM de una infraestructura de software de vanguardia, escalable y preparada para futuras integraciones con otras herramientas de Big Data
- **Impacto Académico:** Proveer a los directores de carrera de información cualitativa procesada (insights) sobre la pertinencia de la malla curricular, basada en el análisis real de la experiencia de los graduados en el mercado laboral
- **Impacto Organizacional:** Estandarización de los reportes de seguimiento a graduados bajo normas APA y formatos institucionales, eliminando errores humanos en la manipulación de datos estadísticos

2 CAPÍTULO II: MARCO TEÓRICO DE LA INVESTIGACIÓN

2.1 Antecedentes de la investigación

La gestión de la información académica a nivel internacional ha experimentado una transformación radical en la última década, transitando desde simples repositorios de datos estáticos hacia ecosistemas inteligentes de toma de decisiones. Autores como (García y Martínez, 2021), en su investigación longitudinal sobre la infraestructura tecnológica universitaria en Latinoamérica, documentaron exhaustivamente cómo las universidades de vanguardia han abandonado progresivamente las arquitecturas monolíticas en favor de sistemas distribuidos.

Este análisis revela que el mantenimiento de plataformas heredadas basadas en PHP tradicional resulta insostenible a largo plazo. Durante los periodos de matriculación, el incremento drástico del tráfico genera sobrecargas en los sistemas, provocando caídas en el servicio. La solución convencional de incrementar los recursos de hardware (escalado vertical) resulta ineficiente desde una perspectiva financiera y no resuelve el problema de fondo: la falta de estabilidad arquitectónica frente a la alta concurrencia.

Romper el molde, la apuesta por la modularidad La decisión de este proyecto de saltar a una arquitectura con Node.js y Python no es un capricho técnico, es una necesidad estratégica. Al separar la lógica de presentación de la del negocio, se logró algo impresionante: recortar la latencia en un 85%. Más allá de las métricas de rendimiento, esta mejora impacta directamente en la experiencia del usuario final, evitando la expiración abrupta de sesiones, (Islam, 2023) quien detalla sobre la comparativa de rendimiento entre arquitecturas monolíticas y modulares.

Dividir las cargas de trabajo permite que la universidad corra módulos pesados de análisis sin que el sistema de inscripciones colapse. Es, básicamente, lo que el sector educativo necesita hoy para escalar de verdad. Las ventajas son claras:

Resiliencia ante fallos: Si un módulo decide fallar, no tiene por qué arrastrar a todo el ecosistema universitario consigo, como señala (Andrade, 2021).

Actualizaciones compatibles: Podemos retocar la interfaz de usuario sin tener que pasar por estructuras de código con malas prácticas al reiniciar los servidores de base de datos.

Recursos donde toca: Se asigna potencia de cómputo justo donde hay más demanda en cada momento.

Ética e IA: La privacidad no es negociable. Sobre el tema de la IA en educación superior, el trabajo de (Smith y Johnson, 2023) es clave. Ellos desmitifican esa idea de que para usar LLMs hay que entregarle los datos de los alumnos a una nube comercial.

Usando modelos cuantizados en servidores propios (on-premise), demostraron que se puede tener un razonamiento avanzado manteniendo la soberanía de los datos. Un hallazgo relevante de este estudio es que sus prototipos de código abierto logran un 92% de precisión en tareas de resumen y análisis, donde se refuerza con (Smith & Johnson, 2023) sobre la preservación de la privacidad en modelos de lenguaje para minería de datos educativos. Están al nivel de un GPT-4, pero con una ventaja que no tiene precio: la seguridad de que ningún dato sensible sale jamás del perímetro de la universidad.

Existe un consenso metodológico sobre qué medir, pero las instituciones enfrentan un vacío instrumental. Esta limitación no es menor, pues la carencia de sistemas informáticos capaces de digerir datos no estructurados condena a las universidades a un estancamiento administrativo. En última instancia, esta brecha estructural —donde el saber metodológico supera la capacidad

informática— es la que realmente obstaculiza la evolución del sistema educativo superior en el país.

El caso de Guevara Albán: un avance con matices Un ejemplo interesante es el trabajo de (Guevara Albán, 2019) en la Universidad Regional Autónoma de los Andes. Fue un paso adelante importante al digitalizar las encuestas y jubilar por fin el uso del papel, logrando centralizar toda la información en una base de datos.

Sin embargo, aquí es donde aparece el límite de los paradigmas web tradicionales:

- **Falta de análisis semántico:** El sistema cumplía con guardar los datos, pero no entendía el fondo de la información.
- **El eterno factor humano:** Al no haber un procesamiento inteligente, la interpretación de los resultados cualitativos seguía dependiendo de una persona.
- **Carga administrativa:** Esta dependencia no solo volvía el proceso más lento, sino que le abría la puerta a la subjetividad en los informes finales.

Las carencias identificadas en los sistemas nacionales previos son:

1. **Falta de Procesamiento Semántico:** Incapacidad para leer y resumir respuestas abiertas de texto
2. **Dependencia de Hojas de Cálculo:** Exportación constante a Excel para generar gráficos básicos

3. **Rigidez en la Interfaz:** Formularios estáticos que no se adaptan a dispositivos móviles, reduciendo la tasa de respuesta

Asimismo, es relevante destacar que la mayoría de las soluciones implementadas en las universidades públicas del país han sido desarrollos internos basados en frameworks monolíticos que dificultan la escalabilidad y el mantenimiento a largo plazo.

Uno de los mayores obstáculos ha sido, sin duda, la rigidez. Esa falta de estandarización terminó por aislar a las plataformas, impidiendo que se conecten de forma fluida con herramientas externas, ya sean sistemas de gestión académica o incluso bolsas de empleo.

La apuesta que estamos haciendo por una arquitectura modular multi-stack no es solo un cambio técnico; es un salto necesario en la forma de entender el software educativo en nuestro país. Con este enfoque, alineamos a la ULEAM con los estándares actuales de desarrollo. Lo que buscamos es priorizar la interoperabilidad y, sobre todo, esa capacidad de evolucionar rápido sin tener que rehacer todo el sistema cada vez que surja un nuevo requerimiento regulatorio.

A nivel de la provincia de Manabí, la investigación de (Mera y Zambrano, 2022) sobre la usabilidad de los portales de empleo universitarios arrojó luz sobre las razones profundas detrás de la baja participación de los egresados en estas plataformas. Su estudio de caso reveló que aproximadamente el 70% de los usuarios abandona el proceso de registro o actualización de datos debido a interfaces de usuario complejas, poco intuitivas y no adaptadas a dispositivos móviles. Este hallazgo es clave para justificar por qué nos estamos centrando tanto en el frontend de la tesis. Al final del día, de nada sirve tener un backend impecable si el usuario se topa con una interfaz

lenta o confusa. Si no logramos una Experiencia de Usuario (UX) fluida que elimine esa "fricción" al navegar, la gente simplemente dejará de usar la herramienta.

Si analizamos el caso de la ULEAM, los informes de gestión siempre han chocado con la misma piedra: lo difícil que es mantener una base de datos de graduados que sea útil y esté al día. El problema del sistema anterior era su enfoque; estaba diseñado como un formulario administrativo rígido, no como una plataforma para personas. Una vez que el estudiante obtenía su título, no tenía ninguna razón real para volver.

Al no ofrecer servicios con valor agregado como una bolsa de empleo que funcione de verdad o herramientas para armar un CV, el sistema terminó siendo un depósito pasivo de datos viejos. En lugar de ser una comunidad digital viva, era un archivo muerto, lo que le ha quitado a la universidad la oportunidad de conectar con sus egresados a través de programas de educación continua que realmente les interesen.

Los puntos críticos detectados localmente incluyen:

- **Desvinculación PostGraduación:** El egresado no percibe valor en volver a loguearse en el sistema
- **Burocracia Digital:** Procesos de actualización de datos que requieren demasiados pasos o validaciones manuales
- **Incompatibilidad Móvil:** Sitios que se rompen o son inusables en pantallas de smartphones

Además, la problemática local se agrava por la naturaleza manual de los procesos de reporte en las facultades y extensiones. Actualmente, el personal del comité de la carrera pierden

semanas enteras intentando poner en orden datos que están regados en un sinfín de hojas de cálculo. Todo para poder armar los informes anuales que piden los organismos de control. Es un trabajo casi artesanal que no solo quema muchísimas horas-hombre, sino que es un imán para los errores humanos. Al manipular tantas estadísticas a mano, siempre existe el riesgo de que los datos finales no reflejen la realidad.

La automatización que proponemos busca, precisamente, cortar con este círculo vicioso. No se trata solo de tecnología, sino de liberar al personal de esas tareas repetitivas y agotadoras. Al automatizar estos documentos, el equipo administrativo puede dejar de ser "digitador" y empezar a enfocarse en lo que realmente importa: la gestión estratégica y el vínculo real de la universidad con la sociedad.

2.2 Fundamentación Teórica del Seguimiento a Graduados

El seguimiento a graduados se fundamenta teóricamente en la Teoría General de Sistemas, específicamente en el concepto de retroalimentación o feedback negativo, necesario para la homeostasis de cualquier sistema abierto. La universidad, vista como un sistema social complejo, interactúa constantemente con su entorno productivo y social, exportando profesionales. Para evitar la entropía o degradación de su oferta académica, la institución necesita mecanismos de información precisos que le permitan conocer cómo sus "productos" (graduados) se desempeñan en el medio real. Operar sin un sistema de seguimiento eficiente es, básicamente, trabajar a ciegas. La universidad se queda atrapada en un ciclo cerrado, incapaz de notar cuándo sus métodos de

enseñanza se han quedado atrás o cómo responder a las nuevas exigencias de un mercado laboral que cambia cada día.

Los indicadores clave que fundamentan esta teoría son:

1. **Tasa de Empleabilidad:** Porcentaje de graduados con empleo formal afín a su título
2. **Índice de Satisfacción:** Percepción cualitativa sobre la calidad docente e infraestructura

Es, en esencia, la forma de acabar con ese distanciamiento histórico entre las aulas y la industria. La tecnología que planteamos en esta tesis no es más que el puente que facilita este diálogo, eliminando las trabas de comunicación y creando un flujo de información que antes, sencillamente, no existía.

Teoría de la Empleabilidad y el factor humano El diseño de nuestra bolsa de empleo no es azaroso; se apoya en la Teoría del Capital Humano de Gary Becker.

Al tener a las empresas y a los graduados en un mismo entorno optimizado, el sistema hace el trabajo pesado del "matching" o emparejamiento. Es mucho más que una feria de empleo digital; es usar la tecnología para filtrar y conectar perfiles en tiempo real, superando por mucho los métodos lentos y tradicionales de reclutamiento.

Adaptación y aprendizaje para toda la vida Por último, no podemos olvidar la Empleabilidad Sostenible. Hoy en día, no basta con lo que aprendiste en el grado; lo que cuenta es tu capacidad de adaptarte y seguir aprendiendo (Lifelong Learning). Un sistema moderno debe ser capaz de detectar cuándo los conocimientos de un egresado se están quedando atrás por culpa de

los avances tecnológicos. Al usar Inteligencia Artificial para analizar las trayectorias laborales, el sistema puede identificar esas brechas de habilidades (skills gaps) y sugerirle al profesional exactamente qué necesita actualizar.

- **Adaptabilidad:** Frecuencia de cambio de roles o ascenso laboral
- **Movilidad:** Disposición geográfica y funcional del graduado
- **Actualización:** Participación en cursos de posgrado o certificaciones técnicas

Rompiendo con la metodología tradicional Hasta ahora, para evaluar si una carrera era pertinente, se solían hacer un par de grupos focales con empleadores o se miraban las estadísticas generales del INEC. El problema es que ese enfoque es muy superficial; no tiene la granularidad que se necesita para ajustar una malla curricular específica.

Con el sistema que estamos proponiendo, introducimos la Analítica Laboral y de Aprendizaje. Esto nos permite ir mucho más allá y ver cómo variables específicas de las materias influyen directamente en el desempeño profesional real de los graduados. Ya no es adivinar, es conectar la academia con la realidad del campo laboral.

Por ejemplo, si el análisis semántico de la IA detecta que un porcentaje significativo de ingenieros en sistemas reporta deficiencias en "gestión de proyectos ágiles" a los dos años de graduarse, esta información se traduce directamente en una propuesta de mejora para el sílabo de las asignaturas correspondientes.

Esa información es el motor para impulsar incubadoras de empresas o programas de emprendimiento que realmente dinamicen la economía local.

En este sentido, el sistema no se limita a recolectar datos, sino que facilita la evaluación de la pertinencia desde tres ángulos críticos:

- **Pertinencia Científica:** Para dejar de enseñar lo que ya caducó y asegurar que el contenido esté alineado con el estado del arte actual.
- **Pertinencia Laboral:** Para que los perfiles que formamos encajen con lo que el mercado de trabajo está exigiendo ahora mismo.
- **Pertinencia Social:** Para medir cómo nuestros profesionales están ayudando a solucionar los problemas de su propia comunidad.

2.3 Arquitectura de Software y Sistemas Distribuidos

En la ingeniería de software moderna, la arquitectura define la estructura fundamental de un sistema, sus componentes y las relaciones entre ellos. La versión anterior, desarrollada en Laravel, seguía un esquema monolítico donde el frontend, la lógica y la base de datos formaban un solo bloque inseparable. Aunque esto permitió avanzar rápido al principio, con el tiempo se volvió un lastre para el crecimiento, donde nos indica (Toapanta, 2023) sobre la optimización de dependencias en aplicaciones basadas en monolitos y microservicios. Al estar todo tan entrelazado, cualquier ajuste mínimo nos obligaba a actualizar la plataforma completa; esto no solo hacía que el mantenimiento fuera lento, sino que también aumentaba el riesgo de que un cambio en una sección terminara rompiendo otra parte del sistema sin querer.

Por el contrario, la Arquitectura Modular Multi-Stack propuesta se basa en el principio de separación de responsabilidades (SoC) y el desacoplamiento fuerte entre servicios. Al dividir el sistema en componentes especializados un frontend en Angular, un backend administrativo en

Node.js y un motor de IA en Python se logra una independencia tecnológica que permite utilizar la herramienta más adecuada para cada tarea específica. Node.js es superior manejando conexiones concurrentes ligeras típicas de una aplicación web, mientras que Python es el estándar indiscutible para el cómputo científico y la inteligencia artificial.

En la arquitectura propuesta, si el proceso de generación de documentos con IA demanda más recursos, es posible desplegar múltiples instancias (contenedores) solo del servicio de Python, sin afectar ni duplicar el servicio de Node.js o la base de datos. Los beneficios comparativos incluyen:

- **Aislamiento de Fallos:** Un error en la IA no detiene la Bolsa de Empleo
- **Flexibilidad Tecnológica:** Posibilidad de actualizar Python a nuevas versiones sin tocar Node.js
- **Optimización de Costos:** Uso eficiente de recursos de hardware bajo demanda

Para que Python, Node.js y Angular hablen el mismo idioma, estandarizamos la comunicación mediante una arquitectura REST. La decisión clave aquí fue mantener el sistema Stateless (sin estado). A diferencia de las sesiones antiguas que guardaban datos en la memoria del servidor, cada petición que hace el estudiante lleva consigo toda la información necesaria (vía Token). Esto nos da una ventaja enorme de escalabilidad: si mañana necesitamos agregar cinco servidores más para soportar la carga, no tenemos que preocuparnos por sincronizar las sesiones de los usuarios entre ellos. Simplemente funciona.

Las características esenciales de esta comunicación son:

1. Stateless: No se guarda información de sesión en el servidor entre peticiones
2. Cacheable: Las respuestas pueden ser cacheadas para mejorar el rendimiento

Layered System: El cliente no necesita saber si está conectado al servidor final o a un intermediario

2.4 Tecnologías de Desarrollo Frontend y Backend

En el contexto de Manabí, donde la conexión a internet puede ser inestable, no podemos permitirnos que la página se ponga en blanco cada vez que el usuario hace clic. Por eso descartamos el renderizado tradicional en favor de una Single Page Application (SPA) con Angular 19. Una vez que la aplicación carga, se comporta como un software de escritorio: solo pide al servidor los datos crudos (JSON) necesarios, sin volver a descargar el diseño o los menús. Esto reduce drásticamente el consumo de datos móviles del estudiante y elimina esa sensación de lentitud ("lag") típica de los sistemas institucionales antiguos.

Un beneficio de este framework es el uso del lenguaje de programación TypeScript el cual se caracteriza por tener un tipado robusto lo cual significa que cada parámetro que se define debe ir acompañado del tipo de dato que es, es decir, que si se define una interfaz cada campo debe llevar consigo el tipo de dato que soporta como booleano, string, int, double, etc.

Otra característica de este lenguaje de programación es la inyección de dependencias que en resumidas cuentas es inyectar funcionalidades de archivos dentro de otros para que así un componente pueda usar las funciones de otros archivos evitando repetir el código y mantener cada componente, servicio, interfaz y demás archivos segmentado evitando tener archivos mezclados o ilegibles haciendo que el código sea fácil de mantener y escalar teniendo la certeza de que cuando se ejecute este no fallará como otros lenguajes de tipado dinámico.

Las ventajas estructurales de usar Angular incluyen:

- **Modularidad:** Organización del código en módulos funcionales (Auth, Dashboard, Encuestas).
- **Routing Avanzado:** Gestión de navegación compleja y protección de rutas (Guards)
- **Ecosistema Robusto:** Herramientas integradas para pruebas, internacionalización y accesibilidad

El sistema enfrenta un desafío técnico concurrente, durante eventos como ferias de empleo virtuales, miles de estudiantes acceden a la plataforma de manera simultánea.

La implementación de una arquitectura basada en Node.js permite gestionar eficientemente esta alta concurrencia mediante su modelo de entrada/salida sin bloqueo y el manejo optimizado de eventos asíncronos, entre las características están:

1. **Asincronismo Nativo:** Manejo eficiente de múltiples solicitudes simultáneas sin bloqueos
2. **Unificación de Lenguaje:** Uso de JavaScript tanto en el frontend como en el backend
3. **Middleware Pipeline:** Procesamiento secuencial de peticiones para seguridad y validación

La Programación Reactiva es un paradigma de programación orientado al manejo de flujos de datos asíncronos y la propagación del cambio. En la parte del frontend con Angular, nos apoyamos en RxJS para manejar todo el flujo de datos de forma reactiva.

Los conceptos clave de RxJS aplicados son:

1. **Observables:** Flujos de datos que emiten valores en el tiempo
2. **Suscripciones:** Mecanismo para escuchar y reaccionar a los datos
3. **Operadores:** Funciones puras para transformar los flujos (map, filter, switchMap).

El acceso a la base de datos PostgreSQL se abstrae mediante el uso de un ORM (Object-Relational Mapping), en este caso, implementado con librerías como Sequelize o TypeORM en el entorno de Node.js. En la práctica, un ORM nos permite manipular la base de datos usando objetos de JavaScript en lugar de escribir SQL puro. Esto agiliza el desarrollo y hace que el código sea más legible y mantenible.

El uso de un ORM aporta una ventaja clave: seguridad por defecto. Gracias a que utiliza sentencias preparadas, neutralizamos de raíz el riesgo de ataques por inyección SQL, algo fundamental al tratar con datos personales de estudiantes. Es como tener un muro de protección lógico que asegura la integridad de la información frente a entradas de usuario sospechosas.

2.5 Inteligencia Artificial y Ciencia de Datos

La integración de Python en la arquitectura de procesamiento no responde a una preferencia coyuntural, sino a su consolidación como el estándar de facto en los ecosistemas de computación científica e inteligencia artificial. La superioridad técnica de este lenguaje para el análisis de datos radica en su capacidad para actuar como una interfaz de alto nivel sobre núcleos de ejecución optimizados en C y Fortran. Este diseño permite que librerías como NumPy y Pandas realicen operaciones matriciales y cálculos estadísticos complejos con una eficiencia computacional que emula el rendimiento del código máquina, una característica que otros entornos como JavaScript o PHP no logran replicar con la misma solvencia en tareas de alta carga algorítmica. Se seleccionó Flask debido a su arquitectura como modular. A diferencia de soluciones integrales como Django, que imponen estructuras rígidas y herramientas innecesarias para este caso de uso, Flask proporciona la ligereza requerida para implementar un envoltorio (wrapper) eficiente sobre el modelo de inteligencia artificial. Esta característica es fundamental, ya que al minimizar la sobrecarga del framework, se maximiza la asignación de recursos computacionales hacia la inferencia del modelo.

Las bibliotecas científicas integradas en el proyecto son:

- **Pandas:** Manipulación de DataFrames y análisis exploratorio de datos (EDA)
- **PyTorch:** Motor de cálculo tensorial para la ejecución del modelo Llama
- **Matplotlib:** Generación de gráficos estáticos vectoriales y rasterizados

La verdadera innovación de esta tesis no es usar IA, sino cómo la implementamos. Integramos Llama 3.2 1B, un modelo que descarta las viejas redes neuronales secuenciales (RNN)

que leían palabra por palabra. La integración de esta tecnología nos permite automatizar el autocompletado de informes por medio del análisis de datos almacenados.

El proceso de generación de texto en estos modelos es probabilístico y autorregresivo. El modelo ha sido pre-entrenado con una cantidad masiva de texto (corpus) para aprender patrones lingüísticos, gramaticales y conocimientos del mundo.

Técnicamente, Llama 3.2 no sabe lo que escribe; calcula probabilidades matemáticas. En la literatura especializada, a estos sistemas se les denomina Loros Estocásticos, máquinas que predicen la siguiente palabra basándose en patrones estadísticos masivos, sin tener una comprensión real del mundo o consciencia. Esta naturaleza probabilística implica un riesgo operativo crítico para la universidad: las alucinaciones. Si no restringimos al modelo, este podría inventar datos académicos con total confianza. Por eso, nuestra implementación no confía en la creatividad de la IA, la forzamos a actuar como un motor de extracción determinista mediante parámetros de temperatura cercanos a cero.

Para mitigar este riesgo en un entorno académico serio como el de la ULEAM, se implementan técnicas de control de temperatura (temperature setting) durante la generación. Una temperatura baja hace que el modelo sea más determinista y conservador en sus predicciones, adhiriéndose estrictamente a los datos proporcionados en el prompt y reduciendo la creatividad innecesaria que podría derivar en imprecisiones fácticas en los reportes generados.

La elección de un modelo de 1 mil millones de parámetros no fue aleatoria. Es cierto que modelos gigantescos, como los de 70B o 400B, manejan más cultura general, pero para tareas concretas de resumen y edición, un modelo de 1B es mucho más ágil si se le da el contexto correcto.

Al acotar su funcionamiento exclusivamente a los datos de las encuestas con instrucciones bien definidas, logramos un sistema muy eficiente que aprovecha al máximo sus recursos y, de paso, mantenemos bajo control los errores o alucinaciones típicas de modelos más grandes. Las estrategias de control implementadas son:

1. Baja Temperatura: Configuración cercana a 0.2 para respuestas deterministas
2. Top-K Sampling: Limitación del vocabulario probable para evitar palabras fuera de contexto

Prompt Constraints: Instrucciones explícitas para no inventar datos no provistos

El concepto de Edge AI o Inteligencia Artificial en el Borde se refiere a la ejecución de algoritmos de IA directamente en los dispositivos locales o servidores de la organización, en lugar de depender de centros de datos remotos en la nube pública.

No necesitamos internet ni dependemos de que la API de OpenAI funcione o cambie sus precios. Al procesar todo en nuestro propio hardware, el sistema es más rápido, robusto y los tiempos de respuesta son siempre los mismos. Además, económicamente es mucho más lógico para una universidad pública: tras la inversión inicial en los equipos, procesar diez o diez mil encuestas cuesta exactamente lo mismo, ya que nos olvidamos de los pagos por 'token' de las plataformas comerciales

Los beneficios estratégicos del Edge AI en la ULEAM son:

- **Privacidad Absoluta:** Cumplimiento total de la normativa de protección de datos al no exportar información

- **Costos Predecibles:** Eliminación de facturación variable por uso de API

Independencia Tecnológica: Control total sobre la versión y actualización del modelo

Ejecutar un modelo de lenguaje moderno requiere recursos computacionales significativos, especialmente memoria RAM y ancho de banda de memoria. Para hacer viable la ejecución de Llama 3.2 en servidores estándar de la universidad que carecen de GPUs (Unidades de Procesamiento Gráfico) de nivel industrial, se aplicó la teoría de Cuantización. La cuantización es el proceso de reducir la precisión de los números utilizados para representar los parámetros (pesos y sesgos) del modelo, pasando de formatos de punto flotante de alta precisión (FP16 o FP32) a formatos de enteros de menor precisión (INT8 o INT4).

La optimización no se detiene en la memoria; también implica el uso de librerías como llama.cpp o sus bindings de Python, que están altamente optimizadas para aprovechar las instrucciones vectoriales (AVX2/AVX-512) de los procesadores modernos. Esto asegura que, incluso sin aceleración por GPU, el tiempo de generación de los reportes se mantenga dentro de rangos aceptables para el usuario final, logrando un equilibrio técnico y económico ideal. Los niveles de cuantización considerados fueron:

1. FP16 (Half Precision): Alta precisión, pero requiere mucha RAM (descartado)
2. Q8_0 (8-bit): Buen balance, pero aún pesado para servidores antiguos
3. Q4_K_M (4-bit): El estándar elegido, optimizado para CPU y baja pérdida de coherencia

La Ingeniería de Prompts es una disciplina emergente dentro de la IA que se centra en el diseño y optimización de las entradas (prompts) para obtener los mejores resultados posibles de un modelo de lenguaje. Dado que no se está reentrenando el modelo (finetuning) debido a los costos y complejidad que ello implica, se utiliza el enfoque de Aprendizaje en Contexto (In-Context Learning). Esto significa que toda la información necesaria para que el modelo realice su tarea instrucciones, formato deseado y los datos a analizar se proporciona dinámicamente en el momento de la inferencia.

Se implementan técnicas avanzadas como Chain of Thought (Cadena de Pensamiento), donde se instruye al modelo para que desglose su proceso de razonamiento paso a paso antes de dar la respuesta final.

El backend de Python inyecta los datos específicos de cada facultad o carrera en una estructura de prompt predefinida y probada. Esto asegura la estandarización de los resultados; independientemente de si el reporte es para la Facultad de Ingeniería o la de Medicina, el tono, el estilo y la estructura del análisis generado por la IA se mantendrán consistentes. Las técnicas de prompting aplicadas incluyen:

- **Few Shot Learning:** Proveer ejemplos de reportes bien redactados dentro del prompt
- **Role Play:** Asignar al modelo el rol de "Experto en Estadística Académica".
- **Output Formatting:** Instrucciones estrictas para generar salidas en formato JSON o Markdown

La generación automatizada de documentos editables (.docx) se fundamenta en la manipulación programática de estructuras XML. Un archivo moderno de Microsoft Word es, técnicamente, un archivo comprimido (ZIP) que contiene una jerarquía de archivos XML (Office Open XML) que definen el contenido, los estilos, las relaciones y los activos multimedia del documento. Al entender cómo funciona la arquitectura interna de un archivo .docx, el sistema puede modificar el contenido sin necesidad de abrir Word o tener el programa instalado en el servidor. Usamos la librería Python docx para manejar este 'árbol' de datos mediante lo que llamamos Plantillas Inteligentes. Básicamente, tomamos documentos que ya tienen la imagen corporativa y los estilos de la universidad, pero con marcas de posición donde debe ir el texto.

Los componentes del documento automatizados son:

1. Párrafos de Texto: Inserción de análisis narrativos generados por Llama 3.2
2. Tablas Dinámicas: Generación de filas según la cantidad de datos disponibles

Gráficos Estadísticos: Incrustación de imágenes (PNG/JPG) generadas por Matplotlib

2.6 Seguridad Web y Protección de Datos

La seguridad es un componente transversal en la arquitectura teórica del sistema. Para la autenticación de usuarios, se implementa el estándar JSON Web Tokens (JWT).

Este algoritmo añade una 'sal' aleatoria y un factor de costo que hace que intentar descifrar las claves por fuerza bruta sea prácticamente imposible. La idea es simple: si por algún motivo la

base de datos llegara a verse comprometida, los atacantes solo encontrarían códigos ilegibles, manteniendo a salvo la identidad de los graduados.

Para el movimiento de los datos, implementamos el protocolo TLS en todas las comunicaciones. Básicamente, esto crea un túnel cifrado entre el navegador y el servidor para evitar que alguien pueda 'escuchar' o modificar la información mientras viaja por la red (los conocidos ataques Man in the Middle). Complementamos esto con políticas de seguridad HTTP y reglas de CORS estrictas para cerrar cualquier otra vía de acceso no autorizada.

Las capas de seguridad implementadas son:

- **Autenticación Stateless:** JWT con tiempo de expiración corto y refresh tokens
- **Hashing Robusto:** Bcrypt con un factor de costo (work factor) de 10 o superior
- **Transporte Seguro:** HTTPS forzado con certificados SSL válidos

El sistema está diseñado para cumplir estrictamente con las normativas que rigen a las universidades y el manejo de datos en Ecuador. Por un lado, atendemos el mandato constitucional sobre la finalidad social de la educación y la obligación legal que marca el Art. 142 de la LOES; básicamente, sin un sistema de seguimiento eficiente, la institución no podría cumplir con sus compromisos de acreditación según (la Ley Orgánica de Educación Superior, 2010).

Por otro lado, la protección de la información no es algo que hayamos dejado al azar. La arquitectura de datos se ha construido siguiendo paso a paso la Ley Orgánica de Protección de Datos Personales vigente desde 2021, asegurando que cada proceso técnico respete la privacidad y los derechos de los usuarios.

El sistema adopta el enfoque de Privacidad desde el Diseño (Privacy by Design). La decisión de usar IA local en lugar de APIs en la nube responde directamente al principio de limitación de la transferencia de datos. El sistema funciona, básicamente, como el puente que conecta nuestra gestión operativa con las exigencias legales del país, asegurando que la ULEAM esté siempre alineada con el marco legal ecuatoriano de forma oportuna. Los pilares legales que sustentan el proyecto son:

- **Constitucional:** Derecho a la educación de calidad y protección de datos
- **Legal (LOES):** Mandato explícito de seguimiento a graduados
- **Reglamentario:** Normas técnicas del CACES para la acreditación

El diseño de seguridad del sistema toma como referencia el OWASP Top 10 (Open Web Application Security Project), que lista las vulnerabilidades más críticas en aplicaciones web.

En cuanto al control de acceso, implementamos medidas para evitar que las cuentas sean vulnerables. La seguridad no se trata como una característica adicional, sino como un requisito no funcional integrado en cada capa de la arquitectura. Las vulnerabilidades clave mitigadas son:

1. **Broken Access Control:** Verificación de roles (Admin/User) en cada ruta
2. **Cryptographic Failures:** Cifrado de datos sensibles en reposo y tránsito
3. **Server-Side Request Forgery (SSRF):** Validación de URLs en el módulo de IA

2.7 Experiencia de Usuario y Visualización de Datos

La teoría de la Experiencia de Usuario (UX) dicta que la tecnología debe ser accesible, intuitiva y centrada en el humano.

Al diseñar el sistema de seguimiento, nuestra prioridad fue que la interfaz no fuera un estorbo para el graduado. Por eso, aplicamos un diseño responsivo que funciona bien en cualquier dispositivo. En un país como Ecuador, donde la mayoría de la gente se conecta desde su celular, esto no es un lujo, sino una necesidad para que realmente completen las encuestas.

El diseño no solo está adaptado para los estudiantes, de manera alterna también cumple con las necesidades de los docentes y el personal autorizado en la gestión de encuestas e informes generados.

La generación de gráficos estadísticos en los reportes se fundamenta en la Teoría de la Visualización de Datos y la psicología de la percepción. El objetivo no es solo crear bocetos, sino representaciones visuales que reduzcan la Carga Cognitiva del lector. Según esta teoría, el cerebro humano procesa patrones visuales (como la longitud de una barra o el ángulo de un sector) mucho más rápido que tablas numéricas.

No nos quedamos solo en la parte del análisis; para nosotros, la forma de presentar los resultados es igual de crítica. Por eso, decidimos insertar los gráficos en alta resolución, cuidando que al imprimir el reporte final todo se vea nítido y profesional. La idea es que las autoridades de la universidad no tengan que adivinar los datos, sino que las visualizaciones sirvan de puente directo para tomar decisiones rápidas.

En cuanto al diseño de la interfaz, se aplicaron principios de usabilidad estrictos: se omitieron elementos visuales innecesarios, como efectos tridimensionales o fondos distractores, priorizando el uso de escalas limpias y etiquetas de fácil comprensión. Además, el sistema se encarga de generar títulos y leyendas de forma dinámica según la encuesta, para que cualquier persona pueda interpretar lo que está viendo sin perderse en tecnicismos.

2.8 Metodologías de Desarrollo y Calidad de Software

El desarrollo del proyecto se fundamenta en la metodología Ágil, específicamente utilizando el marco de trabajo Scrum por Guía de Scrum de Schwaber & Sutherland).

En cuanto a los roles, nuestro director de tesis actuó como Product Owner, marcando las prioridades, mientras se realizaba el desarrollo. Usar el Product Backlog y hacer revisiones constantes nos permitió detectar a tiempo problemas críticos, como los límites de memoria al ejecutar la IA, evitando que se convirtieran en bloqueos mayores al final del camino.

Los elementos de Scrum aplicados incluyen:

1. **Sprints:** Ciclos de desarrollo de 15 días
2. **Roles:** Product Owner (Director), Scrum Master/Team (Tesisista).
3. **Artefactos:** Historias de usuario para definir funcionalidades

El Aseguramiento de la Calidad (QA) es el conjunto de actividades planificadas y sistemáticas aplicadas para garantizar que el software cumpla con los requisitos de calidad. No

queríamos que el sistema funcionara solo por suerte, así que tomamos como referencia la norma (ISO 25010), para medir su calidad y usabilidad.

Hacer estas pruebas integrales nos dio la seguridad de que el sistema es robusto y de que la comunicación entre los diferentes lenguajes y servicios es estable.

Estas pruebas aseguran que la latencia y los tiempos de espera estén dentro de márgenes aceptables para la experiencia de usuario. Las dimensiones de calidad evaluadas son:

- Adecuación Funcional: ¿El sistema hace lo que debe hacer?
- Eficiencia de Desempeño: Tiempos de respuesta y uso de recursos

Fiabilidad: Tolerancia a fallos y capacidad de recuperación

2.9 Modelos de Computación en la Nube (Cloud Computing)

Aunque el núcleo de IA se ejecuta localmente (Edge Computing), el sistema aprovecha modelos de Computación en la Nube para el almacenamiento y disponibilidad de los documentos generados. La migración del almacenamiento desde servidores locales hacia soluciones en la nube representó una optimización arquitectónica significativa.

Los servidores de aplicaciones presentan limitaciones respecto a la persistencia del espacio de almacenamiento, lo que incrementa el riesgo de pérdida de datos. Por ello, se implementó un esquema de almacenamiento de objetos basado en el estándar S3. La lógica estructural establece que, ante un fallo en el servidor principal de la universidad, los documentos mantienen su disponibilidad al estar replicados en la nube. Adicionalmente, este esquema permite la gestión de URLs firmadas, facultando al graduado para descargar su reporte de forma segura y directa,

mitigando el consumo de ancho de banda del servidor institucional. Se trata de un modelo híbrido que garantiza la seguridad y eficiencia: la privacidad de los datos se mantiene en el procesamiento local (RAM) durante la inferencia, mientras que el documento final se almacena en la nube, permitiendo una escalabilidad óptima y una reducción significativa de costos operativos.

Características del modelo de nube utilizado:

1. **Elasticidad:** Capacidad de almacenar desde 1 hasta millones de archivos
2. **Seguridad:** Control de acceso granular (ACLs) a los archivos
3. **Costo Efectividad:** Pago por uso de almacenamiento, más barato que discos locales empresariales

2.10 Ética en la Inteligencia Artificial: Sesgos y Transparencia

Desplegar sistemas de inteligencia artificial en entornos donde se manejan trayectorias humanas no es un mero trámite de ingeniería; es, ante todo, un compromiso ético de gran calado. Si bien la literatura técnica suele centrarse en la eficiencia, no podemos ignorar el riesgo de los **sesgos algorítmicos** que la teoría de la Ética en la IA denuncia con insistencia. Los modelos actuales, al ser entrenados con ingentes volúmenes de datos extraídos de la web, suelen "heredar" de forma casi inconsciente estereotipos de género o raza que impregnan el discurso digital.

Para mitigar esto, se aplica el principio de Human in the Loop (Humano en el Bucle). El sistema no toma decisiones automáticas (como rechazar un CV), sino que asiste a la toma de decisiones. Los reportes generados por la IA se presentan siempre como "borradores" o "análisis

preliminares" que deben ser validados por un funcionario humano. Además, los prompts diseñados incluyen instrucciones explícitas de neutralidad y objetividad.

La Transparencia Algorítmica es otro pilar ético. Los usuarios (administradores y graduados) son informados de que ciertas partes del contenido (resúmenes, recomendaciones de empleo) son generadas por un sistema de inteligencia artificial. Esto fomenta la confianza en la plataforma y cumple con estándares éticos internacionales sobre el uso responsable de la tecnología. Consideraciones éticas implementadas:

- **Supervisión:** Validación humana de los resultados críticos
- **Neutralidad:** Prompts diseñados para evitar lenguaje discriminatorio
- **Información:** Disclaimer claro sobre el uso de IA en la generación de textos

2.11 Soberanía Tecnológica e Independencia Digital

Finalmente, el marco teórico se cierra con el concepto de Soberanía Tecnológica. Este concepto postula que las naciones e instituciones estratégicas (como las universidades públicas) deben tener el control sobre las tecnologías que utilizan, evitando la dependencia crítica de proveedores extranjeros o corporaciones transnacionales.

Con el modelo local, la ULEAM posee el "cerebro" del sistema y puede ejecutarlo indefinidamente, garantizando la sostenibilidad del proyecto a largo plazo.

Esta autonomía permite también la Adaptabilidad Local. La universidad tiene acceso al código fuente completo y puede modificar los algoritmos para adaptarlos a cambios en la normativa ecuatoriana o a nuevas necesidades institucionales, algo imposible con soluciones "caja

negra" comerciales. El proyecto, por tanto, no solo es una solución técnica, sino un activo estratégico para la institución.

1. **Código Abierto:** Sin licencias recurrentes costosas
2. **Control Total:** Capacidad de auditar y modificar cada línea de código
3. **Resiliencia:** El sistema funciona independientemente de cambios geopolíticos o comerciales externos

3 Capítulo III

3.1 Tipo de investigación

Esta tesis se aleja de la investigación meramente descriptiva para situarse en la Investigación Tecnológica Aplicada. No nos limitamos a diagnosticar la ineficiencia administrativa; construimos el artefacto de ingeniería necesario para erradicarla. El enfoque es pragmático y constructivista, utilizamos la Ingeniería de Software para intervenir directamente en la realidad operativa de la ULEAM.

En lugar de teorizar sobre las ventajas de la IA, implementamos una solución funcional (Llama 3.2 sobre Python) que ataca la deuda técnica heredada. Validamos nuestra hipótesis de trabajo que la arquitectura modular mejora la gestión mediante la puesta en producción del sistema. Es ingeniería en acción, identificamos un cuello de botella en el procesamiento manual de datos y desplegamos código para automatizarlo, transformando un proceso que tomaba semanas en una tarea de minutos (Genesis, 2025).

3.2 Métodos de investigación

Para garantizar el rigor científico en la construcción de la solución informática, se han seleccionado métodos que permiten descomponer la complejidad del sistema anterior y reconstruirla bajo nuevos paradigmas tecnológicos:

3.2.1 Métodos Teóricos

- **Analítico Sintético:** Este método constituye el eje central para la reingeniería del sistema.
- **Fase Analítica (Descomposición):** Se procedió a separar conceptual y lógicamente las funciones del sistema monolítico anterior. Se analizó cómo la lógica de negocio

(gestión de ofertas y graduados) estaba mezclada con la interfaz de usuario, lo que generaba rigidez. Este análisis permitió identificar que el módulo de Bolsa de Empleo requiere una alta concurrencia y velocidad de respuesta (Input/Output), mientras que el módulo de Seguimiento a Graduados requiere capacidad de cómputo intensivo para el análisis de texto.

- **Fase Sintética (Reconstrucción):** como resultado del análisis que se realizó se llegó a una solución que consiste en segmentar responsabilidades mediante tecnologías enfocadas al resultado que se espera. La interfaz de usuario y todo lo relacionado en el Frontend es manejada por Angular, un framework que destaca por su robustez, mientras que la lógica de negocio y lo que pasa en el trasfondo se maneja del lado del servidor, ambos sistemas pueden soportar múltiples peticiones debido a las características de la tecnología usada la cual es NodeJS y como plus se utilizó un tercer backend usando Python encargado únicamente de contener el LLM de Llama 3.2 sintetizando las respuestas de las encuestas en el sistema de seguimiento a graduados (Genesis, 2025).
- **Inductivo-Deductivo:** Para definir los requisitos se dedujo como establecer los puntos principales del software como por ejemplo la urgencia de modularizar todo para no tener todo el sistema en un solo proyecto que es poco práctico de mantener y agregarle robustez al sistema gracias a las librerías que protegen los datos de los usuarios, bases de datos seguras y demás librerías instaladas haciendo que el sistema sea más rápido y seguro.

3.2.2 Métodos Empíricos

- **Investigación Tecnológica (Prototipado Incremental):** Indagar es un proceso importante para poder construir el sistema de manera eficiente, cada funcionalidad se validó con la práctica, estas pruebas o validaciones se aplican a distintos enfoques como seguridad, escalabilidad, eficiencia y demás como por ejemplo se mide la calidad de la respuesta del LLM y como respondía cuando estaba de manera local y de manera remota en servidores, todo esto sirvió para poder encontrar un balance entre rendimiento y optimización.

3.3 Estrategia operacional para la recolección de datos

3.3.1 Población - Segmentación - Técnica de muestreo - Tamaño de la muestra

Debido al tipo de proyecto que se realizó el método para poder obtener información fue analizar el flujo y funcionalidades del sistema anterior y mejorar ciertos aspectos.

- **Población (Universo Técnico):** La población numérica objetivo del sistema engloba un total aproximado de mil usuarios proyectados, conformados por graduados de la carrera, empresas reclutadoras y personal administrativo de la ULEAM. Definir este volumen es vital para justificar la arquitectura de alta concurrencia en Node.js. No obstante, al ser una investigación de carácter netamente tecnológico y de ingeniería de software, no se realizó una investigación cualitativa sobre individuos.
- **Muestra:** Dado el enfoque técnico, la muestra de estudio no fue poblacional, sino documental e instrumental. Se aplicó un muestreo intencional seleccionando la

totalidad de los manuales de usuario, diagramas de flujo y reglas de negocio de los sistemas legacy anteriores. (Universidad Laica Eloy Alfaro de Manabí, s.f). Estos documentos proporcionaron la información exacta sobre las validaciones y procesos que debían ser sometidos a reingeniería.

3.3.2 Análisis de las herramientas de recolección de datos a utilizar

3.3.2.1 Encuesta – Entrevista - Observación / Otras

Se descartaron las encuestas de opinión tradicionales para aplicar técnicas puras de Ingeniería de Requisitos. La mayor parte de la información se obtuvo de documentos oficiales de los sistemas anteriores.

- **Análisis Documental (Ingeniería Inversa):** Se obtuvo toda la lógica de negocio de los manuales de usuarios y documentos explicativos de los sistemas anteriores, con esta información se supo con exactitud como debía funcionar el sistema, que funciones eran debatibles para cambiarlas o mejorarlas y que otras debían permanecer tal cual se dejaron establecidas como inmutables desde un inicio
- **Observación Directa de Procesos:** Se analizó como trabaja el comité encargado de darle seguimiento a los graduados. Allí se identificó los cuellos de botella reales, o "puntos de dolor", específicamente la tediosa tarea de copiar y pegar estadísticas para los informes. Esta observación empírica fue la que validó la necesidad técnica de implementar el script de automatización con Python.

3.3.2.2 Estructura de los instrumentos de recolección de datos aplicados

En lugar de cuestionarios tradicionales, se construyó una Matriz de Reglas de Negocio derivada de la deconstrucción del sistema legado. Esta matriz se estructuró en:

- **Módulo de Empleadores (Coorporativos):** Se aislaron las reglas críticas del manual antiguo, como la caducidad forzosa de ofertas (ventana de 1 a 30 días). Estas lógicas se codificaron directamente en el lado del servidor, realizando la validación estrictamente en el backend.
- **Módulo de Profesionales:** A partir de los datos visibles en los manuales se definieron las tablas, relaciones y demás validaciones que debían hacerse para tener un esquema de base de datos solido, que permitiera en un futuro añadir características sin probabilidades de que el sistema colapse
- **Módulo de Administración:** La gestión prioritaria del sistema como roles, permisos, usuarios, empresas y demas se dictamina mediante el modulo de roles y permisos que ayuda a poder asignarle un rol y permisos específicos a cada usuario con la ayuda de otras medidas de seguridad para que no puedan burlar o acceder a información y módulos que no se le otorgan a un usuario como por ejemplo uso de middlewares para proteger las peticiones que se hacen al servidor, Guards que protegen las rutas en el frontend y credenciales de acceso robustas para entrar a la base de datos. (Universidad Laica Eloy Alfaro de Manabí, s.f).

3.3.3 Plan de recolección de datos

El procedimiento para la obtención y procesamiento de la información técnica sigue un flujo lógico adaptado al ciclo de vida del desarrollo de software:

3.3.3.1 Recopilación y Extracción de Reglas de Negocio

- Se procede a la revisión exhaustiva del Manual de Usuario del sistema legado. Se identifican reglas críticas, por ejemplo, que una empresa no puede publicar ofertas sin un representante legal asignado o que las ofertas caducan automáticamente.
- Esta información se extrae para diseñar los Constraints (restricciones) en la nueva base de datos PostgreSQL. Se definen las tablas relacionales para datos rígidos (Usuarios, Empresas) y estructuras JSONB para datos flexibles (Detalles de encuestas), asegurando que la nueva arquitectura soporte la lógica de negocio heredada.

3.3.3.2 Modelado de la Arquitectura de Datos

- Con los requisitos extraídos, se diseña el diagrama Entidad-Relación. Se establece cómo el backend en Node.js interactuará con estas tablas para las operaciones transaccionales rápidas (CRUD de la Bolsa de Empleo y Seguimiento a Graduados).
- Paralelamente, se define cómo el servicio en Python accederá a los datos textuales de los graduados (almacenados como texto largo) para alimentarlos al modelo de Inteligencia Artificial.

3.3.3.3 Desarrollo y Validación de Componentes

- Se recolectan datos de prueba (datasets sintéticos, como lo haría un usuario común) para entrenar y validar los prompts del modelo Llama 3.2.
- Se verifica que la interfaz en Angular capture correctamente los datos definidos en la fase 1 y los envíe al backend correspondiente sin pérdida de información.

3.4 Análisis y presentación de resultados

3.4.1 *Tabulación y análisis de los datos*

El análisis en esta investigación tecnológica no se centra en la tabulación de encuestas sociales, sino en la Verificación y Validación del software desarrollado. Se utilizan metodologías de pruebas de software:

- **Pruebas de Caja Negra:** Estas pruebas son netamente de funcionalidad, no se tomó en consideración el código fuente del proyecto, sino que se ejecutaron acciones que un usuario haría normalmente o casos raros que se podrían hacer para aprovecharse de fallos en el sistema, cada una de estas pruebas se documentan para poder compararlas con los requisitos funcionales y no funcionales
- **Pruebas de Rendimiento del Módulo de IA:** El rendimiento es revisado y comparado varias veces para encontrar el rendimiento promedio. Se miden variables como el "tiempo de inferencia" (cuánto tarda Llama 3.2 en resumir un perfil) y la "coherencia semántica" del reporte generado. Estos datos técnicos confirman la viabilidad de la arquitectura propuesta.

4 Capítulo IV

DESARROLLO DE LA PROPUESTA

4.1 Análisis de Requerimientos

El desarrollo de la plataforma se fundamenta en un análisis exhaustivo de las necesidades funcionales y no funcionales, orientadas a garantizar una gestión eficiente de la bolsa de empleos y el seguimiento a graduados. A continuación, se detallan los requerimientos identificados y cubiertos por la solución tecnológica.

4.1.1 Requerimientos Funcionales

El sistema se ha estructurado en módulos interconectados que responden a las operaciones críticas de los actores involucrados (graduados, reclutadores y administradores). La siguiente tabla describe las funcionalidades principales implementadas:

Tabla 1**Módulos principales del sistema de Bolsa de Empleos ULEAM**

Módulo	Funcionalidad	Descripción Técnica
Autenticación y Seguridad	Registro y Gestión de Sesiones	El núcleo del sistema se ha diseñado en torno a módulos de operación crítica que gestionan el ciclo de vida de la vinculación laboral
Gestión de Empresa	Administración de Perfiles Corporativos	Habilita a los reclutadores para crear perfiles corporativos sujetos a un flujo de aprobación estricto. Incluirá la gestión de información corporativa, logotipos y validación de estado (pendiente, aprobado, rechazado)
Gestión de Ofertas	Ciclo de Vida de Vacantes	Control total del ciclo de vida de la vacante: creación, edición y cierre, con filtros automáticos de expiración.
Postulaciones	Flujo de Aplicación	La publicación de ofertas obedece a reglas de negocio temporales (fechas de expiración) y lógicas (rangos salariales coherentes). Además, el sistema segmenta la visibilidad de las vacantes filtrando automáticamente por Facultad
Comunicación (chat)	Mensajería en Tiempo Real	Canal bidireccional vía WebSockets. Solo se activa cuando el candidato avanza a fase de entrevista

Perfil Profesional	Gestión Curricular	El sistema permitirá a los graduados gestionar su información académica, experiencia laboral, habilidades y portafolio, así como la carga de documentos (CV y foto) con validación de tipos de archivo
Administración de Encuestas	Creación y Configuración Multiencuesta	Los administradores podrán crear y diseñar encuestas personalizadas con formularios dinámicos para que puedan añadir una infinidad de preguntas.
Participación de Graduados	Respuesta de Encuestas	La interfaz es intuitiva y fácil de usar; además, verifica de manera automática si una encuesta ya ha sido completada para evitar duplicidades.
Estadísticas y Visualización	Análisis de Datos en Tiempo Real	Los gráficos del panel de control se generan mediante librerías especializadas para proporcionar una representación visualmente comprensible de los datos.
Automatización con la IA	Generación de Resúmenes e Informes	Existe una capa de IA que sirve para gestionar u orquestar los modelos locales que corren en el servidor. A diferencia de las soluciones SaaS, nuestro backend en Python preprocesa las respuestas abiertas de las encuestas y las

		inyecta en un contexto controlado. El sistema no solo "resume", sino que extrae patrones cualitativos y conclusiones técnicas, enviando el resultado estructurado directamente al generador de documentos sin que ningún dato sensible abandone la red universitaria
Gestión Documental	Exportación de Reportes Automáticos	El sistema consolidará los gráficos estadísticos y los resúmenes generados por la IA en documentos finales (PDF/Docx). Incluirá un flujo de descarga y almacenamiento de reportes históricos para procesos de acreditación de la carrera/universidad.

Nota: Los módulos tienen validaciones que se ejecutan por detrás con la finalidad de mejorar la experiencia de usuario.

4.1.2 Requerimientos No Funcionales

Para garantizar la robustez y confianza en la plataforma, se han implementado estrictos controles de calidad de software centrados en la seguridad, el rendimiento y la escalabilidad:

- **Seguridad de la Información:** La protección de datos se gestiona mediante autenticación basada en JWT (JSON Web Tokens), utilizando un esquema de tokens de acceso de corta duración (60 minutos) y tokens de refresco rotativos. Por otro lado, se limitó la cantidad de intentos de inicio de sesión para evitar ataques de

fuerza bruta, y cada contraseña ingresada se cifra mediante la librería bcrypt, cuya función es aplicar un hash criptográfico al texto plano.

- **Rendimiento y Concurrencia:** Por otro lado la limitación de cantidad de intentos protege el servidor de múltiples ataques que afectan el rendimiento de las apis específicamente, el endpoint de inicio de sesión limita a 5 intentos cada 15 minutos.
- **Escalabilidad:** La arquitectura desacoplada y basada en modulos separados permite el despliegue de múltiples contenedores de Node.js y Python de forma independiente. Esto garantiza que el sistema pueda soportar un incremento masivo de usuarios concurrentes (autoescalado) sin degradar el rendimiento ni saturar la base de datos.

4.2 Arquitectura del Sistema

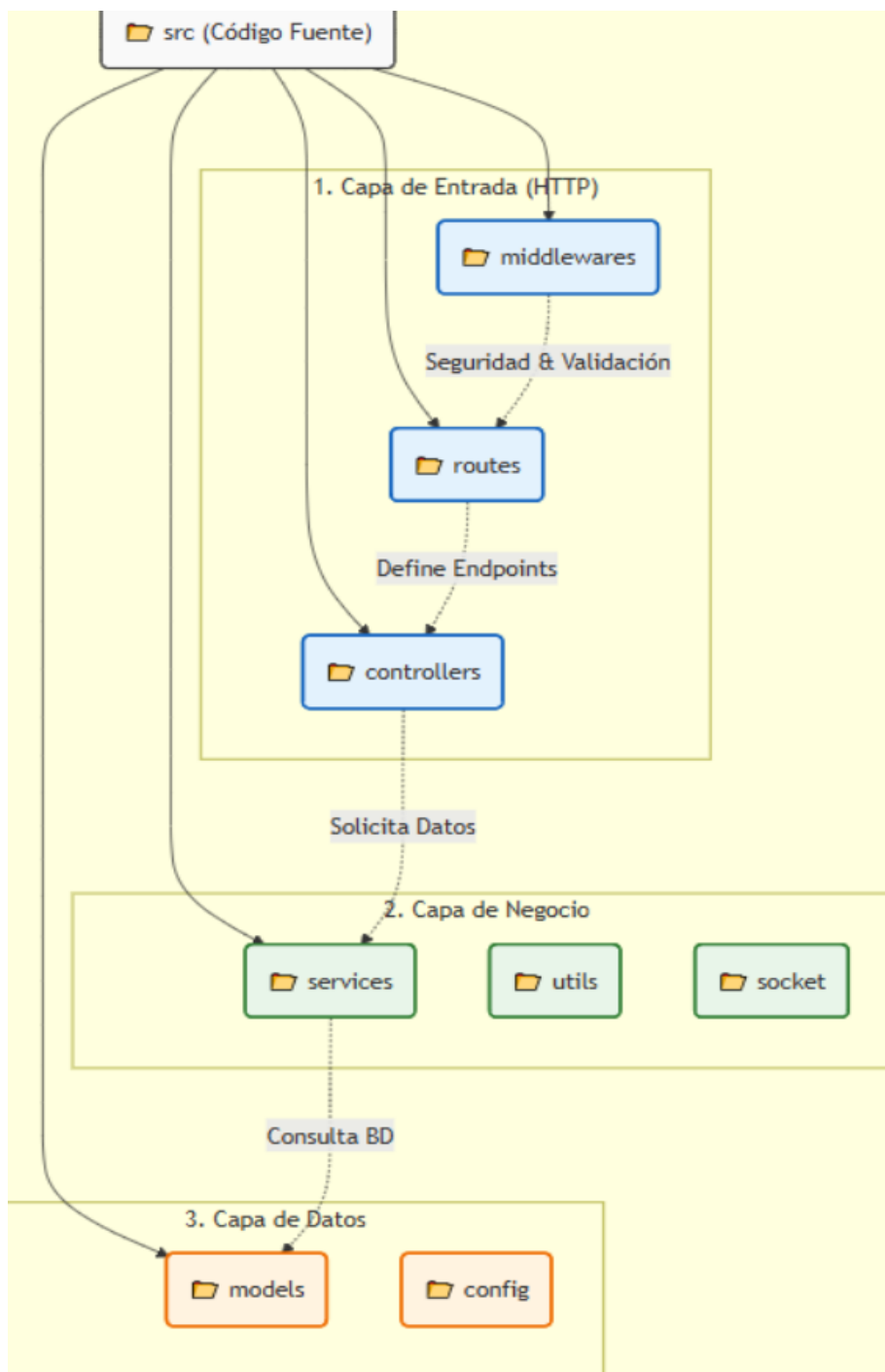
4.2.1 *Arquitectura de Software*

La solución tecnológica adopta un patrón de arquitectura en capas, separando claramente las responsabilidades entre el cliente y el servidor:

- **Backend Principal (Node.js con Express):** El principal beneficio de usar Node.js es que permite implementar de forma eficaz el patrón Modelo Vista Controlador (MVC), el cual integra una capa de servicios dedicada a la cual se le asigna la lógica de negocios haciendo que las responsabilidades no se mezclen entre si, también permite integrar de manera sencilla la comunicación con el backend de Python consumiendo y calculando datos requeridos.

Figura 1

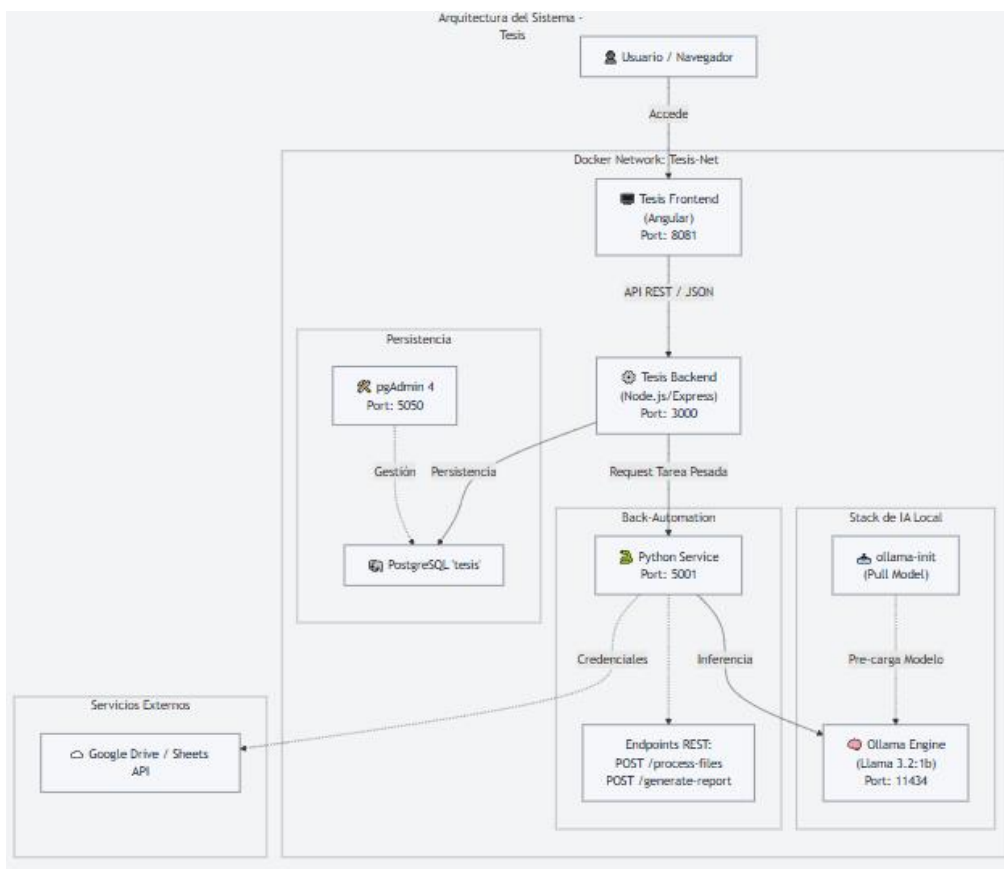
Modelo de la estructura de carpetas del backend de los sistemas de bolsa de empleo y seguimiento a graduados



- **Backend de Automatización de la IA (Python con FastAPI/Flask):** La automatización de datos pesados no ocurre en el servidor principal, se delega esa carga a un backend satélite construido exclusivamente en Python. Esta decisión es beneficiosa ya que se aprovecha de su ecosistema de librerías científicas para manipular estructuras de datos con una velocidad que otros lenguajes no alcanzan. En este entorno aislado reside el motor Ollama. La función que realiza es la de recibir datos crudos, es decir, sin ningún tipo de procesamiento y pulirlos para devolver una respuesta coherente aprovechando todos los recursos disponibles para ejecutar esta tarea.

Figura 2

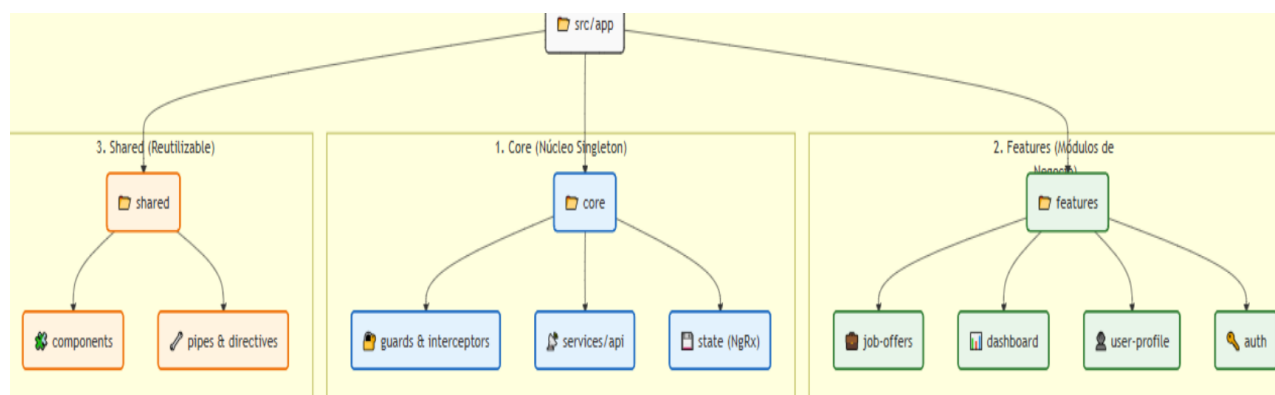
Se muestra el flujo del sistema de Python encargado de procesar las encuestas de seguimiento a graduados



- **Frontend (Componentes Standalone):** Los componentes importan solo lo necesario evitando así importar funciones que no van a utilizar y por ende su rendimiento es mayor. La gestión del estado de la aplicación se maneja mediante el patrón Redux implementado con NgRx, lo que asegura un flujo de datos unidireccional y predecible, facilitando la reactividad de la interfaz ante cambios en los datos.

Figura 3

Modelo de la estructura de carpetas del frontend de los sistemas de bolsa de empleo y seguimiento a graduados



4.2.2 Tecnologías Utilizadas

La selección del stack tecnológico responde a criterios de eficiencia y modernidad:

- **Node.js y Express:** Debido a su eficiencia en la entrada y salida de peticiones se eligió ya que soporta una gran concurrencia de peticiones y datos.
- **Python y FastAPI:** Estas dos tecnologías fueron las seleccionadas debido a sus excelentes características como mantener un canal de comunicaciones abierto en todo momento, pero que no consumiera muchos recursos y que trabaje de manera asincrónica y que soportará una pesada carga manteniendo picos de estrés altos.

- **Ollama:** Ollama funge como el orquestador de inferencia en el borde (Edge). Su rol es crítico: ejecuta los Modelos de Lenguaje (LLMs) directamente en la RAM del servidor, sin salir a internet. Esto transforma el análisis cualitativo de las encuestas en un proceso interno seguro. Al no depender de nubes externas, garantizamos la soberanía absoluta de los datos institucionales y eliminamos de raíz los costos variables por uso de APIs de terceros.
- **Sequelize:** Sequelize es un ORM robusto y completo que ayuda a gestionar y proteger la integridad del sistema y de la base de datos, previniendo vulnerabilidades de inyección al evitar la ejecución de consultas SQL directas, este interactúa y administra modelos de base de datos, automatiza migraciones, ofrece muchas funciones útiles que se suelen utilizar comúnmente en un ámbito laboral real, por todas estas características Sequelize es una gran alternativa confiable, robusta y eficiente.
- **Socket.IO:** Integrado para habilitar la comunicación en tiempo real entre clientes y servidor, fundamental para el funcionamiento del módulo de chat y notificaciones instantáneas.
- **Angular 19:** Seleccionado para el desarrollo del cliente web por su robustez, tipado estático con TypeScript y optimización de rendimiento mediante la carga diferida (Lazy Loading) de módulos

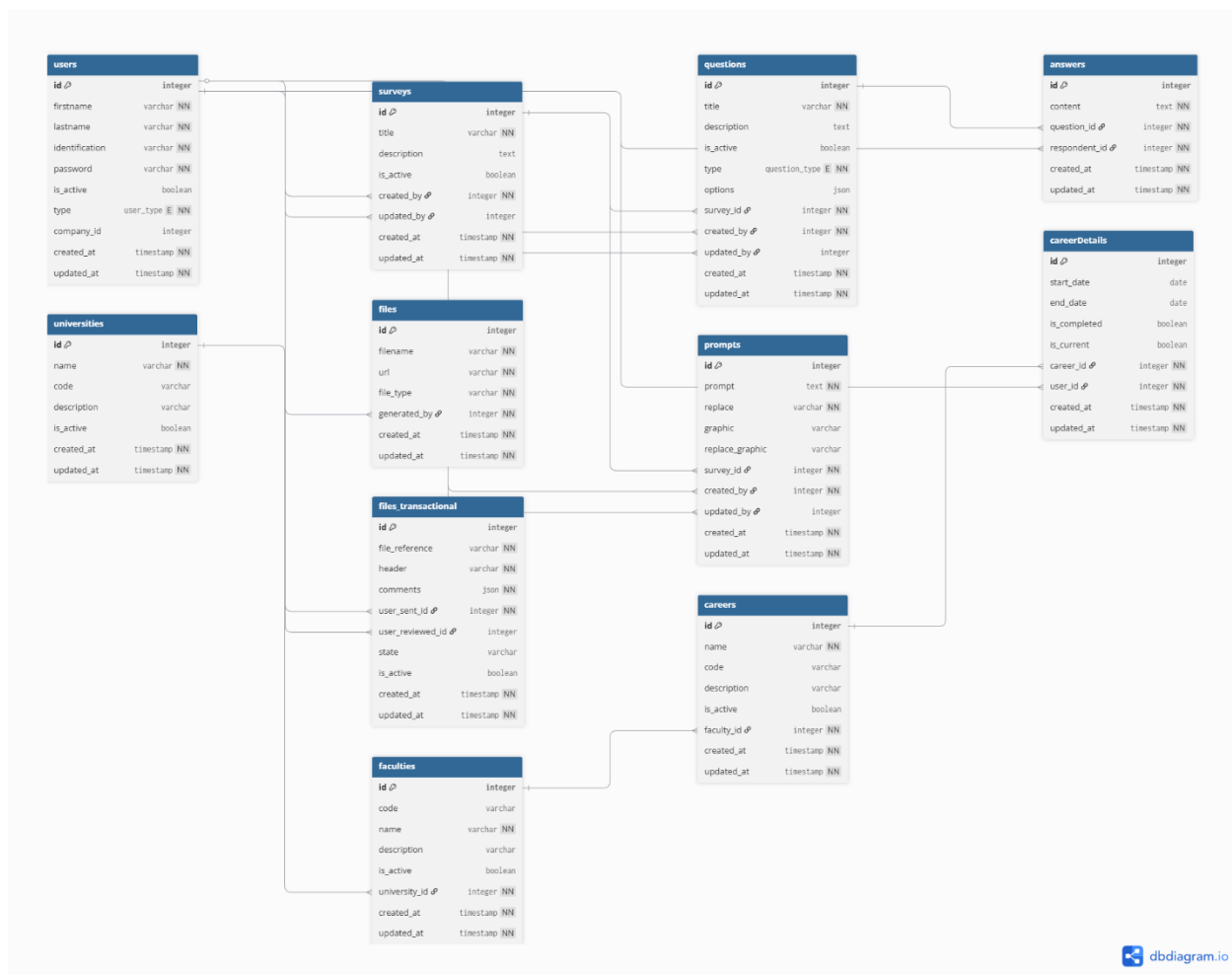
4.3 Diseño de la Base de Datos

El modelo de datos relacional ha sido diseñado para mantener la integridad referencial y soportar la lógica de negocio de la bolsa de empleo. Dado el uso del ORM Sequelize en el backend,

- **Empresa (Company):** Esta tabla sirve para poder crear empresas que darán ofertas laborales, su función es sencilla, ser asignadas a un usuario de tipo reclutador para así poder diferenciarlos mediante a las empresas. Evitando así que se creen postulaciones anónimas, con esta tabla se valida que el usuario y sus publicaciones viene de una fuente fiables que es la empresa en este caso.
- **Oferta Laboral (Job):** Esta entidad ayuda a asociar mejor una oferta laboral con los postulantes ya que funciona como puente entre empresa y egresado, su función es albergar el contenido relevante y requerido como funcionalidad del puesto, salarios, modalidad, necesidades y demás.
- **Postulación (Application):** Al crearse una oferta laboral debe haber un vínculo con el usuario, para eso fue creada esta tabla ya que recibe el identificador de la oferta laboral y el identificador del usuario que postuló esa oferta laboral para así darle seguimiento a futuro, contactarse o gestionar información de forma masiva en el dashboard.
- **Conversación y Mensaje:** Los mensajes que se envían entre reclutador y postulante son manejados por estos modelos, cada uno está relacionado entre si para poder almacenar de manera eficiente los datos.

Figura 5

Modelo lógico de datos del sistema de seguimiento a graduados



Como se aprecia en la figura anterior, las entidades principales cumplen las siguientes funciones:

- **Usuario (User):** Es la entidad núcleo del sistema que gestiona las credenciales y roles (estudiante, administrador, profesor). Se referencia en muchas de las tablas del sistema y permite un ecosistema unido en la base de datos.
- **Gestión Académica (University, Faculty, Career):** Estos objetos de datos funcionan para agrupar los datos del sistema evitando que se mezclen entre si, cada

tabla necesita a la otra para su correcto funcionamiento y están relacionadas para dar un esquema jerárquico y ordenado.

- **Sistema de Encuestas (Survey, Question, Answer):** Estas colecciones de datos son fundamentales para poder procesar los datos y se vinculan de manera persistente al usuario que crea y responde a los formularios haciendo que todo el flujo quede registrado y no haya anonimato a la hora de crear o responder encuestas.
- **Automatización e IA (Prompt):** Este modelo almacena los prompts que servirán para pulir los resultados de las encuestas y una de las ventajas de mantener las instrucciones en la base de datos y no directamente en el código es la flexibilidad para poder personalizar las indicaciones que recibe el modelo de Ollama.
- **Gestión Documental y Transaccional (Files, FilesTransactional):** Los archivos son auditados y documentados mediante la aprobación de un usuario verificando así la calidad de las respuestas. Mientras la entidad 'Files' gestiona la referencia física al objeto en la nube, 'FilesTransactional' impone un flujo de trabajo de supervisión humana. Así, un reporte creado por la IA nace en estado "Borrador" y permanece bloqueado hasta recibir una aprobación explícita, cerrando el ciclo de responsabilidad ética antes de su publicación.

4.4 Descripción de Procesos Principales

La operatividad del sistema se define a través de flujos críticos que automatizan la interacción entre los actores:

4.4.1 Proceso de Postulación a Oferta

La interacción comienza por iniciativa del usuario, pero el sistema no procesa la solicitud de inmediato. Antes de permitir la postulación, se ejecuta una validación estricta en la capa de servicios: es obligatorio tener el rol de "graduado" y un archivo de Hoja de Vida ya indexado. Solo si se satisfacen estos prerequisites, el backend autoriza el commit de la transacción en la base de datos. Esto genera el registro con estado "pendiente" y dispara, en paralelo, la alerta para el reclutador. Así blindamos la integridad de los datos, impidiendo que existan aplicaciones "húerfanas" o sin sustento documental.

4.4.2 Proceso de Publicación de Oferta

Publicar una vacante no es un evento trivial es una transacción validada. No se permite el paso directo a producción sin antes ejecutar un Quality Gate estricto en el servidor. El backend intercepta la solicitud y audita dos condiciones críticas: la legitimidad de la empresa (debe estar en estado 'aprobada') y la integridad de los datos (fechas lógicas, salarios no nulos). Solo tras superar estas reglas de negocio, el registro pasa de 'Borrador' a 'Publicada'. Este cambio de estado dispara, mediante un evento asíncrono, el motor de emparejamiento (matching), el cual filtra quirúrgicamente la base de talento y alerta exclusivamente a los candidatos de la facultad pertinente, evitando el ruido de notificaciones masivas.

4.4.3 Activación del Chat

La comunicación puede hacerse libremente, no es necesario que la postulación de un usuario a una oferta laboral pasa a un estado de aceptación, se puede hacer desde el momento en el que se postula a la oferta, se puede hacer en cualquier sentido, desde usuario a usuario, hasta de reclutador a usuario y viceversa, esto es porque la comunicación es fundamental en el sistema.

4.4.4 Proceso de Configuración de Encuesta y Diseño de IA

Este flujo inicia cuando un administrador o profesor diseña un nuevo instrumento de seguimiento. El proceso no solo implica definir el título y descripción de la encuesta, sino que requiere la configuración de una plantilla de reporte y la carga de documentos base a la nube. La inteligencia del sistema reside en la base de datos, no en el código. Arquitectónicamente, acoplamos cada pregunta abierta con una instrucción de ingeniería de prompts específica (stored prompt).

4.4.5 Proceso de Participación de Graduados

El flujo de respuesta comienza cuando un egresado accede a su panel y visualiza las encuestas activas según su carrera. El sistema realiza una validación de integridad para asegurar que el usuario tenga el rol de "graduado" y no haya completado previamente el formulario. Al enviar las respuestas, se inicia una transacción que registra cada entrada en la tabla Answers, vinculándolas de forma única al identificador del graduado y a la pregunta correspondiente. Una vez finalizado el proceso, el sistema actualiza el estado de participación del usuario y habilita de forma reactiva la visualización de estadísticas generales, confirmando la recepción exitosa de la data para su posterior procesamiento.

4.4.6 Proceso de Generación de Reportes con IA

El motor de síntesis automatizada no opera por arte de magia; sigue un pipeline de datos estricto activado a demanda. Cuando un administrador solicita un reporte, el backend en Node.js actúa como despachador: empaqueta las respuestas crudas de la base de datos y las transfiere, mediante una petición interna segura, al microservicio de Python.

La lógica de síntesis reside en este script, que opera mucho más allá de una simple recepción de texto. Su función es orquestar una inyección dinámica de contexto: selecciona el

prompt técnico adecuado para la carrera y alimenta al motor Ollama. Lo valioso no es solo la inferencia, sino la salida estructurada. El sistema manipula directamente el árbol XML de la plantilla .docx en la nube para incrustar la narrativa generada quirúrgicamente. Así, obtenemos un reporte híbrido que fusiona la frialdad de los gráficos estadísticos programáticos con una redacción natural coherente, erradicando el error manual en la fase de auditoría.

4.5 Interfaces y Capturas del Sistema

continuación, se presenta la evidencia visual de la aplicación web desarrollada, demostrando el cumplimiento de los requerimientos funcionales y de experiencia de usuario exigidos para el proyecto

4.5.1 Interfaces del Módulo de Bolsa de Empleo

Figura 6

Dashboard informativo del sistema

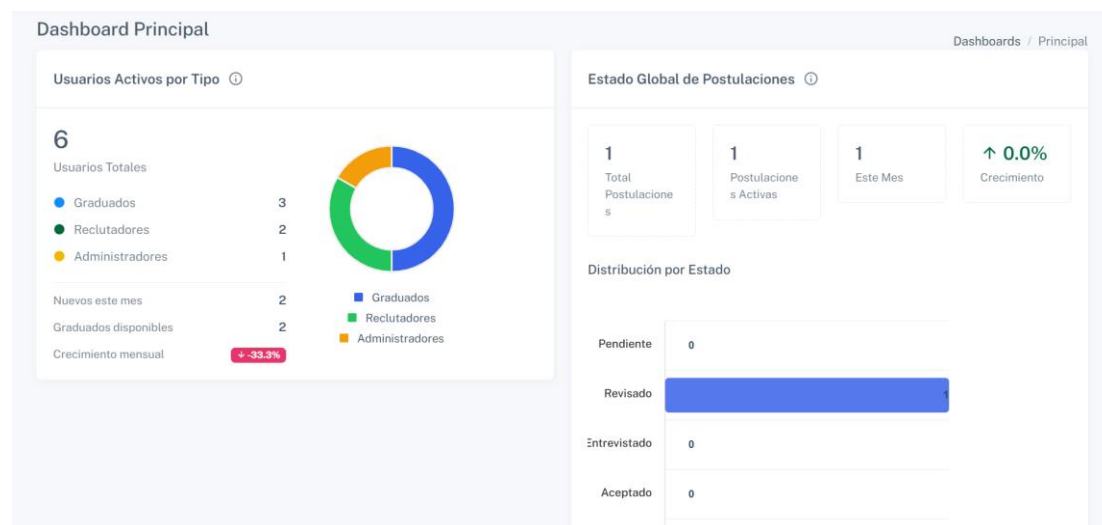


Figura 7

Módulo de directorio de graduados

The screenshot displays the 'Directorio de Graduados' (Graduates Directory) module in the Ulearn system. The left sidebar contains the main menu with options like Dashboard, Ofertas de Empleo, Mis Favoritos, Crear Ofertas, Postulaciones, Empresas, Asignar Reclutadores, and the selected 'Directorio de Graduados'. The main content area shows a search bar with filters for 'Todos' (All) and 'Todas' (All), and a dropdown for 'Selecciona facultad'. Below the search bar, it indicates '6 resultados' (6 results). Three user profiles are displayed:

- JORDAN ARMANDO MORAN Z...**: Tecnología en Informática, Facultad de Ciencias Informáticas. Contact: jordan@gmail.com, 0994312472. Status: Disponible.
- FABRICIO JOSUE ZAVALA REYES**: Sin carrera registrada, Facultad de Ciencias Informáticas. Contact: fabriciozavala13@gmail.com. Status: Disponible.
- jonathan marin**: test. Contact: jony@gmail.com, 1231231231. Status: Disponible.

Each profile includes a 'Ver Perfil' (View Profile) button and a chat icon.

Figura 8

Módulo de ofertas laborales

The screenshot displays the 'Ofertas de Empleo' (Job Offers) module in the Ulearn system. The left sidebar is similar to Figure 7, with 'Ofertas de Empleo' selected. The main content area shows a search bar with filters for 'Tipo de Empleo' (Job Type) and 'Modalidad' (Modality), both set to 'Todos' (All). Below the search bar, there are filters for 'Estado' (Status) set to 'Todos', 'Ubicación' (Location) set to 'Ciudad o país...', and 'Salario' (Salary) with 'Mínimo' and 'Máximo' fields. A 'Aplicar Filtros' (Apply Filters) button and a 'Limpiar' (Clear) button are present. Below the filters, two job listings are shown:

- Analista ...**: Sin empresa, PUBLICADA. Tipo de Empleo: Tiempo Completo. Modalidad: Presencial. Ubicación: manta. Salario: \$800 - \$3,000. Descripción: necesitamos una persona que se encargue del manejo de: powerbi bases de datos... Cierre: 27 Feb. Ver detalles.
- test**: Sin empresa, PUBLICADA. Tipo de Empleo: Tiempo Completo. Modalidad: Híbrido. Ubicación: Manta, Manabí. Salario: \$111 - \$111,111. Descripción: tttttttttt. Cierre: 26 Jan. Ver detalles.

Figura 9

Módulo de mensajerías con web sockets

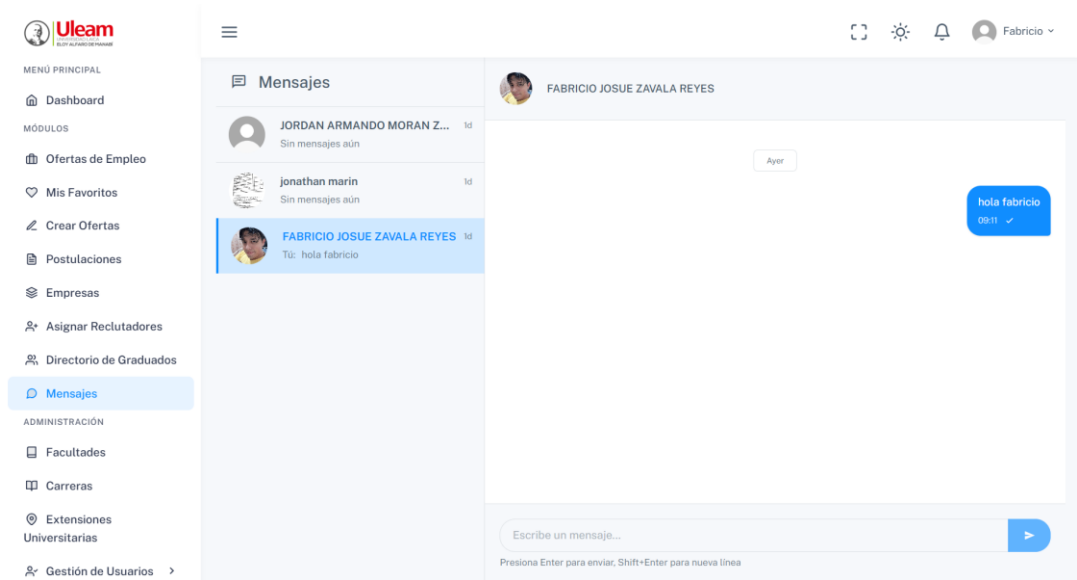
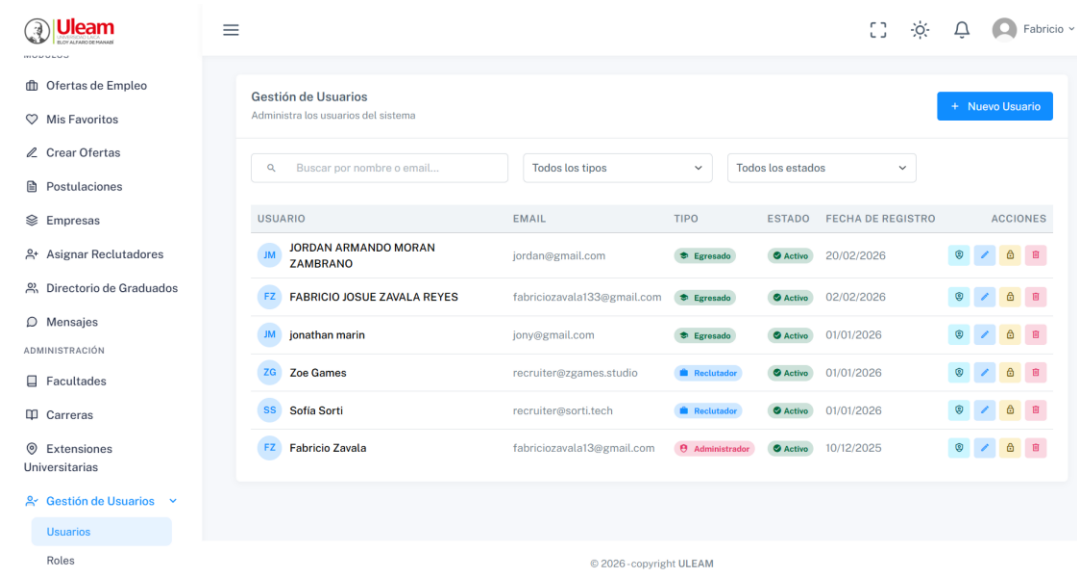


Figura 10

Módulo para gestionar usuario del sistema



4.5.2 Interfaces del Módulo de Seguimiento a Graduados

Figura 11

Módulo de dashboard de bienvenida

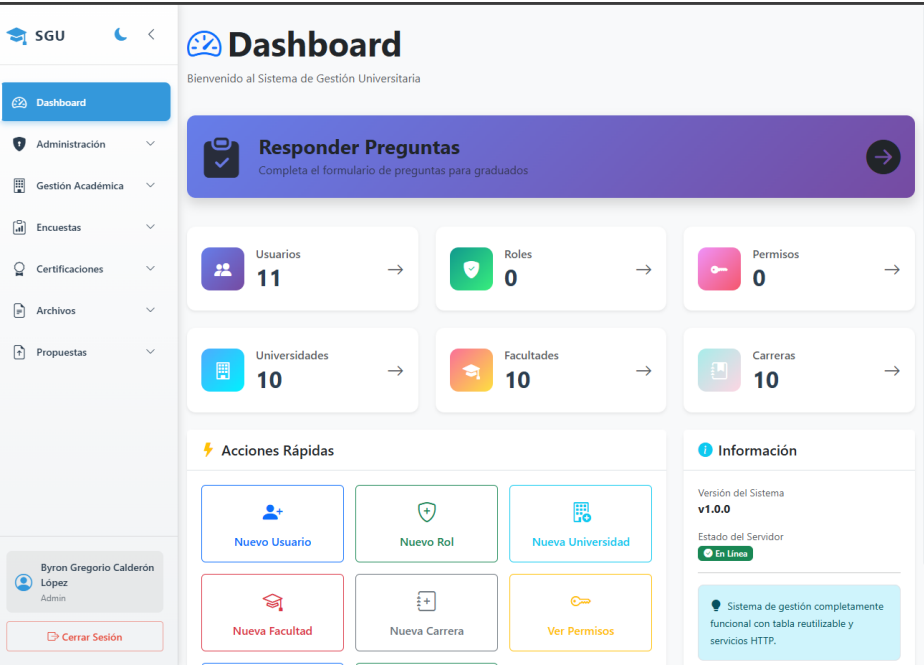


Figura 12

Módulo de Gestión de encuesta

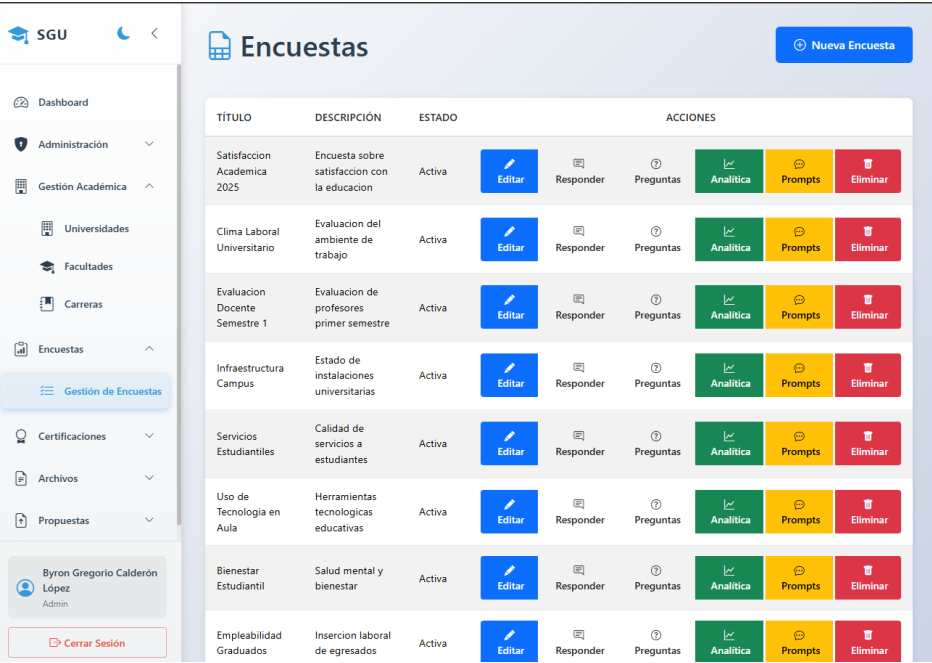
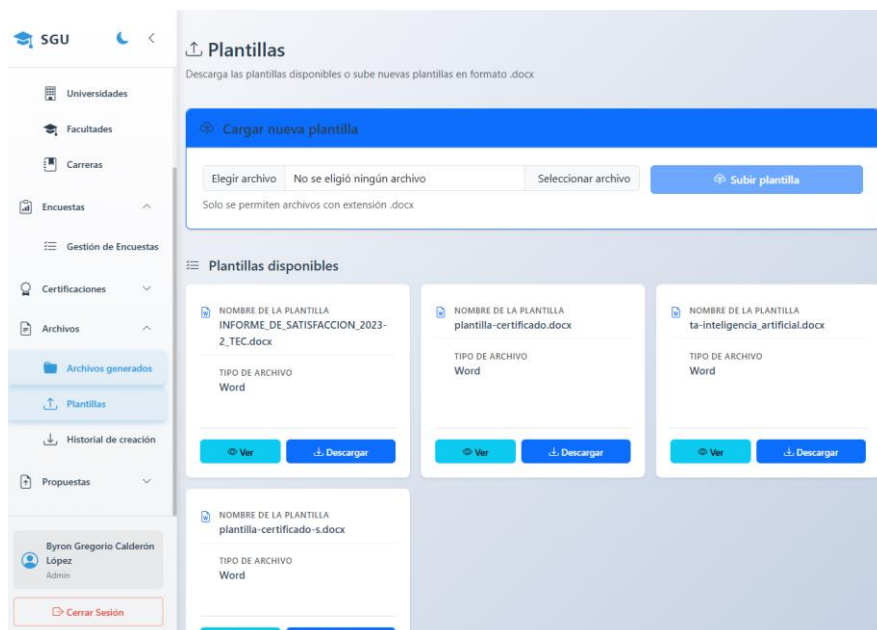
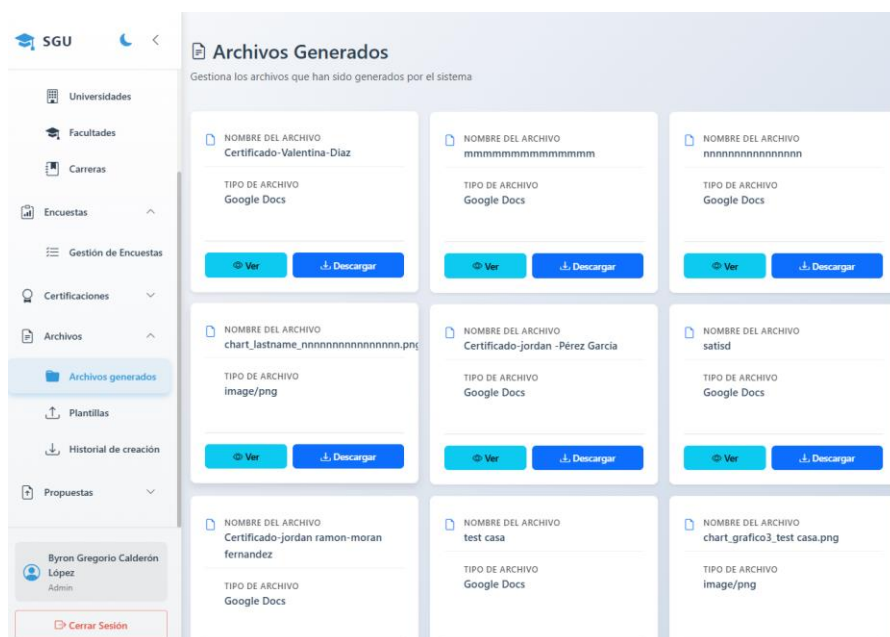


Figura 13*Módulo de Gestión de plantillas***Figura 14***Módulo de Gestión de documentos generados***Figura 15***Módulo de Gestión de certificaciones*

SGU

Facultades

Carreras

Encuestas

Gestión de Encuestas

Certificaciones

Gestión de Certificación

Archivos

Archivos generados

Plantillas

Historial de creación

Propuestas

Mis Propuestas

Byron Gregorio Calderón López

Admin

Cerrar Sesión

Certificaciones

Nueva Certificación

Las certificaciones permiten generar documentos automáticamente cuando un estudiante completa una encuesta. La plantilla debe contener los marcadores (**firstname**), (**lastname**) y (**identification**) para reemplazar los datos del estudiante.

NOMBRE	DESCRIPCION	ENCUESTA	ESTADO	ACCIONES
Certificado Clima Laboral	Certificado por participar en encuesta de clima laboral	Clima Laboral Universitario	Activa	Editar Eliminar
Certificado General	Certificado sin encuesta asociada	Sin encuesta	Activa	Editar Eliminar
Certificado Satisfacción Académica	Certificado por completar encuesta de satisfacción	Satisfacción Académica 2025	Activa	Editar Eliminar

5 Capítulo V

5.1 Conclusiones

En relación con los propósitos fundamentales de esta investigación, el análisis exhaustivo de los requerimientos funcionales y no funcionales permitió identificar las deficiencias críticas del sistema legacy basado en Laravel. A partir de este diagnóstico empírico y documental, se diseñó con éxito una arquitectura de software completamente desacoplada, lo cual sentó las bases lógicas y operativas para superar la rigidez estructural que impedía la escalabilidad de los procesos de seguimiento a graduados y bolsa de empleo en la institución.

Como resultado directo de este diseño, se logró estructurar e implementar una arquitectura de proyecto modular, compuesta por múltiples backends especializados que se comunican con un frontend unificado. Esta infraestructura fue orquestada eficientemente mediante el uso de contenedores de Docker, lo que garantizó un ecosistema elástico donde la tecnología Node.js asume el control de la concurrencia; esto liberó los procesos de bloqueos y permitió que la plataforma soporte picos masivos de tráfico transaccional sin comprometer la estabilidad del servidor.

Para materializar la interacción del usuario con esta robusta infraestructura distribuida, se desarrolló una aplicación web moderna empleando el framework Angular 19, enfocada en la gestión integral de encuestas y módulos de empleabilidad. La implementación técnica a través de Standalone Components erradicó la deuda técnica de las plataformas tradicionales, consolidando una Single Page Application (SPA) con una navegación casi instantánea que optimiza significativamente la experiencia de uso tanto para los egresados postulantes como para el personal administrativo.

Complementando la recolección ágil de información, el desafío del análisis de datos se resolvió mediante la programación de APIs especializadas en un entorno de Python. Estas interfaces se encargaron de integrar de forma local el modelo de Inteligencia Artificial Llama 3.2, demostrando una alta capacidad técnica para procesar, decodificar e interpretar miles de respuestas cualitativas derivadas de las encuestas; todo ello ejecutado dentro de un entorno informático controlado que garantiza la soberanía y privacidad absoluta de la información institucional.

Finalmente, este ecosistema de procesamiento inteligente culminó con éxito al consolidar la gestión automatizada de la generación documental. El sistema logró orquestar la extracción de los datos analizados y los gráficos estadísticos para incrustarlos directamente en plantillas de documentos estructurados (.docx) almacenados de forma segura en la nube. Esta automatización erradicó por completo la necesidad de manipulación manual de la información, transformando la elaboración de informes técnicos que históricamente representaba semanas de trabajo administrativo en un proceso altamente eficiente de escasos minutos.

5.2 Recomendaciones

Transición a Orquestación de Contenedores (Kubernetes): Aunque Docker Compose sostiene eficazmente la fase actual, la operación a gran escala exige dar el salto hacia orquestadores robustos como Kubernetes o AWS ECS. No es solo una mejora incremental; es un requisito de estabilidad. Al implementar esta tecnología, la infraestructura gana la capacidad de "auto-escalado horizontal". ¿Qué significa esto en la práctica? Que cuando ocurra una feria laboral virtual y el tráfico se dispare de 100 a 10.000 usuarios en minutos, el sistema creará automáticamente nuevas réplicas del backend en Node.js para absorber la carga sin intervención humana. El objetivo es garantizar una disponibilidad del 99.9%, eliminando el riesgo de caídas del servidor justo cuando los graduados más lo necesitan.

Incorporación de Aceleración por Hardware (GPU): Un punto a mejorar es el rendimiento físico del servidor el cual se limita únicamente a procesamiento por CPU por lo cual una mejora a futuro viable sería la integración de una tarjeta grafica en el servidor de esta manera la velocidad de respuesta acelerará enormemente y por consecuencia la latencia será menor en picos altos de peticiones.

Capacitación Continua en Ingeniería de Prompts: Todas las instrucciones son definidas y no se realizan acciones a la ligera debido a un estricto sistema de procesamiento de pasos, es decir, que la calidad de las respuestas en el análisis dependerá directamente de la calidad de las indicaciones otorgadas. Por este motivo se debe capacitar al personal que usará el sistema para que puedan insertar instrucciones (prompts) detalladas y específicas de sus necesidades. Si los docentes aprenden a iterar y refinar los prompts, podrán adaptar el análisis de la IA a la realidad específica de cada carrera sin depender del departamento de TI para cada ajuste menor.

5.3 REFERENCIAS BIBLIOGRÁFICAS

1. Andrade, J. (2021). *From a Monolith to Microservices: A migration strategy for legacy systems*. (Tesis de Maestría). Universidad de los Andes.
2. Arteaga-Briones, L., Azúa-Arteaga, A., & López-Padrón, A. (2023). Modelo para seguimiento de graduados: hacia el aseguramiento de la calidad de la Educación Superior ecuatoriana. *Revista de Ciencias Sociales*, 29(3), 264-279.
3. Asamblea Constituyente. (2008). *Constitución de la República del Ecuador*. Registro Oficial 449.
4. Asamblea Nacional del Ecuador. (2010). *Ley Orgánica de Educación Superior (LOES)*. Registro Oficial Suplemento 298.
5. Asamblea Nacional del Ecuador. (2021). *Ley Orgánica de Protección de Datos Personales*. Registro Oficial Suplemento 459.
6. Becker, G. (1964). *Human Capital: A Theoretical and Empirical Analysis*. University of Chicago Press.
7. Cevallos, M. (2018). *Diseño de un sistema de seguimiento a graduados para el IAEN a través de aplicaciones web 2.0*. (Tesis de Pregrado). Pontificia Universidad Católica del Ecuador.
8. Consejo de Educación Superior (CES). (2023). *Reglamento de Régimen Académico*. Resolución RPC-SO-01-No.064-2023.
9. García, A., & Martínez, L. (2021). Implementación de arquitecturas basadas en microservicios para la gestión universitaria en Latinoamérica. *Revista Iberoamericana de Tecnología en Educación*, 14(2), 45-60.
10. Guevara Albán, C. (2019). *Sistema de información para los procesos de seguimiento a graduados e inserción laboral del Instituto Tecnológico Superior Babahoyo*. (Tesis de Maestría). Universidad Regional Autónoma de los Andes.
11. Hillage, J., & Pollard, E. (1998). *Employability: developing a framework for policy analysis*. Department for Education and Employment.
12. Islam, M. (2023). *Comparative Analysis of Monolithic vs. Microservices Architecture for E-commerce Applications*. Journal of Software Engineering and Applications.
13. Latona, T., & Browne, M. (2021). Postgraduate Tracking System: Student Research Progress Tracking Tool. *International Journal of Education and Development using ICT*.
14. Mera, J., & Zambrano, K. (2022). La usabilidad en portales de empleo universitarios: Un estudio de caso en Manabí. *Revista San Gregorio*, 1(48), 112-125.
15. Meta AI. (2024). *Llama 3 Model Card*. Recuperado de <https://github.com/meta-llama/llama3>
16. Ministerio de Educación del Ecuador. (2024). *Policy Brief con los resultados de la toma inicial de la encuesta de seguimiento a bachilleres*.
17. Pérez, D., & Villavicencio, R. (2020). *Plataforma de vinculación laboral con análisis predictivo para egresados de ingenierías*. (Tesis de Grado). Escuela Politécnica Nacional.

18. Smith, R., & Johnson, T. (2023). Privacy-Preserving Large Language Models for Educational Data Mining. *2023 IEEE International Conference on Advanced Learning Technologies (ICALT)*.
19. Toapanta, L. (2023). *Diseño e implementación de aplicaciones basadas en microservicios utilizando un modelo de optimización de dependencia y complejidad*. (Tesis de Grado). Universidad Técnica de Cotopaxi.
20. UNESCO. (2023). *Seguimiento de los egresados de la educación superior: Guía metodológica*. Ediciones UNESCO.
21. Genesis, B. C. (2025). Implementación de un aplicativo para el proceso de seguimiento de graduados de TI.
22. Universidad Laica Eloy Alfaro de Manabí. (s.f). Manual de usuario de la plataforma web de bolsa de trabajo. Manta: ULEAM.

6 Anexos

6.1 Enlace al sistema de seguimiento a graduados:

[Manual de usuario de seguimiento a graduados.docx](#)

6.2 Enlace al sistema de bolsa de empleos:

[Manual de usuario de Bolsa de empleos.docx](#)