

UNIVERSIDAD AMAZÓNICA DE PANDO

ÁREA DE CIENCIAS Y TECNOLOGÍA

CARRERA DE INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

**IMPLEMENTACIÓN DE MÓDULOS DE AUTOMATIZACIÓN Y VINCULACIÓN
LABORAL EN EL SISTEMA DE SEGUIMIENTO A TITULADOS DE LA
UNIVERSIDAD AMAZÓNICA DE PANDO**

POSTULANTE: Univ. Jesus Israel Condori Choquehuanca

TUTOR: Ing. Henry Ivan Surco Alvan

Cobija - Pando – Bolivia

2025

AGRADECIMIENTOS

En primer lugar, a Dios, por permitirme coincidir con personas de inmensa calidad humana y profesional a lo largo de este camino.

A la memoria de mi madre, quien desde el cielo ha sido mi guía constante y me ha acompañado espiritualmente en cada etapa. A mi padre, por brindarme su apoyo absoluto y fortaleza, superando cualquier barrera de distancia.

A mi novia, Mariel, por su paciencia y por ser un soporte fundamental durante la elaboración de este proyecto. A mi amiga y hermana, Carmen, por estar presente en todo momento y brindarme su apoyo incondicional sin importar las circunstancias.

Expreso mi sincero reconocimiento a mi tutor, Ing. Henry Surco, no solo por su dirección académica, sino por impulsarme a persistir en los momentos difíciles y motivarme a alcanzar esta meta.

Finalmente, agradezco a mis colegas, Ing. Dixon G. Villca Ch. e Ing. Yosel Justiniano S. por su valiosa colaboración y guía técnica durante el desarrollo de esta investigación.

DEDICATORIA

A la memoria de mi madre, mi ángel eterno, quien desde el cielo guía mis pasos y me enseñó que no existen imposibles. Este logro es tuyo.

A mi padre, por ser mi ejemplo de fortaleza y perseverancia. Gracias por creer en mí siempre.

RESUMEN

El presente Proyecto de Grado aborda la optimización del proceso de seguimiento a titulados de la Carrera de Ingeniería de Sistemas de la Universidad Amazónica de Pando (UAP) mediante la implementación de módulos de automatización y vinculación laboral. A pesar de contar con un sistema previo, la gestión de la información presentaba limitaciones operativas debido a la dependencia de procesos manuales para la recolección de datos y la falta de integración con oportunidades de empleo, lo que dificultaba el análisis de la inserción laboral y el cumplimiento de requisitos de acreditación.

Para solucionar esta problemática, se desarrolló e implementó una solución tecnológica bajo la metodología ágil Scrum, incorporando dos componentes fundamentales: un generador dinámico de formularios y un módulo de bolsa de trabajo automatizada. El sistema fue construido utilizando una arquitectura modular desacoplada, empleando Spring Boot para el backend y React para el frontend, garantizando escalabilidad y rendimiento. Además, se integraron técnicas de Web Scraping para la extracción automática de ofertas laborales externas, centralizando las oportunidades en una plataforma única para los titulados.

Los resultados obtenidos demuestran una mejora significativa en la eficiencia administrativa, reduciendo el tiempo de tabulación de encuestas a cero y permitiendo la generación de estadísticas en tiempo real para la toma de decisiones académicas. Asimismo, la plataforma fortalece el vínculo entre la universidad y sus profesionales, facilitando su inserción en el mercado laboral y proporcionando evidencias objetivas para los procesos de evaluación y mejora continua de la calidad educativa.

Palabras clave: Seguimiento a Titulados, Automatización de Procesos, Vinculación Laboral, Bolsa de Trabajo, Web Scraping, Generador de Formularios Dinámicos, Metodología Scrum.

ABSTRACT

This Degree Project addresses the optimization of the graduate follow-up process for the Systems Engineering Career at the Universidad Amazónica de Pando (UAP) through the implementation of automation and labor linkage modules. Despite having a previous system, information management presented operational limitations due to the reliance on manual processes for data collection and the lack of integration with job opportunities, which hindered the analysis of job placement and compliance with accreditation requirements.

To address this issue, a technological solution was developed and implemented using the Scrum agile methodology, incorporating two fundamental components: a dynamic form generator and an automated job board module. The system was built using a decoupled modular architecture, employing Spring Boot for the backend and React for the frontend, ensuring scalability and performance. Additionally, Web Scraping techniques were integrated for the automatic extraction of external job offers, centralizing opportunities into a single platform for graduates.

The results obtained demonstrate a significant improvement in administrative efficiency, reducing survey tabulation time to zero and enabling the generation of real-time statistics for academic decision-making. Likewise, the platform strengthens the link between the university and its professionals, facilitating their insertion into the labor market and providing objective evidence for evaluation processes and the continuous improvement of educational quality.

Keywords: Graduate Follow-up, Process Automation, Labor Linkage, Job Board, Web Scraping, Dynamic Form Generator, Scrum Methodology.

ÍNDICE GENERAL

1.1.	INTRODUCCIÓN	1
1.2.	ANTECEDENTES	1
1.3.	ESTADO DEL ARTE	2
1.4.	DESCRIPCIÓN DEL PROBLEMA.....	4
1.5.	FORMULACIÓN DEL PROBLEMA.....	5
1.6.	OBJETIVOS	5
1.6.1.	Objetivo general.....	5
1.6.2.	Objetivos Específicos.....	6
1.7.	METODOLOGÍA.....	6
1.8.	JUSTIFICACIÓN.....	9
1.8.1.	Justificación social	10
1.8.2.	Justificación económica.....	10
1.8.3.	Justificación técnica.....	10
1.9.	ALCANCES	11
2.1.	MARCO INSTITUCIONAL	14
2.1.1.	Universidad Amazónica de Pando	14
2.1.2.	Ingeniería de Sistemas.....	14
2.1.3.	Misión	14
2.1.4.	Visión	14
2.2.	MARCO TEÓRICO.....	15
2.2.1.	Sistemas de Información en Contexto Educativo	15
2.2.2.	Arquitecturas de Aplicaciones Web.....	15
2.2.3.	Gestión de Bases de Datos Relacionales	16
2.2.4.	Metodologías de Desarrollo de Software	16
2.2.5.	Seguimiento a Egresados: Enfoques Teóricos	17
2.3.	MARCO TECNOLÓGICO	17
2.3.1.	Herramientas para el análisis de requerimientos.....	17
2.3.2.	Herramientas para el diseño del sistema	18
2.3.3.	Herramientas para el desarrollo del sistema	20
2.3.4.	Herramientas para la implementación del sistema	26
2.3.5.	Herramientas para la evaluación del sistema	26
2.3.6.	Herramientas para la documentación técnica del sistema.....	27
2.3.7.	Herramientas para estándares de Seguridad Web.....	27

3.1. FASE PRE-GAME	30
3.1.1. Establecimiento del marco Scrum	30
3.1.2. Definición de roles del equipo	30
3.1.3. Construcción del Product Backlog	32
3.1.4 Levantamiento de requerimientos	33
3.1.4.1 Requerimientos funcionales – Módulo Generador Dinámico de Formularios	
34	
3.1.4.2 Requerimientos funcionales – Módulo Sistema de Recolección de Respuestas .	36
3.1.4.3 Requerimientos funcionales – Módulo Vinculación Laboral (Bolsa De Trabajo)	
.....	38
3.1.4.4 Requerimientos no funcionales – Seguridad	41
3.1.4.5 Requerimientos no funcionales – Usabilidad	42
3.1.4.6 Requerimientos no funcionales – Rendimiento	43
3.1.4.7 Requerimientos no funcionales – Modalidad	44
3.1.5. Mapeo de historias de usuario	45
3.1.5.1 Módulo: Generador dinámico de formularios	45
3.1.5.2 Módulo: Recolección de respuestas	46
3.1.5.3 Módulo: Vinculación laboral (bolsa de trabajo)	47
3.1.5.4 Módulo: Estadísticas y Seguimiento a respuestas	48
3.1.5.5 Módulo: Integración externa (scraping)	49
3.1.5.6 Módulo: Gestión de formularios y ciclo de vida	50
3.1.6. Casos de uso del sistema	50
3.1.7. Matriz de Roles y Privilegios (Modelo RBAC)	54
3.2. FASE GAME – Desarrollo Iterativo	54
3.2.1. Planificación General de sprints	55
3.2.2. Desarrollo e Implementación de Módulos	56
3.2.2.1. Arquitectura del Sistema Implementada	59
3.2.2.2. Modelo de Datos	66
3.2.2.3. Diagramas UML del Sistema	67
3.2.3. Interfaces Desarrolladas	73
3.2.3.1 Vistas Del Sistema De Graduados UAP	73
3.2.3.2. Vistas Del Sistema Perlaplus	76
3.3. FASE POST-GAME	80
3.3.1. Resultados del Objetivo Específico 1: Análisis de Requerimientos (Fase Pre-Game)	80

3.3.2. Resultado del Objetivo Específico 2: Diseño de la Arquitectura Modular	82
3.3.3. Resultado del Objetivo Específico 2: Diseño de la Arquitectura Modular	83
3.3.4. Resultados del Objetivo Específico 3 (Automatización de Formularios).....	85
3.3.5. Resultado del Objetivo Específico 4: Servicios de Integración Externa (Bolsa de Trabajo).....	87
3.3.6. Resultados del Objetivo Específico 5: Evaluación de Efectividad (Pruebas de Calidad)	88
3.4 DISCUSIÓN DE RESULTADOS	90
3.4.1. Análisis de la Automatización de Procesos (Sobre el Objetivo 3)	90
3.4.2. Análisis de la Vinculación Laboral Automática (Sobre el Objetivo 4).....	90
3.4.3. Análisis de Escalabilidad y Calidad del Software (Sobre el Objetivo 5)	91
3.4.4. Análisis Crítico de la Metodología Scrum (Desafíos y Soluciones)	92
3.4.5. Contribución al Área de Ingeniería de Sistemas y Gestión Universitaria	93
3.4.6. Arquitectura de Despliegue (Entorno de Producción).....	94
3.4.7. Proceso de Implantación	95
3.4.8. Estrategia de Transferencia Tecnológica (Capacitación)	96
4.1. CONCLUSIONES Y RECOMENDACIONES	99
4.1.1. Conclusiones.....	99
4.1.2. Recomendaciones.....	101
Bibliografía.....	104
<i>ANEXO 1: Árbol de Problemas.....</i>	<i>106</i>
<i>ANEXO 2: Árbol de Objetivos.....</i>	<i>107</i>
<i>ANEXO 3: Organigrama de la carrera de Ingeniería de Sistemas.....</i>	<i>108</i>
<i>ANEXO 4: Cronograma de actividades</i>	<i>109</i>
<i>ANEXO 5: Preparación de instrumentos de recolección</i>	<i>110</i>
<i>ANEXO 6: Preparación de instrumentos de recolección</i>	<i>111</i>
<i>ANEXO 7: Capacitación de manejo del Sistema de Graduados UAP a Titulados de la carrera de Ingeniería de Sistemas.....</i>	<i>111</i>
<i>ANEXO 8: Capacitación de manejo del Sistema de Graduados UAP a Titulados de la carrera de Ingeniería de Sistemas – Llenado de Formulario</i>	<i>112</i>
<i>ANEXO 9: Capacitación de manejo del Sistema de Graduados UAP a Titulados de la carrera de Ingeniería de Sistemas – Conociendo la vista de Bolsa de Trabajo</i>	<i>112</i>
<i>ANEXO 10: Informe de Seguimiento a Graduados de la carrera de Ing. De Sistemas - UAP 2024</i>	<i>113</i>
<i>ANEXO 11: INFORME DE SOCIALIZACIÓN Y CAPACITACIÓN (GRUPO PILOTO)</i>	<i>113</i>

ÍNDICE DE FIGURAS

Figura 1 Caso De Uso: Módulo Formulario.....	52
Figura 2 Caso de Uso - Módulo Respuestas.....	53
Figura 3 Caso de Uso-Módulo Vinculación Laboral.....	53
Figura 4 Arquitectura General Implementada.....	60
Figura 5 Modelo de Datos.....	66
Figura 6 Diagrama de Clases del Backend.....	67
Figura 7 Diagrama de Secuencia - Creación de Formulario.....	68
Figura 8 Diagrama de Secuencia - Publicación de encuesta y Registro de respuestas.....	68
Figura 9 Diagrama de Secuencia - Procesos de scraping de Oferta Laboral a las dos fuentes.....	69
Figura 10 Diagrama de Secuencia - Generación automática de estadísticas.....	69
Figura 11 Diagrama de Componentes General del Sistema.....	70
Figura 12 Diagrama de componentes del Backend (Spring Boot).....	70
Figura 13 Diagrama de componentes del FrontEnd (React).....	71
Figura 14 Diagrama de componentes de scraping.....	71
Figura 15 Diagrama de Despliegue/Implementación.....	72
Figura 16 Vista principal para acceder a la bolsa de trabajo.....	73
Figura 17 Vista principal para acceder a la Encuesta.....	73
Figura 18 Vista de primera pregunta de la encuesta (Formulario 1).....	74
Figura 19 Vista de Preguntas de Tipo Cerrada.....	74
Figura 20 Vista de Adición de Pregunta de Otro.....	75
Figura 21 Vista de Pregunta de Tipo Selección múltiple.....	75
Figura 22 Vista de Finalización de Llenado de encuesta.....	76
Figura 23 Vista de Programación de la encuesta, y visualización de periodo habilitación de la encuesta.....	76
Figura 24 Vista de Programación de Publicación de la Encuesta.....	77
Figura 25 Vista de Estadísticas generadas por Sistema.....	77
Figura 26 Vista de Generación de estadística de respuestas de titulados.....	78
Figura 27 Vista de Creación de Preguntas y Respuestas en el Formulario.....	78
Figura 28 Vista de listado general de estudiantes de la carrera.....	79
Figura 29 Vista de Seguimiento de Estado de Respuesta de Titulados.....	79

ÍNDICE DE TABLAS

TABLA 1 Metodología.....	7
TABLA 2 Priorización inicial del Product Backlog	32
TABLA 3 Requerimiento Funcional: Creación de formulario de encuesta dinámico.....	34
TABLA 4 Requerimiento Funcional: Modificación de formulario de encuesta	35
TABLA 5 Requerimiento Funcional: Publicación de formulario para aplicación	36
TABLA 6 Requerimiento Funcional - Completar formulario de encuesta.....	37
TABLA 7 Requerimiento Funcional - Consulta de respuestas de formularios	37
TABLA 8 Requerimiento Funcional - Importación automatizada de ofertas de empleo mediante web scraping.....	38
TABLA 9 Requerimiento Funcional - Búsqueda y filtrado de ofertas de empleo	39
TABLA 10 Requerimiento Funcional - Aplicación a oferta de empleo	39
TABLA 11 Importar ofertas laborales desde fuentes externas (web scraping)	40
TABLA 12 Requerimientos no funcionales – Seguridad	41
TABLA 13 Requerimientos no funcionales – Usabilidad	42
TABLA 14 Requerimientos no funcionales – Rendimiento	43
TABLA 15 Requerimientos no funcionales - Modalidad.....	44
TABLA 16 Historia de Usuario - Creación de encuesta.....	45
TABLA 17 Historia de Usuario - Edición de Formularios	46
TABLA 18 Historia de Usuario - Publicación y Programación de encuesta.....	46
TABLA 19 Historia de Usuario - Recolección de Información de Titulado	47
TABLA 20 Historia de usuario - Búsqueda y filtrado de ofertas laborales	47
TABLA 21 Historia de Usuario - Postulación a oferta laboral.....	48
TABLA 22 Historia de Usuario - Visualización y Seguimiento de estado de Respuestas de Titulados	48
TABLA 23 Historia de Usuario - Importación de ofertas mediante Scraping Automatizado	49
TABLA 24 Historia de Usuario - Gestión de Encuesta Publicada	50
TABLA 25 Matriz de Roles y Privilegios (Modelo RBAC)	54
TABLA 26 Planificación General de Sprints	55
TABLA 27 Capas de componentes claves desarrollados-Generador Dinámico de Formularios .	56
TABLA 28 Capas de componentes desarrollados- Sistema de Recolección de Respuestas	57
TABLA 29 Capas de Componentes desarrollados - Estadística y tablero de control.....	57
TABLA 30 Capas de componentes desarrollados - Vinculación Laboral	58
TABLA 31 Componentes según la tecnología desarrollada- Motor Web Scraping	58
TABLA 32 Arquitectura del Sistema Implementada.....	59
TABLA 33 Componentes necesarios para el Proyecto en el Front End (React)	61
TABLA 34 Arquitectura de Software Modular en React	62
TABLA 35 Componentes y servicios del frontend.....	63
TABLA 36 Arquitectura Backend (SpringBoot)	64
TABLA 37 Estructura de Paquetes y Responsabilidades del Backend	65
TABLA 38 Matriz de Resultados del Análisis de Necesidades (Fuente: Elaboración propia en base al Estudio de Seguimiento a Graduados 2024)	81
TABLA 39 Resultados de las Decisiones de Arquitectura	83
TABLA 40 Capacidades del Módulo de Automatización	84
TABLA 41 Matriz Comparativa de Tiempos y Movimientos (Manual vs. Automatizado).....	85

TABLA 42 Métricas de Rendimiento del Motor de Extracción (Scraping)	87
TABLA 43 Resultados de Pruebas de Carga y Escalabilidad (Fuente: Elaboración propia en base a simulación con JMeter).....	89
TABLA 44 Matriz de Desafíos Metodológicos y Soluciones Ágiles	92
TABLA 45 Especificaciones del Entorno de Producción.....	94

CAPITULO I:
MARCO INTODUCTORIO

1.1. INTRODUCCIÓN

El seguimiento a titulados es un proceso fundamental para las universidades, ya que permite evaluar la calidad educativa, la pertinencia de la formación académica y la inserción laboral de los egresados. Este proceso facilita la toma de decisiones informadas sobre la mejora continua de los planes de estudio y contribuye al fortalecimiento de la vinculación entre la institución y sus egresados. En la Carrera de Ingeniería de Sistemas de la Universidad Amazónica de Pando (UAP), aunque existe un sistema de seguimiento a titulados, este aún presenta limitaciones operativas que dificultan una recolección eficiente de datos y la interacción fluida con los egresados.

El presente Proyecto de Grado aborda la implementación de módulos adicionales que fortalezcan el sistema existente, específicamente un generador dinámico de formularios y una bolsa de trabajo. Los formularios de recolección son una herramienta esencial en el proceso de acreditación, ya que permiten consolidar información valiosa sobre los egresados y ajustarla a los indicadores específicos requeridos por la currícula de Ingeniería de Sistemas. De esta manera, no solo se optimiza la gestión de datos, sino que también se garantiza el cumplimiento de los estándares de calidad para la acreditación.

La metodología ágil Scrum será utilizada en el desarrollo de este proyecto, permitiendo realizar ajustes constantes a medida que se obtiene retroalimentación de los usuarios clave. Con esta implementación, se espera mejorar el seguimiento a los titulados, facilitar la inserción laboral a través de la bolsa de trabajo y contribuir a la mejora continua de los planes de estudio, fortaleciendo así el vínculo entre la universidad y su comunidad profesional.

1.2. ANTECEDENTES

La Universidad Amazónica de Pando, creada en 1984 mediante Decreto Supremo N° 20511, es una institución autónoma del Sistema de la Universidad Boliviana (SUB), con la misión de formar profesionales íntegros con excelencia académica y compromiso social en el contexto amazónico. La carrera de Ingeniería de Sistemas, establecida en 2009, se orienta a formar expertos en sistemas de información, modelamiento, optimización y gestión empresarial, con el fin de contribuir al desarrollo tecnológico sostenible de la región y del país.

El Sistema de Seguimiento a Titulados, desarrollado por la Dirección de Información Académica, surge como una herramienta para evaluar el impacto de la formación académica y mantener contacto con los egresados. Sin embargo, enfrenta limitaciones técnicas que impiden su plena implementación. Aunque está disponible en todas las carreras de la universidad, y no se ha implementado completamente en cada una, y depende de procesos manuales, lo que retrasa la actualización de los datos y dificulta el seguimiento adecuado de los titulados.

Actualmente la recolección de datos se realiza manualmente mediante formularios en Word y Excel, donde la responsable debe contactar individualmente a cada titulado para completar los formularios, lo que genera demoras. La gestión de la información también es manual, lo que impide un análisis eficiente de la situación laboral de los egresados. Además, la falta de integración con una bolsa de trabajo limita la conexión de los titulados con oportunidades laborales y dificulta el seguimiento de su empleabilidad. (Ver Anexos 5 y 6)

1.3. ESTADO DEL ARTE

Los sistemas de información web se han consolidado como una herramienta fundamental en múltiples ámbitos, al permitir la gestión eficiente y el acceso remoto a datos en tiempo real. En el ámbito académico, han surgido sistemas de seguimiento de titulados que mejoran la recolección y el análisis de información sobre egresados, lo que facilita la toma de decisiones estratégicas y el fortalecimiento de los programas educativos. A continuación, se presenta una revisión de trabajos de grado relevantes que han sido desarrollados o implementados en esta área.

(CCANTO BALVIN, 2023) *Implementación de un Sistema Informático de Seguimiento a Egresados y la Mejora de su Gestión en la Universidad Peruana los Andes* (Universidad Peruana Los Andes)

- Objeto de Estudio: Analizar y mejorar la gestión de seguimiento a egresados en la Universidad Peruana Los Andes mediante un sistema informático que optimice el manejo de su información y fortalezca el vínculo con la institución.
- Objetivo General: Mejorar la gestión de seguimiento a egresados, mediante la implementación de un Sistema Informático de Seguimiento a Egresados en la Universidad Peruana Los Andes.

- Conclusiones: La implementación del sistema informático permitió mejorar significativamente la gestión de seguimiento a egresados, optimizando la recolección y análisis de datos, fortaleciendo la relación universidad-egresados y promoviendo una toma de decisiones más informada.

(MORALES CAMACHO, 2023) *Diseño e Implementación de un Módulo de Seguimiento a Graduandos y Egresados del Programa Talento Magdalena.* (Universidad del Magdalena)

- Objeto de Estudio: Diseñar e implementar un módulo web para gestionar y dar seguimiento a graduandos y egresados del programa Talento Magdalena, optimizando el manejo de su información académica y laboral.
- Objetivo General: Diseñar e implementar un módulo de seguimiento a graduandos y egresados del programa Talento Magdalena.
- Conclusiones: El módulo de seguimiento fue implementado exitosamente, mejorando la gestión de la información en la oficina de Desarrollo Estudiantil. Se diseñó bajo arquitectura MVC utilizando PHP y Laravel, e integró una base de datos local debido a fallas en los servicios del sistema anterior. La solución optimizó los procesos, redujo errores, y demostró eficiencia en pruebas. Se recomienda evaluarlo semestralmente para adaptarse a cambios en plataformas institucionales.

(POMA RODRIGUEZ H. E., 2021) *Sistema de Encuesta Web para el Seguimiento a los Titulados de la Universidad Mayor de San Andrés.* (Universidad Mayor de San Andrés)

- Objeto de Estudio: Desarrollar e implementar un sistema web de encuestas dirigido a los titulados de la Universidad Mayor de San Andrés (UMSA), con el fin de recolectar y procesar información sobre su situación profesional, académica y laboral posterior a su graduación.
- Objetivo General: Desarrollar e implementar un sistema web de encuestas para los titulados de la UMSA que permita almacenar la información recopilada y generar cuadros evaluativos para análisis institucional.
- Conclusiones: El módulo de seguimiento fue implementado con éxito, mejorando la gestión de la información en la oficina de Desarrollo Estudiantil. El sistema fue diseñado bajo arquitectura MVC utilizando PHP y Laravel e integró una base de datos local debido a fallas en los servicios

del sistema anterior. La solución optimizó los procesos, redujo los errores y demostró eficiencia en pruebas. Se recomienda evaluar semestralmente para adaptarse a cambios en las plataformas institucionales.

1.4. DESCRIPCIÓN DEL PROBLEMA

El sistema de seguimiento a titulados implementado en la Universidad Amazónica de Pando presenta limitaciones funcionales que restringen su eficacia operativa. Actualmente carece de módulos para la recolección automatizada de datos mediante encuestas adaptadas al currículo de cada carrera, lo que obliga a mantener procesos manuales de contacto individual con los egresados. Asimismo, no cuenta con una plataforma de bolsa de trabajo integrada que facilite la vinculación de los titulados con oportunidades laborales del mercado regional y nacional.

Las principales causas que originan esta problemática son:

- Ausencia de un generador dinámico de formularios. El sistema no dispone de herramientas para crear y personalizar encuestas según las necesidades específicas de cada carrera, lo que impide adaptar los instrumentos de recolección de datos a las particularidades curriculares de cada programa académico.
- Dependencia de procesos manuales. La recopilación de información sobre la situación laboral de los titulados requiere contacto individual mediante llamadas telefónicas y mensajes de WhatsApp, registrando manualmente los datos en hojas de Excel. Este proceso es prolongado y poco eficiente, requiriendo múltiples intentos de contacto antes de obtener una respuesta del titulado.
- Falta de integración con plataformas de empleo. El sistema opera de manera aislada, sin conexión con bolsas de trabajo institucionales o plataformas externas de reclutamiento, limitando las oportunidades de empleabilidad de los egresados y dificultando el seguimiento de su inserción laboral.

Las limitaciones descritas generan los siguientes efectos:

- Gestión ineficiente de datos. La dependencia de procesos manuales genera una base de datos desactualizada e incompleta. Durante el periodo académico II/2024, solo 19 de los 90 titulados (21%) de la Carrera de Ingeniería de Sistemas respondieron las encuestas de seguimiento, evidenciando la baja efectividad del método actual.

- Análisis limitado de empleabilidad. La ausencia de datos sistematizados impide realizar análisis estadísticos sobre la situación laboral de los egresados, identificar tendencias del mercado laboral o evaluar la pertinencia de los programas académicos con las demandas del sector productivo.
- Debilitamiento del vínculo universidad-egresado. La falta de una plataforma que facilite la interacción continua entre la universidad y sus titulados reduce las oportunidades de actualización profesional, servicios de orientación laboral y retroalimentación para la mejora curricular.
- Dificultad para la acreditación. Los organismos de acreditación universitaria exigen evidencias sobre el seguimiento sistemático a egresados y su empleabilidad. El sistema actual no genera reportes estandarizados que cumplan con estos requerimientos, comprometiendo los procesos de evaluación externa de calidad.

La implementación de módulos avanzados que incorporen un generador dinámico de formularios, automatización de procesos de recolección de datos y una plataforma integrada de bolsa de trabajo permitirá optimizar la gestión del seguimiento a titulados. Estas mejoras facilitarán la obtención de información actualizada y confiable, fortalecerán la empleabilidad de los egresados mediante la vinculación con oportunidades laborales, y proporcionarán datos para la toma de decisiones académicas orientadas a la mejora continua de la calidad educativa.

1.5. FORMULACIÓN DEL PROBLEMA

¿Cómo optimizar la recolección y gestión de información sobre la inserción laboral de los titulados de la carrera de Ingeniería de Sistemas de la Universidad Amazónica de Pando mediante la implementación de módulos de automatización?

1.6. OBJETIVOS

1.6.1. Objetivo general

Implementar módulos de automatización en el sistema de seguimiento a titulados mediante metodología Scrum para optimizar la recolección automatizada de datos, mejorar la interacción con egresados y fortalecer la gestión de información sobre su situación laboral y profesional en la carrera de Ingeniería de Sistemas de la Universidad Amazónica de Pando.

1.6.2. Objetivos Específicos

- Analizar las necesidades funcionales del sistema actual mediante técnicas de levantamiento de requerimientos (entrevistas, encuestas, observación) para identificar y priorizar al menos tres módulos críticos de implementación.
- Diseñar la arquitectura modular del sistema utilizando patrones de desarrollo web escalables para garantizar la integración eficiente de nuevas funcionalidades y la extensibilidad futura del sistema.
- Desarrollar módulos especializados de automatización aplicando tecnologías web modernas (React/Spring Boot) para mejorar la recolección de datos y la interacción de los titulados con el sistema.
- Implementar servicios de integración externa mediante APIs y bases de datos centralizadas para conectar el sistema con recursos externos relevantes para el seguimiento profesional de los egresados.
- Evaluar la efectividad de los módulos implementados a través de métricas de usabilidad, rendimiento y satisfacción de usuarios para validar las mejoras en el proceso de seguimiento y planificar futuras expansiones.

1.7. METODOLOGÍA

Para el desarrollo del presente proyecto se aplicará la metodología Scrum, un marco de trabajo ágil que organiza el desarrollo en ciclos iterativos e incrementales denominados sprints, permitiendo la entrega continua de valor y la adaptación constante a los cambios de requerimientos (Schwaber y Sutherland, 2020). La elección de Scrum se basa en su capacidad para manejar la incertidumbre propia de los proyectos de sistemas de información educativos, donde los requisitos pueden cambiar a lo largo del desarrollo. Además, se destaca por su enfoque en la validación continua con los actores clave, asegurando que cada entrega genere valor real para la institución.

La metodología se organizará en tres fases clave: PRE-GAME, donde se definirá el marco de trabajo y se creará el Product Backlog inicial; GAME, que consiste en un desarrollo iterativo a través de nueve sprints distribuidos en tres bloques temáticos; y POST-GAME, centrada en la estabilización y el despliegue final del sistema. Esta estructura asegura una entrega progresiva de productos parciales que van directamente hacia el objetivo global,

incorporando retroalimentación de los usuarios en cada ciclo y adaptándose a nuevas necesidades que surjan durante el desarrollo.

TABLA 1 Metodología

FASES	DESCRIPCIÓN	TÉCNICAS	HERRAMIENTAS	PRODUCTOS
PRE--GAME	<ul style="list-style-type: none"> -Establecimiento del marco Scrum -Definición de roles y construcción de la lista inicial de tareas del producto 	<ul style="list-style-type: none"> -Lista de tareas del producto -Mapeo de historias de usuario -Estimación de esfuerzo 	<ul style="list-style-type: none"> -Google Meet -Google Docs 	<ul style="list-style-type: none"> -Lista de tareas del producto priorizado -Criterios de finalización -Roles definidos

<p style="text-align: center;">GAME</p>	<ul style="list-style-type: none"> -Desarrollo de los módulos en el sistema con funcionalidades básicas de seguimiento a titulados - Implementación de módulos especializados de recolección y gestión de datos - Desarrollo de funcionalidades avanzadas e integración de servicios externos 	<ul style="list-style-type: none"> -Planificación del Sprint -Reuniones diarias -Revisión del Sprint -Evaluación del Sprint -Pruebas de aceptación del usuario -Integración de la API -Pruebas de rendimiento -Pruebas de seguridad 	<ul style="list-style-type: none"> -Spring Boot -React -Node.js + Puppeteer (Scraping) - PostgreSQL -JMeter -OWASP Threat Dragon 	<ul style="list-style-type: none"> -Módulo de Formularios Dinámicos -Generador de encuestas -APIs básicas -Dashboard actualizado -Estadísticas
--	--	---	---	---

POST-GAME	Estabilización, optimización y preparación para producción	<ul style="list-style-type: none"> -Implementación final en el sistema -Optimización del rendimiento -Capacitación para usuarios 	Servidor de la Universidad Amazónica de Pando, herramientas de monitoreo, documentación técnica	<ul style="list-style-type: none"> -Sistema en producción -Documentación completa -Capacitación
------------------	--	---	---	--

Nota: Esta tabla resume el enfoque metodológico utilizado en el proyecto, detallando las fases, herramientas y productos entregables.

Fuente: Elaboración propia

1.8. JUSTIFICACIÓN

La implementación de módulos de fortalecimiento en el sistema de seguimiento a titulados de la carrera de Ingeniería de Sistemas de la Universidad Amazónica de Pando se justifica por la necesidad institucional de modernizar los procesos de gestión académica y fortalecer la vinculación con egresados mediante herramientas tecnológicas que permitan la recolección, análisis y gestión eficiente de información sobre su desempeño profesional. Este proyecto responde a la demanda de optimizar los mecanismos de retroalimentación para la mejora continua del programa académico, cumpliendo con estándares de calidad educativa que requieren evidencias objetivas sobre la inserción laboral y satisfacción de los titulados. La importancia del proyecto radica en su contribución al fortalecimiento del sistema de aseguramiento de la calidad académica, proporcionando datos confiables para la toma de decisiones estratégicas sobre actualización curricular, identificación de competencias emergentes y desarrollo de programas de formación continua que respondan a las necesidades del mercado laboral regional y nacional.

1.8.1. Justificación social

Desde la perspectiva social, el proyecto beneficia directamente a los 90 egresados de Ingeniería de Sistemas al facilitarles el acceso a oportunidades laborales específicas y mantener un vínculo continuo con su alma máter, promoviendo el desarrollo profesional y la inserción efectiva en el mercado tecnológico. Indirectamente, impacta positivamente en la sociedad pandina al fortalecer la formación de profesionales competentes que contribuyan al desarrollo tecnológico regional, mientras que la universidad se beneficia al contar con información actualizada que le permita adaptar su oferta académica a las demandas sociales y productivas del contexto amazónico boliviano.

1.8.2. Justificación económica

La implementación del sistema genera beneficios económicos al optimizar los procesos administrativos mediante la automatización de la recolección de datos, reduciendo costos operativos asociados a métodos manuales y liberando recursos humanos para actividades de mayor valor agregado. Para los titulados, facilita el acceso a mejores oportunidades laborales a través de la bolsa de trabajo integrada, incrementando potencialmente sus ingresos económicos, mientras que para la universidad representa una inversión estratégica que fortalece su posicionamiento institucional y contribuye a la sostenibilidad financiera a largo plazo mediante la mejora de indicadores de calidad que impactan en procesos de acreditación y financiamiento.

1.8.3. Justificación técnica

Técnicamente, el proyecto aporta valor al implementar una arquitectura modular y escalable basada en tecnologías web modernas (React/Spring Boot) que garantiza la extensibilidad del sistema para futuras funcionalidades, mientras que la adopción de metodología Scrum asegura la entrega incremental de valor y la adaptación continua a nuevos requerimientos. La integración de servicios externos mediante APIs y el diseño de una base de datos centralizada proporcionan una infraestructura robusta que mejora la eficiencia en el manejo de información, reduce la redundancia de datos y facilita la generación de reportes analíticos que contribuyen a la toma de decisiones basada en evidencias para el mejoramiento continuo de la calidad educativa.

1.9. ALCANCES

Aplicando la metodología Scrum, se ha definido el alcance funcional del sistema a través de historias de usuario priorizadas en el Product Backlog, abarcando los siguientes módulos y características:

Alcance Funcional:

- 1. Módulo de Generación Dinámica y Gestión de Datos:** Se implementará un motor de formularios que permita crear, modificar y publicar encuestas personalizadas (situación laboral, satisfacción académica) con lógica de validación automática. El alcance incluye la captura de respuestas y su almacenamiento estructurado centralizado.
- 2. Módulo de Vinculación Laboral:** El sistema incluirá servicios de integración con al menos dos plataformas de empleo externas (Trabajando.com.bo, Trabajopolis) mediante técnicas de extracción automatizada de datos (Web Scraping). Esto permitirá visualizar ofertas laborales filtradas por ubicación y palabras clave, centralizando las oportunidades para el titulado.
- 3. Módulo Administrativo y Estadístico:** Se desarrollará una vista gerencial capaz de generar automáticamente gráficos estadísticos en tiempo real basados en la data recolectada, facilitando la toma de decisiones sin procesamiento manual previo.

Alcance Técnico y de Integración: La arquitectura técnica se basa en Spring Boot para el backend (API REST), React para el frontend con diseño responsivo, y PostgreSQL como gestor de base de datos relacional. El sistema contempla la interoperabilidad con los sistemas académicos institucionales existentes (Sistema Integrado Siringuero) para la validación de usuarios.

Estrategia de Transferencia Tecnológica: El proyecto garantiza la sostenibilidad y operación autónoma del sistema mediante una capacitación diferenciada:

- 1. Para los Usuarios Finales (Titulados):** Se realizará una validación funcional con un grupo piloto y se entregará material multimedia (Video Tutorial Interactivo) sobre el llenado de formularios. Este enfoque asíncrono permite masificar la capacitación sin depender de sesiones presenciales.
- 2. Para los directores de Carrera:** La capacitación se gestionará mediante la entrega de Documentación Técnica Estandarizada (Manual de Administrador). Este material servirá

como guía de autoformación para que las autoridades dominen la elaboración de encuestas, el monitoreo de avance nominal y la interpretación del tablero de control estadístico.

Documentación Entregable: Se elaborará la documentación completa que incluye: especificaciones de requisitos (ERS), manuales de arquitectura y despliegue, y guías de usuario específicas para cada rol, asegurando la mantenibilidad del software.

Límites del Proyecto (Exclusiones): El proyecto no incluye el desarrollo de aplicaciones móviles nativas (Android/iOS), limitándose a interfaces web responsivas (Mobile-First). Asimismo, no implementa funcionalidades de inteligencia artificial predictiva ni machine learning avanzado, basando su motor de recomendación en coincidencia de patrones y palabras clave.

CAPITULO II: MARCO REFERENCIAL

2.1. MARCO INSTITUCIONAL

2.1.1. Universidad Amazónica de Pando

La Universidad Amazónica de Pando (UAP) es una institución de educación superior autónoma que forma parte del Sistema de la Universidad Boliviana (SUB). Fue creada mediante el Decreto Supremo N° 20511 del 21 de septiembre de 1984 y por la Ley N° 653 del 18 de octubre de 1984. La Constitución Política del Estado Plurinacional de Bolivia establece que la educación superior debe desarrollar procesos de formación profesional, generación y divulgación de conocimientos orientados al desarrollo integral de la sociedad, y debe ser intracultural, intercultural y plurilingüe. La misión de la UAP es formar profesionales idóneos con excelencia académica, pensamiento crítico y compromiso social, desarrollando investigación científica y tecnológica, y promoviendo la interacción social en un contexto de diversidad social e interculturalidad para contribuir al desarrollo integral de la Amazonía. La visión de la UAP es ser una institución educativa reconocida a nivel nacional e internacional para el año 2025, con carreras acreditadas, laboratorios y procesos administrativos certificados, enfocada en una gestión basada en resultados orientados al bienestar de la comunidad universitaria y al desarrollo integral de la región amazónica y del país.

2.1.2. Ingeniería de Sistemas

La Carrera de Ingeniería de Sistemas, está enmarcada y legalmente constituida en base al Estatuto Orgánico Universitario, Reglamentos, Políticas y Normativas vigentes. Fue creada mediante Resolución del Consejo Académico Universitario N° 034/2009 y homologada en el Honorable Consejo Universitario mediante la resolución N° 043/2009, del 28 de septiembre de 2009.

2.1.3. Misión

Formar profesionales íntegros, emprendedores, competentes en Sistemas de Información y Comunicación, Modelamiento y Optimización, Dirección y Gestión Empresarial, capaces de afrontar la era digital mediante la ciencia y la tecnología contribuyendo al desarrollo sostenible de la región y del país.

2.1.4. Visión

Carrera líder en investigación, transferencia tecnológica y formación de profesionales íntegros, con desempeño integral en el campo de la Ingeniería en Sistemas.

2.2. MARCO TEÓRICO

2.2.1. Sistemas de Información en Contexto Educativo

Un sistema de información web en educación superior es una plataforma accesible por navegador que integra diferentes componentes tecnológicos para gestionar información académica y administrativa en tiempo real. Entre sus elementos esenciales se encuentran el backend, encargado de procesar las reglas de negocio; el frontend, que permite la interacción entre el usuario y el sistema; una base de datos relacional, responsable del almacenamiento estructurado de la información; y módulos de seguridad, que garantizan la confidencialidad e integridad de los datos (Pulido Romero et al., 2019). Estas plataformas se caracterizan por su escalabilidad, lograda a través de arquitecturas multicapa; su ubicuidad, al operar mediante protocolos web como HTTP o HTTPS; y su capacidad para auditar el historial de transacciones realizadas. En el ámbito académico, los sistemas de información web se aplican ampliamente para el seguimiento del desempeño estudiantil, la gestión curricular y, en particular, el seguimiento a egresados. Su implementación permite recopilar, actualizar y analizar información sobre inserción laboral de manera eficiente, generando ciclos de retroalimentación útiles para la mejora continua. En el caso específico de la Carrera de Ingeniería de Sistemas de la UAP, la pertinencia de este tipo de sistema radica en su capacidad para reemplazar los procesos manuales basados en formularios físicos, habilitando análisis estadísticos en tiempo real y fortaleciendo la toma de decisiones sustentada en evidencia.

2.2.2. Arquitecturas de Aplicaciones Web

Las aplicaciones web modernas siguen la arquitectura cliente-servidor multicapa, donde el navegador ejecuta la vista (HTML/CSS/JavaScript) y el servidor provee servicios REST sobre HTTP (Java + Spring Boot). Este enfoque facilita la separación de responsabilidades: presentación, lógica y persistencia (CELI PÁRRAGA et al., 2023). Patrones de diseño como MVC (Modelo-Vista-Controlador) y microservicios promueven mantenibilidad y escalabilidad horizontal. El uso de React en el cliente aprovecha un DOM virtual para renderizados eficientes, mientras que Spring Boot simplifica la configuración del contenedor y expone APIs RESTful seguras mediante Spring Security. En entornos educativos, estas arquitecturas permiten desplegar sistemas modulares, por

ejemplo, un microservicio exclusivo de encuestas a titulados y sostener picos de demanda durante periodos de graduación.

2.2.3. Gestión de Bases de Datos Relacionales

Una base de datos relacional organiza la información en tablas con claves primarias y foráneas, garantizando integridad referencial y eliminando redundancias PostgreSQL, por ser ORDBMS open-source, ofrece transacciones ACID, extensiones JSONB y funciones agregadas que facilitan consultas analíticas. Las prácticas de normalización (1FN-3FN) mejoran la consistencia, mientras que índices B-tree y GIN aceleran búsquedas. En sistemas de seguimiento a titulados, el modelo de datos suele contemplar egresado, historial laboral, programas de posgrado y retroalimentaciones. La optimización se logra mediante vistas materializadas para reportes y políticas de partición por gestión académica. Además, estrategias de backup y recuperación (puntos de restauración, WAL archiving) son críticas para resguardar la información sensible de los egresados y cumplir normativas de protección de datos. En este proyecto, una BD bien diseñado asegura consultas rápidas para generar estadísticas de inserción laboral y cruzar competencias adquiridas con requerimientos del mercado. (Elmasri y B. Navathe, 2016).

2.2.4. Metodologías de Desarrollo de Software

Entre los enfoques ágiles, Scrum destaca por dividir el proyecto en sprints de duración regular, cada uno con planificación, ejecución, revisión y retrospectiva. Sus roles Product Owner, Scrum Master y Development Team fomentan la entrega incremental y la inspección continua. Para proyectos académicos, Scrum reduce la sobrecarga documental y permite validar entregables con personal administrativo en ciclos cortos. Complementariamente, la Programación Orientada a Aspectos (AOP) introduce el concepto de aspectos para encapsular funcionalidades transversales (seguridad, logging), mejorando la modularidad de aplicaciones Java (COLOMBO y PACE, 2022). La combinación Scrum + AOP facilita incorporar requisitos de auditoría y métricas sin alterar el código de negocio principal. En el sistema de seguimiento a titulados, esta sinergia permite añadir, por ejemplo, un aspecto de trazabilidad de cambios en los perfiles de egresados sin reescribir los controladores REST. (Schwaber y Sutherland, 2020).

2.2.5. Seguimiento a Egresados: Enfoques Teóricos

El seguimiento a egresados se sustenta en teorías de mejora continua y vinculación universidad-sociedad, donde la retroalimentación de titulados informa la pertinencia curricular y las necesidades del mercado laboral (CCANTO BALVIN, 2023). Modelos de gestión del talento proponen métricas de empleabilidad, trayectoria profesional y satisfacción con la formación. Herramientas web, como módulos de encuestas integradas, permiten aplicar el ciclo PDCA (Plan-Do-Check-Act) sobre la malla curricular: planificar ajustes, desplegarlos, medir impacto y corregir desviaciones. Además, enfoques de learning analytics extraen patrones a partir de respuestas longitudinales de los egresados, identificando competencias emergentes. En la UAP, disponer de un sistema centralizado que recoja y visualice estos datos empodera a la carrera para justificar actualizaciones de asignaturas, proponer cursos de formación continua y fortalecer redes de colaboración con el sector productivo amazónico.

2.3. MARCO TECNOLÓGICO

2.3.1. Herramientas para el análisis de requerimientos.

Para garantizar la correcta ingeniería de requisitos, se utilizaron herramientas que permitieron combinar la formalidad del modelado visual con la flexibilidad de la gestión ágil, bajo normativas internacionales.

- **Google Workspace (Docs & Sheets)**

Definición: Es un conjunto de herramientas de productividad basadas en la nube ofrecidas por Google. Estas herramientas incluyen aplicaciones para el correo electrónico (Gmail), almacenamiento en la nube (Google Drive), comunicación (Google Meet, Google Chat), y aplicaciones de productividad como Google Docs (procesador de texto) y Google Sheets (hoja de cálculo), entre otros. Google Workspace está diseñado para facilitar la colaboración en tiempo real entre equipos y organizaciones, ofreciendo una plataforma integrada que permite acceder y trabajar en documentos, hojas de cálculo, presentaciones y otros archivos desde cualquier lugar con conexión a Internet. (Google, 2025)

Aplicación en el Análisis: Se utilizó como el entorno principal para la Elicitación Colaborativa y Validación de Requisitos.

- 1. Trabajo Síncrono:** Facilitó las sesiones de revisión con los Stakeholders (Tutor y Asesor). En lugar de intercambiar múltiples versiones de archivos estáticos, se trabajó sobre un "Documento Vivo", permitiendo la corrección inmediata de la redacción de los requisitos funcionales.
- 2. Historial de Versiones:** La plataforma garantizó la trazabilidad de cambios, permitiendo recuperar versiones anteriores del documento de requisitos si una modificación no era aprobada.
- 3. Feedback Directo:** Se utilizó el sistema de "Comentarios" para resolver dudas sobre reglas de negocio específicas (ej. criterios de filtrado del scraping) directamente sobre el texto del requerimiento, agilizando el ciclo de aprobación del ERS.

2.3.2. Herramientas para el diseño del sistema

- **Enterprise Architect**

Descripción: es una herramienta de modelado visual desarrollada por Sparx Systems que permite diseñar, analizar y gestionar arquitecturas empresariales, sistemas, software y procesos de negocio. Se usa para crear modelos detallados que representan estructuras organizacionales, procesos, sistemas y relaciones entre ellos, facilitando la comunicación y documentación técnica y de negocio. En el contexto de modelos de negocio, Enterprise Architect permite elaborar diagramas como:

- Modelos de procesos de negocio (BPMN)
- Modelos UML (casos de uso, clases, actividades)
- Mapas de valor y flujos de trabajo

Su uso facilita la planificación, análisis y mejora continua de procesos y estructuras dentro de una organización, apoyando la toma de decisiones y alineando la estrategia con la operación. (Systems, 2024)

Uso: Se utilizó para documentar la arquitectura completa del sistema, asegurando la trazabilidad entre los requisitos y la implementación. Se elaboraron los siguientes artefactos:

- 1. Modelado Estructural (Lógica del Negocio):**

Diagrama de Clases del Generador Dinámico: Se diseñó el modelo de dominio complejo para el módulo de encuestas, definiendo las relaciones de composición

entre Encuestas, Secciones, Preguntas y Respuestas, lo cual fue la base para el diseño del esquema de base de datos relacional.

2. Modelado de Comportamiento (Interacción):

Diagramas de Secuencia: Se modelaron los flujos críticos para validar la comunicación asíncrona entre el Frontend y el Backend:

- Creación y Publicación de Encuestas: Detallando la instanciación de metadatos.
- Procesos de Scraping: Visualizando la orquestación entre el servicio Node.js, la extracción en fuentes externas y la consolidación en Java.
- Generación de Estadísticas: Definiendo el algoritmo de cálculo automático de reportes.

3. Modelado de Arquitectura (Componentes):

- **Diagrama de Componentes General:** Para visualizar la arquitectura de alto nivel.
- **Desglose por Capas:** Se elaboraron diagramas específicos para el Backend (Controladores y Servicios Spring Boot), el Frontend (Componentes React/JSX) y el Módulo de Scraping (Microservicio de Extracción), evidenciando el desacoplamiento modular del sistema.

4. Modelado de Implementación:

- **Diagrama de Despliegue:** Para representar la topología física del sistema, detallando los nodos de ejecución (Servidor de Aplicaciones, Contenedor de Base de Datos, Cliente Web) y los protocolos de comunicación utilizados en el entorno de producción de la DIA.

5. Modelado de Comportamiento (Casos de Uso):

- Se elaboraron los Diagramas de Casos de Uso para formalizar los requisitos funcionales capturados en el Product Backlog. La herramienta permitió definir gráficamente los Límites del Sistema (System Boundary), identificar a los actores clave (Titulado, Director de carrera, Scraping Automatizado).

- **Power Designer**

Descripción: Es una herramienta de modelado de datos y arquitectura empresarial desarrollada por SAP que permite diseñar, analizar y administrar bases de datos, procesos de negocio y arquitecturas de sistemas complejos. Su principal función es

ayudar a las organizaciones a visualizar y gestionar el diseño de sus sistemas de información, facilitando la integración entre los diferentes componentes de TI y el negocio.

PowerDesigner soporta diversos tipos de modelado, incluyendo:

- a. Modelado de datos (conceptual, lógico y físico)
- b. Modelado de procesos de negocio (BPMN)
- c. Generación automática de código para bases de datos

Gracias a su capacidad de sincronizar modelos con bases de datos reales, ayuda a mantener la coherencia entre el diseño y la implementación, optimizando el desarrollo y mantenimiento de sistemas. (SAP, 2024)

Uso: Se utilizó estrictamente en la fase de Análisis y Diseño de Datos, cumpliendo dos funciones críticas antes de la codificación:

- 1. Diseño Conceptual (Normalización):** Permitted elaborar el Diagrama Entidad-Relación (E-R) para validar la lógica del negocio (ej. identificar cardinalidades complejas y eliminar redundancias) antes de escribir una sola línea de código.
- 2. Definición de Tipos y Atributos:** Sirvió como el "Plano Arquitectónico" de referencia. En él se definieron los tipos de datos exactos (Enteros, Varchar, Date) y las restricciones. Este modelo fue la guía base para programar las Clases de Entidad (@Entity) en Spring Boot, delegando posteriormente al framework (vía JPA/Hibernate) la tarea operativa de la generación automática de tablas y relaciones en la base de datos PostgreSQL.

2.3.3. Herramientas para el desarrollo del sistema

- **Spring Tool Suite (STS)**

Descripción: Es un entorno de desarrollo integrado (IDE) especializado para el desarrollo de aplicaciones basadas en el framework Spring, una plataforma muy popular para crear aplicaciones en Java. STS está basado en Eclipse, pero añade herramientas, asistentes y configuraciones optimizadas específicamente para trabajar con los proyectos Spring de manera más eficiente. (CELI PÁRRAGA R. J., 2022)

Uso: Fue el entorno de ingeniería principal para la codificación del Backend. Su elección se justificó por tres funcionalidades críticas que aceleraron el desarrollo:

1. **Boot Dashboard:** Permitió gestionar el ciclo de vida del servidor (iniciar, detener, reiniciar) y visualizar los puertos de despliegue del microservicio localmente sin necesidad de comandos de consola.
 2. **Asistencia Inteligente de Código:** Proporcionó autocompletado y validación sintáctica para las anotaciones estereotipo de Spring (@Controller, @Service, @Repository), reduciendo errores de configuración.
 3. **Integración con Maven:** Facilitó la gestión del archivo pom.xml, descargando y actualizando automáticamente las librerías necesarias (dependencias de JPA, Security, Web) desde los repositorios centrales.
- **Visual Studio Code** es un editor de código fuente multiplataforma desarrollado por Microsoft que permite trabajar en Windows, macOS y Linux. Su objetivo es ofrecer un entorno ligero pero potente que incluya funcionalidades esenciales como depuración, control de versiones (Git integrado), resaltado de sintaxis, autocompletado inteligente (IntelliSense), fragmentos de código (snippets) y la posibilidad de extenderse mediante extensiones. Además, VS Code soporta un modelo de trabajo eficiente para desarrollos web y en la nube, e incluye versiones web que permiten edición ligera sin necesidad de instalación. (MICROSOFT, 2025)

Uso: Se utilizó exclusivamente como el entorno de desarrollo para el Microservicio de Web Scraping. Su elección se fundamentó en tres capacidades críticas para trabajar con Puppeteer:

1. **Depuración de Node.js:** Permitió ejecutar el debugger paso a paso para inspeccionar el comportamiento del bot mientras navegaba por las páginas web, facilitando la detección de selectores CSS erróneos.
2. **Terminal Integrada:** Facilitó la gestión de paquetes NPM (Node Package Manager) y la ejecución de pruebas rápidas de los scripts de extracción sin salir del editor.
3. **Soporte de Sintaxis:** Su autocompletado inteligente para librerías asíncronas (async/await) redujo significativamente los errores de sintaxis en el manejo de promesas durante la navegación automatizada.

TECNOLOGÍAS DEL LADO DEL SERVIDOR (BACKEND)

- **SPRINGBOOT:**

Descripción: es un framework basado en Spring que permite desarrollar aplicaciones Java de manera rápida y con una configuración mínima. Proporciona un conjunto de herramientas y dependencias preconfiguradas que simplifican el proceso de desarrollo, especialmente para aplicaciones web y microservicios, facilitando la creación de proyectos listos para producción. (Team Spring, 2025)

Uso en el proyecto: Constituye el núcleo arquitectónico del sistema, desempeñando tres funciones críticas:

1. **Orquestador de Servicios RESTful:** Expone los endpoints de la API que consumen tanto el cliente web (React) como el servicio de scraping (Node.js), centralizando la lógica de negocio.
2. **Gestión de Persistencia (Spring Data JPA):** Implementa la capa de acceso a datos utilizando el estándar JPA sobre Hibernate, mapeando las clases Java directamente a las tablas de PostgreSQL (ORM) para optimizar las operaciones CRUD.
3. **Seguridad y Autenticación (JWT):** En lugar de sesiones de servidor tradicionales, se implementó un mecanismo de seguridad basado en el estándar JSON Web Token (JWT).
 - **Funcionamiento:** Permite una autenticación sin estado (Stateless). Cuando el titulado inicia sesión, el sistema genera un token cifrado y firmado digitalmente. Este token es enviado en el encabezado (Header Authorization) de cada petición subsiguiente, permitiendo al servidor validar la identidad del usuario de manera eficiente sin consultar la base de datos en cada "clic", lo cual es ideal para la arquitectura distribuida del proyecto.

- **POSGRESQL**

Descripción: es un sistema de gestión de bases de datos relacional orientado a objetos (ORDBMS), de código abierto y altamente extensible. Soporta estándares SQL avanzados, integridad referencial, transacciones ACID, y permite la creación de tipos de datos personalizados, funciones definidas por el usuario y mecanismos de control de concurrencia. Es utilizado para aplicaciones que requieren alta confiabilidad, escalabilidad y complejidad en el manejo de datos. (MOMJIAN, 2013)

Uso: Se implementó como el repositorio central de información, cumpliendo tres funciones estructurales:

- 1. Persistencia del Núcleo Académico:** Almacena el modelo relacional complejo diseñado en PowerDesigner (Tablas: Titulados, Encuestas, Respuestas). Su motor de restricciones (Constraints y Foreign Keys) actúa como la última línea de defensa para garantizar la Integridad Referencial, impidiendo, por ejemplo, que existan respuestas huérfanas sin un titulado asociado.
- 2. Interoperabilidad con el Backend:** Interactúa nativamente con el driver JDBC de Spring Boot. Gracias al dialecto de Hibernate para PostgreSQL, el sistema traduce automáticamente las entidades Java a sentencias SQL optimizadas, gestionando eficientemente el Pool de Conexiones para soportar la concurrencia de usuarios (validada en las pruebas de estrés).

- **JAVA**

Descripción: Es un lenguaje de programación de propósito general, orientado a objetos y basado en clases, que permite desarrollar aplicaciones portables, seguras y robustas. Fue diseñado para ejecutarse en múltiples plataformas gracias a la Java Virtual Machine (JVM), lo que garantiza que un programa escrito en Java pueda funcionar en diferentes sistemas operativos sin necesidad de modificaciones. (CELI PÁRRAGA R. J., 2022).

- **NODE JS** es un entorno de ejecución de un solo hilo, de código abierto y multiplataforma para crear aplicaciones de red y del lado del servidor rápidas y escalables. Se ejecuta en el motor de ejecución de JavaScript V8, y utiliza una arquitectura de E/S basada en eventos y sin bloqueos, lo que la hace eficiente y adecuada para aplicaciones en tiempo real. (KINSTA, 2025)

- **PUPPETER (MOTOR DE NAVEGACIÓN AUTOMATIZADA)**

Descripción: Puppeteer es una biblioteca de programación para Node.js que proporciona una interfaz de alto nivel para controlar navegadores (principalmente Chrome/Chromium) mediante el Protocolo de DevTools. Se utiliza para automatizar tareas de navegación web como pruebas de interfaz, extracción de datos (web scraping) y generación de capturas de pantalla o PDFs. La ejecución puede realizarse en modo headless (sin interfaz visible) lo que facilita su integración en pipelines de CI/CD. Por ejemplo, un estudio comparativo de herramientas de automatización de navegadores

incluye a Puppeteer evaluando su arquitectura, desempeño y características frente a otras herramientas como Selenium y Cypress. (GARCÍA et al., 2024)

Uso: Su implementación fue crítica para el módulo de extracción debido a que las bolsas de trabajo seleccionadas (Trabajando.com.bo, Trabajopolis) utilizan Renderizado del Lado del Cliente (CSR). A diferencia de un crawler tradicional, Puppeteer ejecuta el JavaScript del sitio, permitiendo:

- 1. Automatizar flujos complejos:** Gestionar la paginación asíncrona (clic en "Siguiente" sin recargar URL).
- 2. Evasión de Bloqueos:** Simular latencia humana y eventos de entrada (mouse/teclado) para mitigar los mecanismos básicos de detección anti-bot (WAF).

- **CHEERIO:**

Descripción: Implementación rápida y ligera de jQuery diseñada para el servidor.

Uso: En la arquitectura del Scraper, Cheerio actúa como el componente de extracción selectiva. Una vez que Puppeteer recupera el código fuente de la página, Cheerio recorre el árbol de nodos DOM para identificar y extraer los metadatos críticos de cada oferta laboral (Cargo, Empresa, Ubicación, Fecha) mediante selectores CSS precisos. Su uso optimiza el consumo de memoria del servidor, ya que procesa texto plano en lugar de mantener objetos gráficos en memoria.

- **EXPRESS.JS**

Descripción: Es un framework web minimalista y flexible para Node.js que facilita la creación de aplicaciones web y APIs. Fue diseñado para simplificar el proceso de construcción de aplicaciones web al proporcionar un conjunto robusto de características para manejar solicitudes HTTP, rutas, middleware y otras funcionalidades comunes de aplicaciones web. Express.js es muy popular debido a su simplicidad, escalabilidad y eficiencia. (Contributors, 2025)

Uso: Se implementó para estructurar y exponer la lógica de extracción de datos como una API RESTful interna. Su función crítica es actuar como capa de interoperabilidad, recibiendo las peticiones HTTP del sistema central (Sistema de Graduados UAP) y entregando los resultados del scraping depurados en formato JSON estándar, facilitando así el desacoplamiento entre el motor de búsqueda y la aplicación principal.

TECNOLOGÍAS DEL LADO DEL CLIENTE (FRONT END)

Para el desarrollo de la interfaz de usuario, se priorizó la construcción de una Single Page Application (SPA) que garantice la fluidez en la navegación y la adaptabilidad a dispositivos móviles.

- **REACT.JS Y JSX (Sintaxis de Desarrollo)**

Descripción: Es una biblioteca de JavaScript desarrollada por Facebook, utilizada para construir interfaces de usuario interactivas, especialmente en aplicaciones web de una sola página (SPA). Su principal característica es el uso de componentes reutilizables y un flujo de datos unidireccional, lo que permite desarrollar interfaces modulares, eficientes y fáciles de mantener. (CASAS, 2021)

Uso: Se implementó React.js integrando la sintaxis JSX (JavaScript XML) para la arquitectura del Motor de Formularios Dinámicos. Esta combinación permitió ejecutar un renderizado condicional declarativo, gestionando la lógica de ramificación de preguntas en tiempo real (visibilidad dependiente de respuestas previas). Gracias a la gestión optimizada del Virtual DOM, se garantiza una actualización instantánea de la interfaz sin recargas de página (SPA), factor crítico para mitigar la fatiga cognitiva del usuario y asegurar la completitud del llenado..

- **BOOTSTRAP**

Descripción: Es un framework de código abierto desarrollado originalmente por Twitter que facilita el diseño y desarrollo de sitios web responsivos y móviles. Proporciona una colección de componentes predefinidos en HTML, CSS y JavaScript, como botones, menús, formularios y grillas, que permiten construir interfaces visuales modernas sin necesidad de programar estilos desde cero. Es ampliamente utilizado por su compatibilidad con navegadores y su facilidad de uso. (CUESTA, 2020)

Uso: Se utilizó Bootstrap para la maquetación "Mobile-First" de los formularios.

- **Sistema de Grillas:** Permitted estructurar la alineación de las preguntas para que se visualicen en una sola columna en celulares y se expandan en pantallas de escritorio.
- **Estilos de Formulario:** Se implementaron las clases nativas de Bootstrap (form-control, input-group) combinadas con CSS personalizado para lograr la

interfaz limpia tipo "Quiz", garantizando que los botones y campos de texto tengan áreas táctiles adecuadas para dedos en pantallas táctiles.

2.3.4. Herramientas para la implementación del sistema

- **Apache TOMCAT** es un servidor web y contenedor de servlets desarrollado por la Fundación Apache que permite ejecutar aplicaciones web basadas en Java. Es una implementación de referencia para las especificaciones Java Servlet, JavaServer Pages (JSP), Java Expression Language y WebSocket, que forman parte de la plataforma Java EE (Jakarta EE). Tomcat se utiliza ampliamente para desplegar aplicaciones Java en entornos de producción y desarrollo debido a su robustez, flexibilidad y comunidad activa. (SÁNCHEZ J. M., 2022)
- **Backend (Spring Boot): Maven** es una herramienta de automatización y gestión de proyectos para aplicaciones Java. Facilita la compilación, empaquetado, gestión de dependencias, documentación y despliegue de software. Maven utiliza un archivo de configuración llamado pom.xml donde se definen las dependencias, plugins y otras configuraciones del proyecto. Gracias a Maven, los desarrolladores pueden manejar proyectos complejos con múltiples módulos y asegurar la reproducibilidad y consistencia en el proceso de construcción. (GARCÍA L. , 2021)
- **Frontend (React): npm:** (Node Package Manager) es el sistema de gestión de paquetes predeterminado para el entorno de ejecución Node.js. Permite a los desarrolladores instalar, compartir y gestionar librerías y dependencias para proyectos JavaScript y frontend, como aplicaciones React. Con npm, se pueden administrar paquetes tanto locales como globales, facilitando la reutilización de código y la colaboración entre desarrolladores. (PÉREZ C. , 2021)

2.3.5. Herramientas para la evaluación del sistema

- **Apache JMeter**
Descripción: Es una herramienta de código abierto diseñada para realizar pruebas de rendimiento y carga en aplicaciones web, bases de datos, servicios web y más. Su

principal objetivo es simular el comportamiento de los usuarios en un entorno controlado para evaluar el rendimiento y la capacidad de un sistema bajo carga. JMeter fue creado originalmente para probar aplicaciones web, pero con el tiempo se ha expandido para ser utilizado en pruebas de otros servicios como bases de datos, servidores FTP, servicios SOAP y REST, entre otros. (Foundation, 2025)

Uso: Fue la herramienta fundamental para las pruebas de estrés (Objetivo 5), permitiendo simular la concurrencia de 2,200 usuarios para validar la escalabilidad del servidor.

2.3.6. Herramientas para la documentación técnica del sistema

- **Manual de usuario** de un sistema de información es un documento que proporciona instrucciones detalladas y claras para que los usuarios finales puedan operar y utilizar correctamente el sistema. Incluye desde la instalación, configuración, uso de funcionalidades, solución de problemas comunes, hasta recomendaciones para maximizar el aprovechamiento del sistema. Su propósito es facilitar la interacción del usuario con el sistema, mejorar la experiencia y reducir errores en su uso. (MARTÍNEZ, 2020)
- **Diccionario de base de datos** es un repositorio centralizado que contiene información descriptiva sobre los datos almacenados en una base de datos. Incluye metadatos como la estructura de las tablas, tipos de datos, relaciones entre tablas, restricciones, índices y descripciones de los elementos. Este diccionario ayuda a los desarrolladores y administradores a entender y gestionar la base de datos, facilitando su mantenimiento, diseño y uso eficiente. (PÉREZ J. , 20019)

2.3.7. Herramientas para estándares de Seguridad Web

- **Proyecto OWASP (Open Web Application Security Project)**
Definición: El OWASP es una fundación sin fines de lucro dedicada a mejorar la seguridad del software. Su misión es permitir que organizaciones diseñen, desarrollen, adquieran, operen y mantengan aplicaciones confiables mediante herramientas, documentación, estándares y comunidades abiertas.
OWASP publica de forma libre y accesible información, guías, herramientas y recomendaciones que ayudan a identificar y mitigar las vulnerabilidades más críticas

que afectan a las aplicaciones web. Entre sus recursos más conocidos está OWASP Top 10, un listado reconocido internacionalmente de los riesgos de seguridad más peligrosos en aplicaciones web. (Celi Álvarez, 2025)

Aplicación en el Proyecto: Se utilizó este estándar como guía de buenas prácticas durante la fase de codificación (Secure Coding) para mitigar las siguientes vulnerabilidades:

- **Inyección SQL:** Mitigada mediante el uso de Spring Data JPA, que utiliza consultas parametrizadas por defecto, impidiendo la manipulación de la base de datos PostgreSQL.
- **Cross-Site Scripting (XSS):** Mitigada nativamente por React.js, que escapa automáticamente cualquier contenido renderizado en la vista, evitando la ejecución de scripts maliciosos.
- **Pérdida de Autenticación:** Mitigada mediante la implementación de tokens JWT firmados criptográficamente, eliminando el uso de IDs de sesión secuenciales vulnerables.

CAPITULO III: DESARROLLO

3.1. FASE PRE-GAME

3.1.1. Establecimiento del marco Scrum

Se estableció el marco de trabajo Scrum para el desarrollo del sistema, definiendo ciclos iterativos de 2 semanas (sprints) con revisiones y retrospectivas al finalizar cada ciclo.

Se determinó que el proyecto se ejecutaría en 12 sprints distribuidos en 3 bloques temáticos:

- **Bloque 1 (Sprints 1-3):** Análisis, diseño de base de datos y arquitectura base.
- **Bloque 2 (Sprints 4-9):** Desarrollo del núcleo (Core): Generador de Encuestas y Scraping.
- **Bloque 3 (Sprints 10-12):** Integración, pruebas de estrés (JMeter) y despliegue.

Ceremonias y Tiempos: El ciclo de vida se configuró respetando los eventos canónicos de Scrum:

- **Sprint Planning (4 horas):** Inicio del ciclo para seleccionar historias del Product Backlog.
- **Daily Scrum (15 minutos):** Sincronización diaria para reportar impedimentos.
- **Sprint Review (2 horas):** Demostración funcional de los avances (Demo) ante el Tutor/Asesor.
- **Sprint Retrospective (1.5 horas):** Análisis de mejora continua del proceso de desarrollo.

3.1.2. Definición de roles del equipo

Para garantizar la correcta integración de los nuevos módulos dentro del ecosistema tecnológico de la Universidad (Sistema Siringuero), se adaptaron los roles de Scrum considerando la estructura institucional de la Dirección de Información Académica (DIA).

A. Product Owner (Dueño del Producto Académico)

- PhD. Humberto Fernández Calle (Tutor)
- **Función Principal:** Garante de la calidad investigativa y metodológica.
- **Responsabilidades:**
 - Validación de los entregables frente a los requisitos del reglamento de grado.
 - Priorización del Backlog en función del alcance académico comprometido.
 - Aprobación de la documentación técnica y manuales de usuario.

B. Scrum Master (Facilitador Metodológico)

- Ing. Henry Ivan Surco Alvan (Asesor)
- **Función Principal:** Aseguramiento de la metodología y supervisión de avance.
- **Responsabilidades:**

- Eliminación de impedimentos en el flujo de trabajo.
- Monitoreo del cumplimiento de los Sprints y tiempos de desarrollo.
- Revisión de la coherencia entre el código desarrollado y los objetivos específicos.

C. Development Team (Equipo de Desarrollo e Implementación)

El equipo de ejecución técnica estuvo conformado por dos roles operativos diferenciados:

a) Desarrollador Full-Stack (Postulante):

- Univ. Jesus Israel Condori Choquehuanca
- **Responsabilidad:** Ejecución del 100% de la codificación de los nuevos módulos (Web Scraping y Formularios Dinámicos). Encargado de la lógica de negocio, diseño de interfaces en React y consumo de servicios.

b) Arquitecto de Software / Referente Técnico (DIA):

- Ing. Dixson Gonzalo Villca Choque (**Apoyo en Desarrollo de Sistemas - DIA**)
- **Responsabilidad:** Proveedor de la infraestructura base. Su rol fue crítico para facilitar el acceso a las API REST del Sistema Integrado Siringuero y la base de datos de titulados. Supervisó que los nuevos módulos cumplieran con los estándares de seguridad e interoperabilidad de la Universidad.

D. Stakeholders Clave (Interesados)

- **Lic. Guillermina Suarez Noza (Directora DIA):** Aprobación institucional para el despliegue.
- **Lic. Eduardo Zubieta Copeticon (director Carrera Ing. Sistemas):** Beneficiario directo de los reportes.
- **Ing. Fader Cabrera Arandia (responsable de Evaluación y Acreditación del Área de Ciencias y Tecnología):** Validador de los instrumentos de recolección de datos (formularios).

E. Mecanismos de Comunicación y Coordinación

- **Desarrollo Colaborativo In-situ (Unidad de Sistemas Académicos):** Se estableció una dinámica de trabajo presencial en las instalaciones de la Unidad de Sistemas Académicos (dependiente de la DIA). Se realizó programación conjunta (Pair Programming) con el Ing. Dixson Villca para el análisis, documentación y

consumo seguro de los endpoints (API REST) del Sistema Siringuero, garantizando la correcta integración de los datos en el entorno de producción de la universidad.

- **Revisiones de Avance:** Sesiones quincenales de Sprint Review con el Asesor y Tutor para validar funcionalidad vs. documento.

3.1.3. Construcción del Product Backlog

Se construyó el Product Backlog inicial con 11 historias de usuario organizadas por módulos funcionales y priorizadas según valor de negocio y dependencias técnicas.

TABLA 2 *Priorización inicial del Product Backlog*

ID	Historia de Usuario	Módulo	Prioridad	Story Points	Sprint Asignado
HU-009	Autenticación segura con JWT	Seguridad y acceso	Alta	5	Sprint 1
HU-011	Sincronización con Siringuero	Integración institucional	Media	8	Sprint 2-3
HU-001	Creación de formularios dinámicos	Generador de formularios	Alta	8	Sprint 4
HU-002	Edición de formularios publicados	Generador de formularios	Alta	5	Sprint 4
HU-003	Publicación con fechas de vigencia	Generador de formularios	Alta	5	Sprint 5
HU-010	Fecha límite y cierre de encuesta	Gestión de formularios	Media	3	Sprint 5
HU-004	Responder formulario publicado	Recolección de respuestas	Alta	8	Sprint 6
HU-007	Panel de estadísticas con gráficos	Estadísticas y seguimiento	Alta	8	Sprint 6-7
HU-005	Búsqueda y filtrado de ofertas	Vinculación laboral	Alta	8	Sprint 8
HU-006	Postulación a ofertas laborales	Vinculación laboral	Alta	5	Sprint 9

HU-008	Importación automática (scraping)	Integración externa	Media-Alta	13	Sprint 10-11
---------------	-----------------------------------	---------------------	-------------------	----	--------------

Nota: Se presenta el listado de requerimientos priorizados al inicio del proyecto para organizar los sprints de desarrollo.

Fuente: Elaboración propia

3.1.4 Levantamiento de requerimientos

El proceso de ingeniería de requisitos se ejecutó durante las primeras 2 semanas del proyecto (Fase Pre-Sprint), aplicando un enfoque mixto de recolección de datos para garantizar que el software resuelva los problemas críticos de la DIA.

Técnicas de Elicitación Aplicadas: Para triangular la información, se utilizaron cuatro técnicas complementarias

- **Entrevistas Semiestructuradas:** Se realizaron 3 sesiones formales con la Responsable de Seguimiento a Titulados para comprender el flujo normativo de la acreditación.
- **Observación Directa (Job Shadowing):** Se dedicaron 2 jornadas laborales a observar “in situ” los procesos manuales actuales, identificando cuellos de botella en la transcripción de formularios físicos a Excel.
- **Análisis Documental:** Se realizó la ingeniería inversa de los instrumentos actuales (Formularios en Word/Excel y Reportes de Acreditación pasados) para extraer los campos de datos obligatorios.

Mapa de Actores Involucrados (Stakeholders): La validación de necesidades contó con la participación activa de los roles clave del negocio:

- **Nivel Estratégico:** Directora de Información Académica (DIA).
- **Nivel Táctico:** Director de la Carrera de Ingeniería de Sistemas.
- **Nivel Operativo:** Responsable de Seguimiento a Titulados.
- **Usuarios Finales:** 2 titulados representantes que validaron la usabilidad del módulo de bolsa de trabajo.

Resultado del Proceso (Product Backlog): Como resultado de esta fase, se identificaron 3 Módulos Principales (Gestión de Encuestas, Vinculación Laboral y Analítica), los cuales se desglosaron en un Product Backlog inicial. Este listado priorizado transformó los requerimientos brutos en Historias de Usuario técnicas y criterios de aceptación verificables.

3.1.4.1 Requerimientos funcionales – Módulo Generador Dinámico de Formularios

Este apartado describe las especificaciones funcionales del núcleo del sistema. El módulo de Generador Dinámico tiene como objetivo abstraer la estructura de las encuestas, permitiendo al administrador definir, versionar y publicar instrumentos de recolección de datos sin necesidad de modificar el código fuente o realizar nuevos despliegues. Los requisitos detallados a continuación garantizan la autonomía de la Dirección de Información Académica para adaptar los formularios a los criterios cambiantes de acreditación.

TABLA 3 *Requerimiento Funcional: Creación de formulario de encuesta dinámico*

RF-001 – Creación de formulario de encuesta dinámico	
Identificador	RF-001
Nombre	Creación de formulario de encuesta dinámico
Descripción	El sistema permitirá a directores de carrera crear formularios personalizados con diferentes tipos de preguntas de forma dinámica sin programación.
Actor/Usuario	Director de Carrera
Prioridad	Alta
Precondiciones	Usuario autenticado con rol director de carrera
Entradas	Título Sección, descripción, tipo de pregunta, texto de la pregunta, opciones de respuesta, indicador de opción añadir otra opción.
Proceso	Interfaz visual para crear el formulario, ingresar nombre del formulario, luego en el formulario agregar las preguntas, seleccionar tipo, validar y guardar.
Salidas	Formulario guardado en base de datos con identificador único.
Excepciones	Error de validación o conexión a la base de datos.
Criterios de Aceptación	Creación rápida de formulario, formulario almacenado correctamente, confirmación visible.
Dependencias	Ingreso al Sistema PerlaPlus que es exclusivamente para directores de carrera.

Nota: Especificación técnica del requerimiento para la creación de formularios, incluyendo sus entradas, procesos y criterios de aceptación.

Fuente: Elaboración propia

TABLA 4 *Requerimiento Funcional: Modificación de formulario de encuesta*

RF-002 - Modificación de formulario de encuesta	
Identificador	RF-002
Nombre	Modificación de formulario de encuesta
Descripción	Permitir a directores de carrera modificar formularios en estado 'Publicado'.
Actor/Usuario	Director(a) de carrera
Prioridad	Alta
Precondiciones	Formulario existente; usuario autenticado.
Entradas	Identificador del formulario, cambios solicitados en las preguntas o las opciones de tipo de respuesta
Proceso	Recuperar formulario, validar cambios, aplicar modificaciones y guardar en base de datos.
Salidas	Formulario actualizado con registro de auditoría.
Excepciones	Error de validación o formulario inexistente.
Criterios de Aceptación	Cambios aplicados correctamente, registro actualizado, confirmación visible.
Dependencias	RF-001

Nota: Detalle de las condiciones y validaciones necesarias para permitir la edición de encuestas existentes en el sistema.

Fuente: Elaboración propia

TABLA 5 *Requerimiento Funcional: Publicación de formulario para aplicación*

RF-003 - Publicación de formulario para aplicación	
Identificador	RF-003
Nombre	Publicación de formulario para aplicación
Descripción	Permite publicar el formulario y habilita la opción de responder en el Sistema de Graduados de la UAP en el menu “Encuesta Titulados”
Actor/Usuario	Director(a) de Carrera
Prioridad	Alta
Precondiciones	Formulario en estado 'Borrador' con preguntas completas.
Entradas	Identificador del formulario, fechas de inicio y cierre.
Proceso	Validar formulario, cambiar estado a 'Publicado' y registrar publicación.
Salidas	Formulario publicado.
Excepciones	Error al validar o guardar la encuesta.
Criterios de Aceptación	Acceso restringido de la encuesta solo a los titulados de la carrera, publicación confirmada.
Dependencias	RF-001, RF-002

Nota: Describe el proceso lógico para habilitar una encuesta y ponerla a disposición de los titulados.

Fuente: Elaboración propia

3.1.4.2 Requerimientos funcionales – Módulo Sistema de Recolección de Respuestas

En este apartado se especifican los requisitos para la interfaz de captura de datos orientada al usuario final (Titulado). Este módulo implementa una arquitectura de persistencia de estado, permitiendo el 'Guardado Parcial' de las respuestas para que el usuario pueda interrumpir y retomar el llenado en múltiples sesiones sin pérdida de información. Asimismo, se definen las reglas de validación en tiempo real (Frontend) y la integridad transaccional (Backend) necesarias para asegurar la calidad del dato antes de su consolidación definitiva.

TABLA 6 *Requerimiento Funcional - Completar formulario de encuesta*

RF-004 — Completar formulario de encuesta	
Identificador	RF-004
Nombre	Completar formulario de encuesta
Descripción	Permitir a titulados acceder y completar el formulario.
Actor/Usuario	Titulado(a) de carrera
Prioridad	Alta
Precondiciones	Formulario publicado y vigente.
Entradas	Respuestas a preguntas y datos sobre recorrido profesional.
Proceso	Verificar formulario activo, mostrar preguntas, validar respuestas, guardar en base de datos.
Salidas	Respuestas almacenadas y confirmación de recepción.
Excepciones	Formulario cerrado o respuestas incompletas.
Criterios de Aceptación	Una sola respuesta por usuario, guardado correcto y validación de campos.
Dependencias	RF-003

Nota: Especificaciones para la interfaz de llenado de encuestas por parte del usuario final.

Fuente: Elaboración propia

TABLA 7 *Requerimiento Funcional - Consulta de respuestas de formularios*

RF-005 — Consulta de respuestas de formularios	
Identificador	RF-005
Nombre	Consulta de respuestas de formularios
Descripción	Los directores de carrera acceden al listado de titulados, consultan el estado de respuesta de la encuesta publicada y analizan los resultados estadísticos generados por el sistema.
Actor/Usuario	Director de carrera
Prioridad	Alta
Precondiciones	Usuario autenticado y respuestas registradas.
Entradas	Identificador del formulario y filtros.
Proceso	Cargar respuestas, aplicar filtros y permitir exportación.

Salidas	Tabla de respuestas y archivo Excel/CSV exportado.
Excepciones	Error al exportar o sin resultados.
Criterios de Aceptación	Datos completos, filtros aplicados en menos de 2 segundos, exportación exitosa.
Dependencias	RF-004

Nota: Define los parámetros para que los administradores puedan visualizar los datos recolectados.

Fuente: Elaboración propia

3.1.4.3 Requerimientos funcionales – Módulo Vinculación Laboral (Bolsa De Trabajo)

Este apartado establece las especificaciones técnicas para el subsistema de integración externa. El módulo de Vinculación Laboral tiene como función principal la interoperabilidad asíncrona con el mercado laboral digital. Los requerimientos detallan el comportamiento del motor de extracción automatizada (Web Scraping), encargado de consolidar, normalizar y depurar ofertas provenientes de fuentes heterogéneas. El objetivo es transformar datos no estructurados de la web pública en información estructurada y buscable dentro de la plataforma institucional.

TABLA 8 *Requerimiento Funcional - Importación automatizada de ofertas de empleo mediante web scraping*

RF-006 — Importación automatizada de ofertas de empleo mediante web scraping	
Identificador	RF-006
Nombre	Importación automatizada de ofertas de empleo mediante web scraping
Descripción	Proceso que extrae ofertas de portales externos y las almacena en un archivo json, y se muestra información relevante.
Actor/Usuario	Sistema (proceso automatizado)
Prioridad	Alta
Precondiciones	Servicio de scraping activo y URLs configuradas.
Entradas	Configuración de scraping y datos HTML extraídos.
Proceso	Ejecutar scraping, extraer información, validar duplicados y guardar nuevas ofertas.
Salidas	Ofertas laborales registradas y log actualizado.
Excepciones	Fallo de conexión o estructura de portal cambiada.

Criterios de Aceptación	Ejecución automática, sin duplicados, importación completada en tiempo esperado.
Dependencias	RF-007

Nota: Ficha técnica del proceso automático de extracción de datos desde portales web externos.

Fuente: Elaboración propia

TABLA 9 *Requerimiento Funcional - Búsqueda y filtrado de ofertas de empleo*

RF-007 — Búsqueda y filtrado de ofertas de empleo	
Identificador	RF-007
Nombre	Búsqueda y filtrado de ofertas de empleo
Descripción	Permitir a titulados buscar ofertas publicadas mediante filtros.
Actor/Usuario	Titulado / Egresado
Prioridad	Alta
Precondiciones	Usuario autenticado y ofertas activas.
Entradas	Palabra clave, ubicación.
Proceso	Aplicar filtros y mostrar resultados ordenados.
Salidas	Lista de ofertas disponibles con detalles.
Excepciones	Sin resultados o error en búsqueda.
Criterios de Aceptación	Resultados en menos de 3 segundos, combinación de filtros efectiva.
Dependencias	RF-006

Nota: Describe las funcionalidades de búsqueda que permiten al usuario filtrar ofertas laborales.

Fuente: Elaboración propia

TABLA 10 *Requerimiento Funcional - Aplicación a oferta de empleo*

RF-008 — Aplicación a oferta de empleo	
Identificador	RF-008
Nombre	Aplicación a oferta de empleo
Descripción	Permitir a titulados acceder a la información completa de los trabajos listados en el sistema de Graduados UAP

Actor/Usuario	Titulado / Egresado
Prioridad	Alta
Precondiciones	Usuario autenticado
Entradas	Ninguna
Proceso	Hacer clic en Ver Oferta Completa
Salidas	Ninguna
Excepciones	Trabajo expirado
Criterios de Aceptación	Validaciones correctas, registro y notificación exitosos.
Dependencias	RF-007

Nota: Detalla el flujo de interacción para que un titulado se postule a una vacante disponible.

Fuente: Elaboración propia

TABLA 11 *Importar ofertas laborales desde fuentes externas (web scraping)*

RF – 009 - Importar ofertas laborales desde fuentes externas (web scraping)	
Identificador	RF-009
Nombre	Importar ofertas laborales desde fuentes externas (web scraping)
Descripción	Proceso automatizado que extrae ofertas de dos portales, evita duplicados y guarda en BD con marca de fuente y URL de origen.
Actor/Usuario	Sistema (proceso automatizado)
Precondiciones	Microservicio activo; URLs y selectores configurados; conexión a internet y BD.
Entradas	HTML/JS de los portales; cron de ejecución; selectores CSS.
Proceso	Scheduler - Puppeteer headless - extracción con selectores - validación de duplicados - inserción - logs - cierre.
Salidas	Registros nuevos en ofertas_laborales; logs de proceso; notificación de fallo crítico.
Excepciones	Portal no responde/estructura cambia - log y continuar con la otra fuente; BD caída → abortar y notificar.

Criterios de aceptación	Automático por cron; sin duplicados (por URL); visible como “Fuente Externa”; ≤ 30 min; redirección a URL original.
Dependencias	RF-007 (búsqueda y filtrado de ofertas)
Evidencia	Log de scraping y reporte de importaciones.

Nota: Especificación adicional sobre los parámetros de conexión y extracción de las fuentes externas de empleo.

Fuente: Elaboración propia

3.1.4.4 Requerimientos no funcionales – Seguridad

Este apartado define los estándares de protección de la información implementados para garantizar la confidencialidad e integridad de los datos académicos. Dado el diseño distribuido del sistema, los requerimientos no funcionales de seguridad se basan en una arquitectura de autenticación sin estado (Stateless). Se especifican los protocolos para la generación, firma y expiración de tokens de acceso (JWT), así como las políticas de encriptación de credenciales mediante algoritmos de hashing robustos (BCrypt), asegurando que ninguna contraseña se almacene en texto plano en la base de datos.

TABLA 12 *Requerimientos no funcionales – Seguridad*

ID	Categoría	Descripción	Criterios de aceptación	Evidencia
RNF-001	Autenticación	Autenticación con JWT (expira a 60 min) y refresh tokens con rotación segura; revocación inmediata al cerrar sesión.	1) Sin token/ inválido/ expirado- 401; rol insuficiente- 403. 2) Validación de token ≤ 10 ms por request.	Configuración JWT y pruebas de acceso.
RNF-002	Autorización	Control de acceso por roles: ADMIN, DIRECTOR DE CARRERA, TITULADO, SERVICIO (scraper).	1) TITULADO solo puede iniciar sesión en solo dispositivo.	Matriz de permisos por ruta.

RNF-003	Protección de datos	Uso de PosgreSQL en Servidor Físico de la Universidad Amazónica de Pando, implementación de Firewall Físico.	1) Información de titulados protegida en la Base de Datos del Sistema Integrado Siringuero.	Servidor Físico, Firewall Físico.
RNF-004	Scraping responsable	Scraping Paralelo identificado, ventanas de ejecución y pausas entre requests;	1) Pausas ≥ 10 s entre páginas. 2) Job fuera de horas pico.	Log del job de scraping. (json combinado de ambas fuentes)

Nota: Enumera las restricciones y estándares de seguridad aplicados para proteger la información del sistema.

Fuente: Elaboración propia

3.1.4.5 Requerimientos no funcionales – Usabilidad

Este apartado establece los criterios de calidad para la interacción humano-computadora, priorizando el paradigma Mobile-First. Los requerimientos de usabilidad dictan que la interfaz debe adaptarse fluidamente a cualquier resolución de pantalla (desde viewports móviles de 360px hasta monitores de escritorio), garantizando que los elementos táctiles (botones, selectores) tengan las dimensiones ergonómicas adecuadas para evitar errores de pulsación. El objetivo es eliminar las barreras de acceso tecnológico, permitiendo que el titulado complete el seguimiento desde su smartphone sin necesidad de zoom o desplazamientos horizontales.

TABLA 13 *Requerimientos no funcionales – Usabilidad*

ID	Categoría	Descripción	Criterios de aceptación	Evidencia
RNF-005	Compatibilidad	UI (React) soporta últimas 2 versiones de Chrome, Firefox, Edge; Safari con degradaciones no críticas documentadas.	1) Checklist cross-browser sin bloqueantes.	Planillas de verificación.

			2) Incidencias críticas = 0 antes del release.	
RNF-006	Diseño responsivo	Vistas principales responsive (320 px a 1440 px), sin scroll horizontal en móvil.	1) Rotación de pantalla refluye < 100 ms. 2) Sin desbordes de layout.	Capturas y reporte de pruebas.

Nota: Criterios de diseño para garantizar una experiencia de usuario intuitiva.

Fuente: Elaboración propia

3.1.4.6 Requerimientos no funcionales – Rendimiento

En este apartado se establecen las características operativas del despliegue. El sistema se define bajo una arquitectura Web-Based (Basada en Web), lo que implica la centralización de la lógica y los datos en los servidores de la universidad, eliminando la necesidad de instalación en el cliente. Los requisitos detallados a continuación garantizan la independencia de plataforma, asegurando que el servicio sea consumible desde cualquier sistema operativo o dispositivo que cuente con un navegador estándar compatible con HTML5.

TABLA 14 *Requerimientos no funcionales – Rendimiento*

ID	Categoría	Descripción	Criterios de aceptación	Evidencia
RNF-007	Estadísticas	Gráficas renderizan en ≤ 2 s con agregaciones o vistas materializadas.	1) Gráficas actualizan en ≤ 3 s.	Vista de Estadísticas de encuesta
RNF-008	Jobs/Scraping	Dos fuentes completan en ≤ 30 min; si no, partición y reintento con backoff.	1) No impacta API pública. 2) Log con importados, duplicados, errores.	Logs de job y métricas.

RNF-009	Base de datos	Índices en estado, área, fecha_publicación y particionado si respuestas > 1M.	1) Explain sin full scan. 2) p95 de consultas ≤ 300 ms.	Scripts de índices/partición.
RNF-010	Exportaciones	Exportes > 5 s se ejecutan asíncronos y notifican al usuario al completar.	1) Cola con estados. 2) Disponibilidad de archivos por 7 días.	Capturas de cola y descargas.

Nota: Métricas de velocidad y tiempos de respuesta esperados del sistema.

Fuente: Elaboración propia

3.1.4.7 Requerimientos no funcionales – Modalidad

Este apartado especifica la naturaleza operativa de la solución. El sistema se define bajo la modalidad de Aplicación Web (Web-based) de tipo 'Cliente Ligero' (Thin Client). Esto implica que los usuarios (titulados y administrativos) no requieren instalar ningún software adicional ni plugins propietarios en sus dispositivos. El acceso es universal a través de cualquier navegador web estándar (Chrome, Firefox, Edge, Safari), garantizando la disponibilidad del servicio tanto desde la Intranet universitaria como desde redes externas (Internet) las 24 horas del día.

TABLA 15 *Requerimientos no funcionales - Modalidad*

ID	Categoría	Descripción	Criterios de aceptación	Evidencia
RNF-011	Arquitectura	Capas claras (Controller, Service, Repository) en Spring; componentes/hook en React; scraper desacoplado (Node).	1) Sin dependencias cíclicas. 2) Bajo acoplamiento (métricas estáticas).	Diagramas y análisis estático.

RNF-012	Configuración/secretos	Config por entorno (dev/staging/prod) vía variables; secretos fuera del repo.	1) Revisión sin credenciales. 2) Templates con placeholders.	Auditoría de repositorio, servidor.
RNF-013	Migraciones/rollback	Migraciones versionadas (Flyway/Liquibase) con up/down y pruebas en staging.	1) Registro de migraciones. 2) Ensayo de rollback documentado.	Bitácora de cambios.
RNF-015	Observabilidad	Logs JSON con correlación; métricas	1) Alertas: p95>800 ms 10 min; 5xx>2% 5 min;	Control por consola y alertas.

Nota: Condiciones de operación y disponibilidad del software.

Fuente: Elaboración propia

3.1.5. Mapeo de historias de usuario

3.1.5.1 Módulo: Generador dinámico de formularios

Este módulo constituye el núcleo arquitectónico de la solución, diseñado para desacoplar la lógica de negocio de la estructura de datos. Funciona como un Motor de Renderizado basado en Metadatos, permitiendo al administrador construir, editar y desplegar instrumentos de recolección de datos (encuestas) de manera visual.

TABLA 16 *Historia de Usuario - Creación de encuesta*

ID	Historia de usuario
HU-001	Como director de carrera, quiero crear formularios de encuesta con tipos de preguntas (texto, opción única, múltiple, fecha, multiple con limite de selección) para recolectar información alineada a acreditación.
Criterios de aceptación	<ul style="list-style-type: none"> - Debe permitir al menos 7 tipos de preguntas configurables por campo obligatorio y orden. - Guardado en estado Borrador y Publicado.

- Previsualización antes de guardar. - Creación en ≤ 5 s con confirmación visible.

Puntos	8
Prioridad	Alta

Nota: Narrativa del usuario administrador para la generación de instrumentos de recolección.

Fuente: Elaboración propia

TABLA 17 *Historia de Usuario - Edición de Formularios*

ID	Historia de usuario
HU-002	Como director de carrera, quiero editar formularios Publicado para corregir o actualizar contenido sin perder consistencia.
Criterios de aceptación	- Si hay respuestas, mostrar advertencia y requerir confirmación. - Mantener historial de versiones con usuario y timestamp. - Actualización en ≤ 3 s.
Puntos	5
Prioridad	Alta

Nota: Requisitos desde la perspectiva del usuario para modificar encuestas.

Fuente: Elaboración propia

TABLA 18 *Historia de Usuario - Publicación y Programación de encuesta*

ID	Historia de usuario
HU-003	Como director de carrera, quiero publicar formularios para que los titulados respondan dentro de un período de vigencia.
Criterios de aceptación	- Activar/desactivar por fechas de inicio y cierre.
Puntos	5
Prioridad	Alta

Nota: Funcionalidad para definir la vigencia y visibilidad de los formularios.

Fuente: Elaboración propia

3.1.5.2 Módulo: Recolección de respuestas

Este módulo constituye la interfaz de interacción principal para el usuario Titulado. Funciona como un motor de renderizado en el cliente (Frontend), diseñado para interpretar la estructura de datos configurada previamente por el director de carrera.

Al ingresar al sistema y acceder al submenú 'Encuesta', el módulo consulta la API para recuperar el instrumento activo de la gestión. A diferencia de un formulario estático, la interfaz se construye dinámicamente en tiempo real, desplegando los campos (cajas de texto, selectores, opción múltiple) y aplicando las reglas de validación correspondientes. Esto permite que el titulado visualice únicamente las preguntas pertinentes a su perfil, garantizando una experiencia de usuario fluida y personalizada.

TABLA 19 *Historia de Usuario - Recolección de Información de Titulado*

ID	Historia de usuario
HU-004	Como Titulado, quiero responder un formulario publicado para enviar mis respuestas de forma segura y confirmada.
Criterios de aceptación	- Validar campos obligatorios y tipos (tipo texto, numérico). - Prevenir respuestas duplicadas por usuario. - Confirmación de envío en ≤ 5 s. - Compatible móvil/desktop.
Puntos	8
Prioridad	Alta

Nota: Experiencia del titulado al ingresar sus datos en el sistema.

Fuente: Elaboración propia

3.1.5.3 Módulo: Vinculación laboral (bolsa de trabajo)

Este módulo implementa una arquitectura de interoperabilidad basada en extracción de datos (Web Scraping). Funciona mediante un microservicio autónomo (desarrollado en Node.js) que actúa como un agente de búsqueda automatizado. Su función es conectarse periódicamente a portales externos de alta relevancia (Trabajando.com.bo, Trabajopolis), extraer las ofertas laborales mediante selectores DOM, normalizar la información (estandarizar títulos, fechas y empresas) y consolidarla en un repositorio centralizado. Esto permite al sistema ofrecer una 'Bolsa de Trabajo Unificada' sin depender de APIs públicas costosas o inexistentes.

TABLA 20 *Historia de usuario - Búsqueda y filtrado de ofertas laborales*

ID	Historia de usuario
HU-005	Como Titulado, quiero buscar y filtrar ofertas por palabra clave, ubicación.

Criterios de aceptación	- Combinación de filtros simultáneos. - Resultados con paginación (20 ítems). - Orden por más recientes. - Tiempo de respuesta \leq 500 ms p95.
Puntos	8
Prioridad	Alta

Nota: Interacciones del usuario para localizar oportunidades de trabajo.

Fuente: Elaboración propia

TABLA 21 *Historia de Usuario - Postulación a oferta laboral*

ID	Historia de usuario
HU-006	Como Titulado, quiero postular a una oferta enviando mi perfil y CV para ser considerado por la empresa.
Criterios de aceptación	- Validar perfil \geq 80% completado. - Evitar visualización duplicada de la misma oferta.
Puntos	5
Prioridad	Alta

Nota: Pasos que sigue el usuario para aplicar a una oferta seleccionada.

Fuente: Elaboración propia

3.1.5.4 Módulo: Estadísticas y Seguimiento a respuestas

Este módulo constituye la capa de Inteligencia de Negocios (Business Intelligence) del sistema. Funciona como un Tablero de Control (Dashboard) Administrativo que procesa en tiempo real la información recolectada, transformando los datos brutos en indicadores visuales comprensibles (Gráficos de barras, tortas y tendencias). Su objetivo es automatizar la generación de estadísticas para los informes de acreditación, eliminando el error humano asociado a la tabulación manual. Además, permite el seguimiento nominal, identificando individualmente a los titulados que no han completado la encuesta mediante un sistema de estados.

TABLA 22 *Historia de Usuario - Visualización y Seguimiento de estado de Respuestas de Titulados*

ID	Historia de usuario
HU-007	Como director de carrera, quiero acceder en el sistema a un panel donde pueda visualizar estadísticas generadas a partir de las respuestas de los titulados para todas las preguntas de tipo cerrada, presentadas en gráficos y tablas, a fin de respaldar la toma de decisiones académicas.

Criterios de aceptación	- KPIs: gráficas por pregunta, secciones de formulario, satisfacción (1–5). - Render inicial ≤ 2 s; filtros ≤ 3 s.
Puntos	8
Prioridad	Alta

Nota: Herramientas de monitoreo para el control del avance en las encuestas.

Fuente: Elaboración propia

3.1.5.5 Módulo: Integración externa (scraping)

Este módulo actúa como una capa de interoperabilidad técnica entre el sistema institucional y el ecosistema laboral externo. Funciona como un motor de Extracción, Transformación y Carga (ETL) automatizado.

- 1. Extracción:** Un microservicio autónomo (Node.js) navega simuladamente por portales de empleo seleccionados.
- 2. Transformación:** El motor analiza la estructura DOM (HTML) no estructurada, limpia el ruido (publicidad), normaliza los formatos (fechas, ubicación) y detecta duplicados.
- 3. Carga:** Finalmente, inyecta los datos limpios y estructurados en la base de datos local, permitiendo que las ofertas sean consultables internamente sin depender de la disponibilidad de la fuente original.

TABLA 23 *Historia de Usuario - Importación de ofertas mediante Scraping Automatizado*

ID	Historia de usuario
HU-008	Como Sistema (servicio programado), quiero importar ofertas desde dos portales externos para enriquecer la bolsa de trabajo.
Criterios de aceptación	- Evitar duplicados por URL de origen. - Etiquetar “Fuente Externa” y mantener link original. - Job total ≤ 30 min; logs con importados/duplicados/errores.
Puntos	13
Prioridad	Media-Alta

Nota: Ejecución técnica del proceso de obtención de ofertas externas.

Fuente: Elaboración propia

3.1.5.6 Módulo: Gestión de formularios y ciclo de vida

Este módulo administra la gobernanza de los instrumentos, implementando un control estricto del ciclo de vida mediante un flujo de estados definidos: Borrador (Edición permitida), Publicado (Visible para titulados, estructura bloqueada) y Cerrado (Solo para lectura de las estadísticas).

Su característica técnica más relevante es la implementación de la Baja Lógica (Soft Delete). A diferencia de un borrado físico convencional, este mecanismo simplemente cambia el estado de visibilidad del registro en la base de datos, garantizando la inmutabilidad de los datos históricos. Esto asegura que, aunque se elimine una encuesta de la vista administrativa, las respuestas asociadas de gestiones pasadas permanezcan intactas para fines estadísticos.

TABLA 24 *Historia de Usuario - Gestión de Encuesta Publicada*

ID	Historia de usuario
HU-009	Como director de carrera, quiero poner fecha límite a una encuesta publicada para detener nuevas respuestas y generar un corte de análisis.
Criterios de aceptación	- Cambio de estado a Cerrado con timestamp y usuario. - Bloqueo de nuevas respuestas desde el cierre. - Posibilidad de reabrir con ampliación de límite fecha de llenado en la encuesta.
Puntos	3
Prioridad	Media

Nota: Opciones administrativas para el manejo de encuestas activas.

Fuente: Elaboración propia

3.1.6. Casos de uso del sistema

Para formalizar las interacciones funcionales, se identificaron y modelaron los actores que intervienen en el sistema, clasificándolos según su naturaleza (humanos y sistemas externos) y su nivel de privilegio.

A. Actores Humanos (Usuarios)

1. Director de Carrera (Actor Primario)

Rol: Es el usuario estratégico del sistema.

Responsabilidades: Tiene privilegios para diseñar y publicar los instrumentos de seguimiento (encuestas), gestionar el ciclo de vida de los formularios y visualizar el Tablero de Control Estadístico para la toma de decisiones. Es el único autorizado para crear encuestas y formularios.

2. Titulado (Actor Primario)

Rol: Es el usuario final y principal fuente de información.

Responsabilidades: Interactúa con la interfaz pública para completar las encuestas asignadas y mantener actualizados sus datos de contacto. Además, es el consumidor directo del Módulo de Vinculación Laboral, pudiendo consultar y filtrar las ofertas de empleo centralizadas.

3. Administrador de TI (Actor Secundario)

Rol: responsable del mantenimiento técnico.

Responsabilidades: Se encarga de la gestión de usuarios (altas/bajas de directores), monitoreo de los servicios (Backend/Database) y verificación de los logs de sincronización. No interviene en la definición de las preguntas de la encuesta.

B. Actores de Sistema (Sistemas Externos)

4. Agente de Scraping (Actor Sistema)

Rol: Servicio autónomo (Bot).

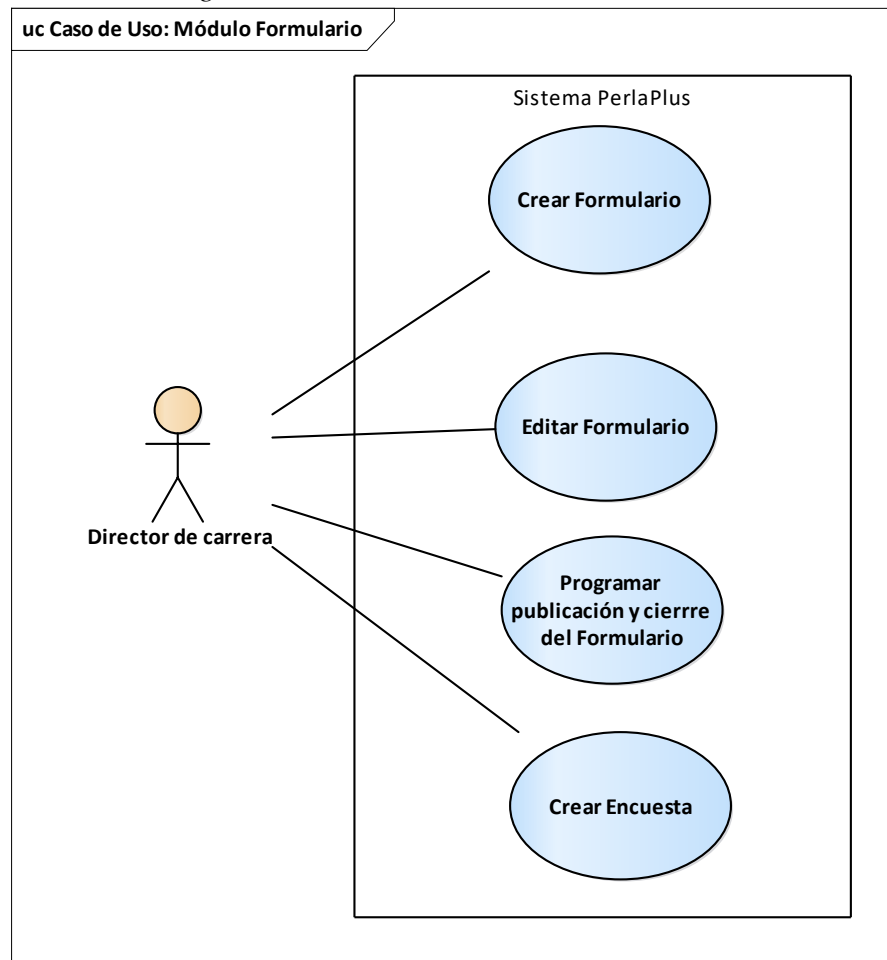
Responsabilidades: Actúa como un actor temporal que ejecuta tareas programadas. Su función es navegar por los portales externos, extraer ofertas laborales e inyectarlas en la base de datos local sin intervención humana.

5. Sistema Académico Siringuero (Actor Sistema)

Rol: Fuente de Verdad Institucional.

Responsabilidades: Sistema legado de la UAP que provee, mediante interoperabilidad, la nómina oficial de graduados. El sistema de seguimiento consulta a este actor para validar la identidad de los titulados y evitar registros fraudulentos.

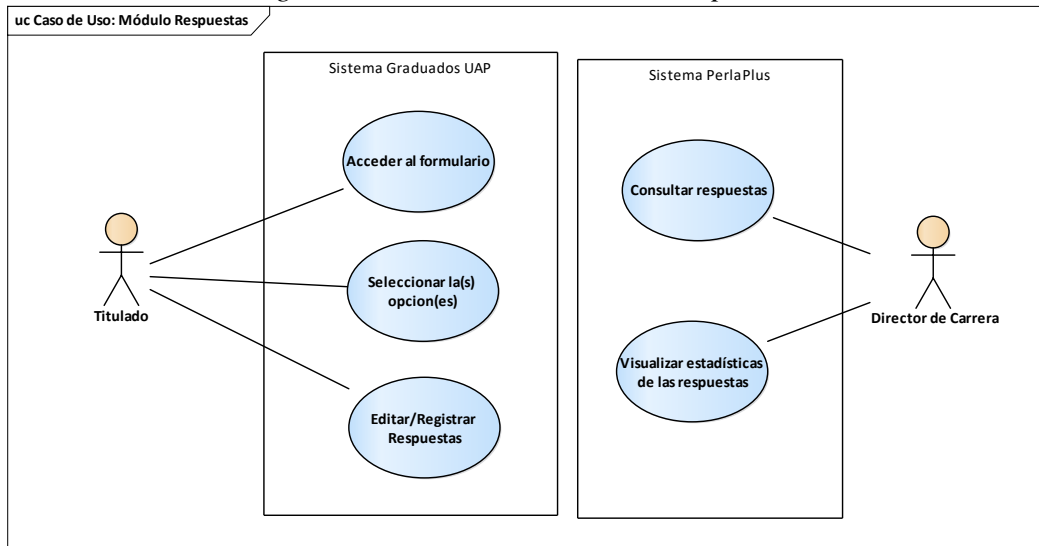
Figura 1 Caso De Uso: Módulo Formulario



Nota: Diagrama que ilustra las interacciones del administrador con las funciones de creación, edición y gestión de encuestas dinámicas.

Fuente: Elaboración propia

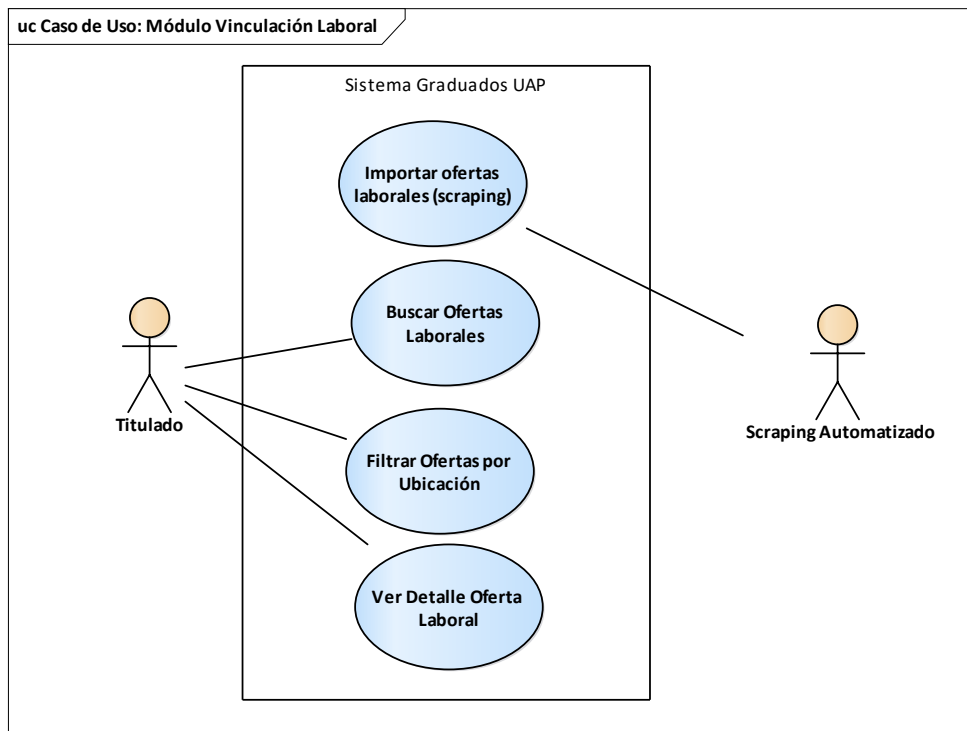
Figura 2 Caso de Uso - Módulo Respuestas



Nota: Representación de los procesos que permiten al titulado completar la encuesta y al administrador visualizar los resultados.

Fuente: Elaboración propia

Figura 3 Caso de Uso-Módulo Vinculación Laboral



Nota: Esquema de las funciones de búsqueda, filtrado y postulación a ofertas de empleo por parte del usuario.

Fuente: Elaboración propia

3.1.7. Matriz de Roles y Privilegios (Modelo RBAC)

TABLA 25 *Matriz de Roles y Privilegios (Modelo RBAC)*

Módulo / Funcionalidad	Director de Carrera	Titulado (Graduado)	Scraper (Bot)
Gestión de Usuarios	Lectura (Ver lista)	Solo su Perfil	-
Diseño de Encuestas	TOTAL (Crear/Editar)	Lectura (Responder)	-
Gestión de Formularios	TOTAL (Crear/Editar)	-	-
Gestión de Preguntas	TOTAL (Crear/Editar)	-	-
Bolsa de Trabajo	-	Lectura y Filtrado	Escritura (Importar e Insertar ofertas)
Respuestas de Titulados	Lectura (Estadísticas)	Escritura (Su propia respuesta)	-
Seguimiento de Respuestas	Lectura (Listado de titulados y estado de respuesta)	-	-
Gráficos Estadísticos	Lectura (Gráficos)	-	-

Nota: Cuadro de doble entrada que asigna permisos específicos a cada tipo de usuario del sistema.

Fuente: Elaboración propia

3.2. FASE GAME – Desarrollo Iterativo

Esta fase constituye la etapa de construcción y ejecución técnica del proyecto. Se caracteriza por la implementación del ciclo de vida iterativo, donde los requisitos priorizados en el Product Backlog son transformados sistemáticamente en incrementos de software funcional.

Durante esta etapa, se ejecutaron los Sprints de Desarrollo (del 1 al 12), aplicando prácticas de codificación bajo estándares de arquitectura limpia (Clean Architecture) en el Backend (Spring

Boot) y diseño modular en el Frontend (React). Cada iteración integró actividades de análisis, diseño técnico, codificación y pruebas unitarias, asegurando que cada entrega aporte valor tangible a la Dirección de Información Académica.

3.2.1. Planificación General de sprints

TABLA 26 *Planificación General de Sprints*

Sprint	Duración	Bloque	Objetivo Principal	HU Asignadas	Story Points	Velocidad Acumulada
1	2 semanas	Bloque 1	Diseño de arquitectura general	-	0*	5
3	2 semanas	Bloque 2	Creación y edición de formularios	HU-001, HU-002	13	26
4	2 semanas	Bloque 2	Publicación y gestión de vigencia	HU-003	8	34
5	2 semanas	Bloque 2	Recolección de respuestas	HU-004	8	42
6	2 semanas	Bloque 2	Panel de estadísticas	HU-007	8	50
7	2 semanas	Bloque 2	Búsqueda de ofertas laborales	HU-005	8	58
8	2 semanas	Bloque 2	Postulaciones	HU-006	10	68
10	2 semanas	Bloque 3	Web scraping de ofertas (parte 1)	HU-008 (parcial)	7	75
11	2 semanas	Bloque 3	Web scraping de ofertas (parte 2)	HU-008 (parcial)	6	81
12	2 semanas	Bloque 3	Integración, pruebas y optimización	-	0*	81

Nota: Cronograma de iteraciones y objetivos planteados para el desarrollo.

Fuente: Elaboración propia

3.2.2. Desarrollo e Implementación de Módulos

- **Módulo 1: Generador Dinámico de Formularios (Sprints 4-5)**

Este módulo representa el núcleo del sistema. Se implementó la lógica para mapear relaciones complejas de base de datos (@OneToMany) hacia una interfaz visual, permitiendo la creación de estructuras jerárquicas (Encuesta, Sección, Pregunta).

- Historias Implementadas: HU-001, HU-002, HU-003
- Arquitectura de Componentes:

TABLA 27 Capas de componentes claves desarrollados-Generador Dinámico de Formularios

Capa	Componentes Clave Desarrollados	Función Técnica
Backend (API)	FormularioController, SeccionController	Exposición de Endpoints REST para el CRUD transaccional.
Persistencia	FormularioRepository, PreguntaRepository	Gestión de consultas JPA y lógica de "Soft Delete".
Frontend (UI)	CrearFormulario.jsx, SelectorTipoPregunta.jsx	Interfaz dinámica que renderiza inputs según la selección del admin.

Nota: Estructura técnica del módulo de creación de encuestas.

Fuente: Elaboración propia

Resultado del Incremento: Motor de encuestas operativo capaz de gestionar múltiples versiones de instrumentos con lógica de validación íntegra.

- **Módulo 2: Sistema de Recolección de Respuestas (Sprint 6)**

El foco de este sprint fue la persistencia y la experiencia de usuario. Se desarrolló el mecanismo de guardado transaccional para asegurar que las respuestas de los titulados se almacenen con integridad referencial, evitando pérdida de datos.

- Historia Implementada: HU-004.
- Arquitectura de Componentes:

TABLA 28 Capas de componentes desarrollados- Sistema de Recolección de Respuestas

Capa	Componentes Desarrollados	Clave	Función Técnica
Backend (API)	RespuestaService, AntiDuplicationGuard		Lógica de negocio para validar campos obligatorios y bloquear envíos dobles.
Frontend (UI)	ResponderFormulario.jsx, WizardStep.jsx		Renderizado del formulario "paso a paso" con validación visual.

Nota: Componentes de software encargados de capturar los datos del usuario.

Fuente: Elaboración propia

Resultado del Incremento: Interfaz de usuario final optimizada para móviles, con confirmación de envío y bloqueo de duplicidad por ID de usuario.

Módulo 3: Estadísticas y Tablero de Control (Sprint 7)

Se implementó la capa de inteligencia de negocios. Se optimizaron las consultas SQL nativas en los repositorios para realizar conteos y agrupaciones (GROUP BY) directamente en la base de datos, garantizando velocidad en la generación de gráficos.

- **Historia Implementada:** HU-007.
- **Arquitectura de Componentes:**

TABLA 29 Capas de Componentes desarrollados - Estadística y tablero de control

Capa	Componentes Desarrollados	Clave	Función Técnica
Backend (API)	EstadisticaService, ReporteDTO		Agregación de datos y transformación a formato JSON para gráficos.
Frontend (UI)	Dashboard.jsx, Chart.js	Librería	Visualización interactiva con filtros dinámicos (Por gestión/carrera).

Nota: Resultado del Incremento: Dashboard administrativo capaz de renderizar estadísticas en tiempo real (< 2 segundos).

Fuente: Elaboración propia

D. Módulo 4: Vinculación Laboral - Buscador (Sprints 8-9)

En esta etapa se construyó la interfaz de consulta para los titulados. Se desarrollaron algoritmos de filtrado en el Backend para permitir búsquedas multicriterio (palabra clave + ubicación) sobre la base de datos de ofertas.

- **Historias Implementadas:** HU-005, HU-006.
- **Arquitectura de Componentes:**

TABLA 30 Capas de componentes desarrollados - Vinculación Laboral

Capa	Componentes Desarrollados	Clave	Función Técnica
Backend (API)	OfertaLaboralController, SpecificationAPI		API de búsqueda avanzada utilizando JPA Specifications para filtros dinámicos.
Frontend (UI)	BuscarOfertas.jsx, DetalleOferta.jsx		Grid de resultados con paginación y redirección a fuentes externas.

Nota: Arquitectura del módulo de conexión con el mercado laboral.

Fuente: Elaboración propia

Resultado del Incremento: Buscador de empleos centralizado funcional, con filtros de alta precisión.

Módulo 5: Motor de Web Scraping (Sprints 10-11)

Se integró el microservicio independiente encargado de la interoperabilidad. A diferencia del núcleo en Java, este componente se desarrolló en **Node.js** para aprovechar su manejo asíncrono del DOM.

- **Historia Implementada:** HU-008.
- **Arquitectura del Microservicio:**

TABLA 31 Componentes según la tecnología desarrollada- Motor Web Scraping

Componente	Tecnología	Función Técnica
Motor de Extracción	Puppeteer (Node.js)	Navegación simulada (Headless Browser) para extracción de datos en Trabajando.com.bo/Trabajopolis.
Gestor de Tareas	Node-Cron	Programación de la ejecución automática (Daily Job: 02:00 AM).

Normalizador	DataCleaner.js	Algoritmo de limpieza y detección de duplicados (Hash check) antes de insertar en BD, json.
---------------------	----------------	---

Nota: Tecnologías empleadas para el motor de extracción de datos.

Fuente: Elaboración propia

Resultado del Incremento: Robot de software autónomo que inyecta un promedio de 1250 ofertas diarias al sistema sin intervención humana.

Sprint 12: Integración Final y Hardening

El último ciclo se dedicó a la estabilización del sistema ("Hardening"). Se realizaron pruebas integrales (*End-to-End*) unificando los módulos de Java y Node.js.

- **Optimización:** Se crearon índices en PostgreSQL (CREATE INDEX) para acelerar las consultas de reportes.
- **Corrección de Errores:** Resolución de 8 *bugs* menores detectados en la validación de formularios móviles.
- **Validación de Rendimiento:** Pruebas preliminares para asegurar el cumplimiento del tiempo de respuesta (RNF-009).

3.2.2.1. Arquitectura del Sistema Implementada

TABLA 32 *Arquitectura del Sistema Implementada*

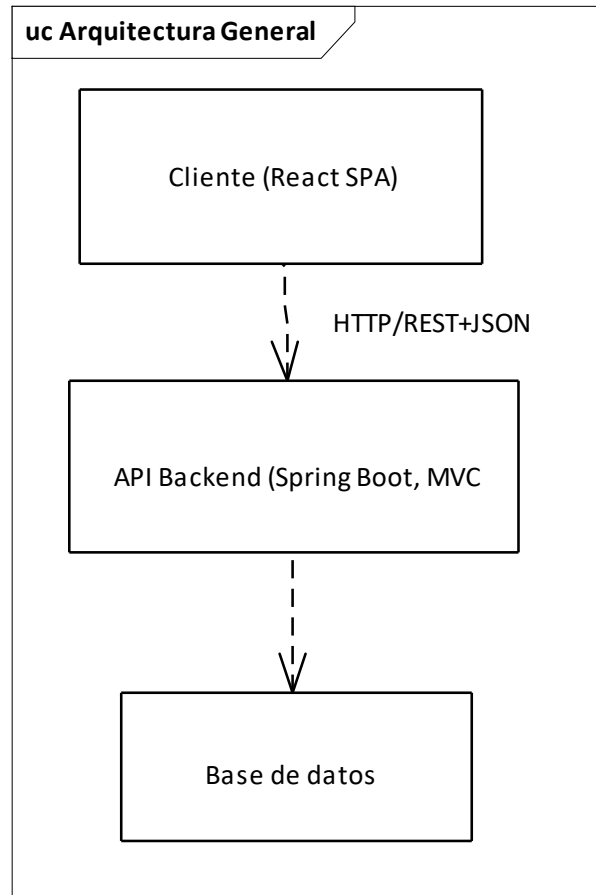
Componente	Descripción
Tipo	Arquitectura Web Cliente-Servidor, Frontend-Backend claramente separados
Frontend	Single Page Application (SPA) desarrollado con React
Backend	API REST desarrollada con Spring Boot
Comunicación	HTTP/REST con intercambio de datos en formato JSON
Patrón	Backend: MVC (Modelo-Vista-Controlador); Frontend: Component-Based (React)

Nota: Visión global de la estructura tecnológica y sus interacciones.

Fuente: Elaboración propia

Figura 4 Arquitectura General Implementada

Fuente: Elaboración propia



Nota: Vista de alto nivel que muestra la integración entre el cliente web, el servidor de aplicaciones y los servicios externos de scraping.

a. Arquitectura Frontend (React)

El frontend corresponde a una arquitectura Single Page Application (SPA) desarrollada en React 18, con enrutamiento del lado del cliente para navegación sin recargas completas de página. El empaquetado y servidor de desarrollo se gestionan con Vite 5 empleando el plugin `@vitejs/plugin-react-swc`, optimizando la compilación de JSX y la experiencia de desarrollo en entorno ESM.

Stack y versiones

La aplicación está configurada como módulo ECMAScript (`"type": "module"`) y define scripts estandarizados para desarrollo, compilación, previsualización y análisis estático,

asegurando reproducibilidad del entorno de trabajo. A continuación, se documentan las versiones declaradas:

TABLA 33 Componentes necesarios para el Proyecto en el Front End (React)

Componente	Versión
React	^18.3.1
React DOM	^18.3.1
React Router DOM	^6.27.0
Vite	^5.4.1
@vitejs/plugin-react-swc	^3.5.0
Bootstrap	^5.3.3
React-Bootstrap	^2.10.5
React Hot Toast	^2.4.1
React Icons	^5.5.0
@react-oauth/google	^0.12.1
ESLint	^9.9.0

Nota: Librerías y recursos requeridos para la interfaz de usuario.

Fuente: Elaboración propia

Estructura de src y roles

A continuación, se detalla la arquitectura de directorios implementada en el cliente web (**Frontend**). Para garantizar la mantenibilidad y escalabilidad del proyecto, se adoptó una estructura modular basada en componentes.

Como se observa en la **Tabla 25**, la organización del código fuente (src) sigue el principio de Separación de Responsabilidades:

- **Vistas y Enrutamiento:** Las páginas completas (/pages) se separan de la lógica de navegación (App.jsx).

- **Reutilización:** Los elementos visuales repetitivos (botones, modales) se abstraen en la carpeta /components, permitiendo su reutilización en múltiples vistas sin duplicar código.
- **Gestión de Estado Global:** Se implementó el patrón Provider mediante la Context API de React en /context, centralizando la lógica de seguridad (Tokens JWT) para que sea accesible desde cualquier parte de la aplicación sin necesidad de propalar propiedades (Prop Drilling).

TABLA 34 *Arquitectura de Software Modular en React*

Carpeta/Archivo	Rol en la app	Detalles clave
src/pages	Vistas de dominio	Páginas enrutables como Encuesta y OfertasLaborales, consumidas por el enrutador principal.
src/sections	Layout persistente	Contiene Header y Sidebar que envuelven el contenido de las rutas protegidas, separando layout de contenido.
src/components	Componentes reutilizables	Piezas transversales como ModalLoad y elementos UI reutilizables en múltiples páginas.
src/context	Estado global	Proveedor y hook de autenticación (UsuarioProvider/useUsuario) accesibles en toda la app para control de flujo por token.
src/assets	Recursos y estilos	Estilos globales y recursos compartidos consumidos desde main.jsx y App.jsx (por ejemplo, index.css, App.css).
src/App.jsx	Enrutamiento y layout	Declara Routes/Route, controla acceso por token y monta el layout persistente (Header, Sidebar).
src/main.jsx	Bootstrap de la app	createRoot sobre #root y jerarquía de proveedores: BrowserRouter, UsuarioProvider, GoogleOAuthProvider y Toaster.

Nota: Organización del código y componentes en el cliente web.

Fuente: Elaboración propia

Componentes y servicios del frontend

En este apartado se detallan los pilares técnicos que sostienen la interfaz de usuario. La aplicación se construyó bajo el **paradigma de Single Page Application (SPA)**, lo que implica

que el navegador carga un único documento HTML y actualiza el contenido dinámicamente sin recargar la página completa.

Como se describe en la tabla, la arquitectura se basa en cuatro servicios transversales:

1. **Componentización:** Se aplicó una jerarquía estricta donde los *Layouts* envuelven a las *Vistas*, y estas a su vez reutilizan *Componentes* atómicos, facilitando el mantenimiento.
2. **Gestión de Estado (Context API):** Se centralizó la información crítica (como el Token de sesión y datos del usuario) para que esté disponible en todo el árbol de componentes sin redundancia.
3. **Enrutamiento Declarativo:** Mediante React Router, se gestiona la navegación interna, permitiendo una experiencia de usuario fluida e instantánea.
4. **Capa de Comunicación:** Se implementó un servicio cliente HTTP para consumir la API REST del Backend de forma asíncrona."

TABLA 35 *Componentes y servicios del frontend*

Elemento	Descripción
Estructura de Componentes	Jerarquía: Layout - Vistas (páginas) - Componentes reutilizables
Gestión de estado	Context API
Enrutamiento	React Router para navegación SPA
Servicios de comunicación	Axios o Fetch para llamadas HTTP con backend (API REST, JSON)

Nota: Detalle de los servicios integrados en la aplicación visual.

Fuente: Elaboración propia

b. Arquitectura Backend (SpringBoot)

El backend implementa una arquitectura por capas con Spring Boot, exponiendo endpoints HTTP mediante Spring MVC, con configuración global de CORS y capacidad de empaquetado para contenedores de servlets a través de SpringBootServletInitializer, gestionado con Maven como sistema de construcción.

La clase principal `ApiSistituladosApplication` y el inicializador de servlet confirman el arranque de Spring Boot y la opción de despliegue como JAR, en concordancia con las guías de Spring Boot 3.x

TABLA 36 *Arquitectura Backend (SpringBoot)*

Sección	Descripción
Construcción y despliegue	El proyecto usa Maven (<code>pom.xml</code>) para gestionar dependencias, plugins y el ciclo de build, generando artefactos reproducibles alineados con Spring Boot. La clase <code>ServletInitializer</code> extiende <code>SpringBootServletInitializer</code> para habilitar empaquetado WAR en contenedores externos además del JAR ejecutable, brindando flexibilidad de despliegue según el entorno.
Punto de arranque	<code>ApiSistituladosApplication</code> define <code>main</code> e invoca <code>SpringApplication.run</code> para inicializar el contexto y publicar el backend como punto de entrada. En la misma clase se declara un Bean <code>WebMvcConfigurer</code> para centralizar la configuración global de CORS aplicada a todos los controladores y rutas.
Capa web y CORS	La política CORS global se aplica mediante <code>addCorsMappings</code> , definiendo <code>allowedOrigins</code> y <code>allowedMethods</code> para permitir el consumo desde el frontend en desarrollo y facilitar pruebas de integración cliente-servidor. Al registrarse como configuración Web MVC al inicio, la política impacta de forma uniforme a cualquier controlador REST del proyecto.
Paquetes y responsabilidades	Estructura por capas típica de Spring: <code>controllers</code> (orquestración HTTP), <code>services</code> (lógica de negocio), <code>dao/repository</code> (acceso a datos), <code>entity</code> (modelo persistente), y <code>DTOs request/response</code> para entrada/salida de la API. Módulos transversales: <code>config</code> y <code>security</code> para beans, filtros y seguridad, y <code>utils</code> para utilidades comunes, promoviendo bajo acoplamiento y alta cohesión.
Evidencias en el repositorio	<code>ApiSistituladosApplication.java</code> : clase principal, arranque de Spring y CORS global vía <code>WebMvcConfigurer</code> . <code>ServletInitializer.java</code> : soporte WAR extendiendo <code>SpringBootServletInitializer</code> para despliegue en contenedor de servlets. <code>pom.xml</code> : configuración Maven de dependencias y plugins coherente con Spring Boot. HELP.md : referencias a Spring Web y Spring Data JPA para Spring Boot 3.x como guía del proyecto.

Nota: Esquema del servidor y la lógica de negocio en Java.

Fuente: Elaboración propia

TABLA 37 Estructura de Paquetes y Responsabilidades del Backend

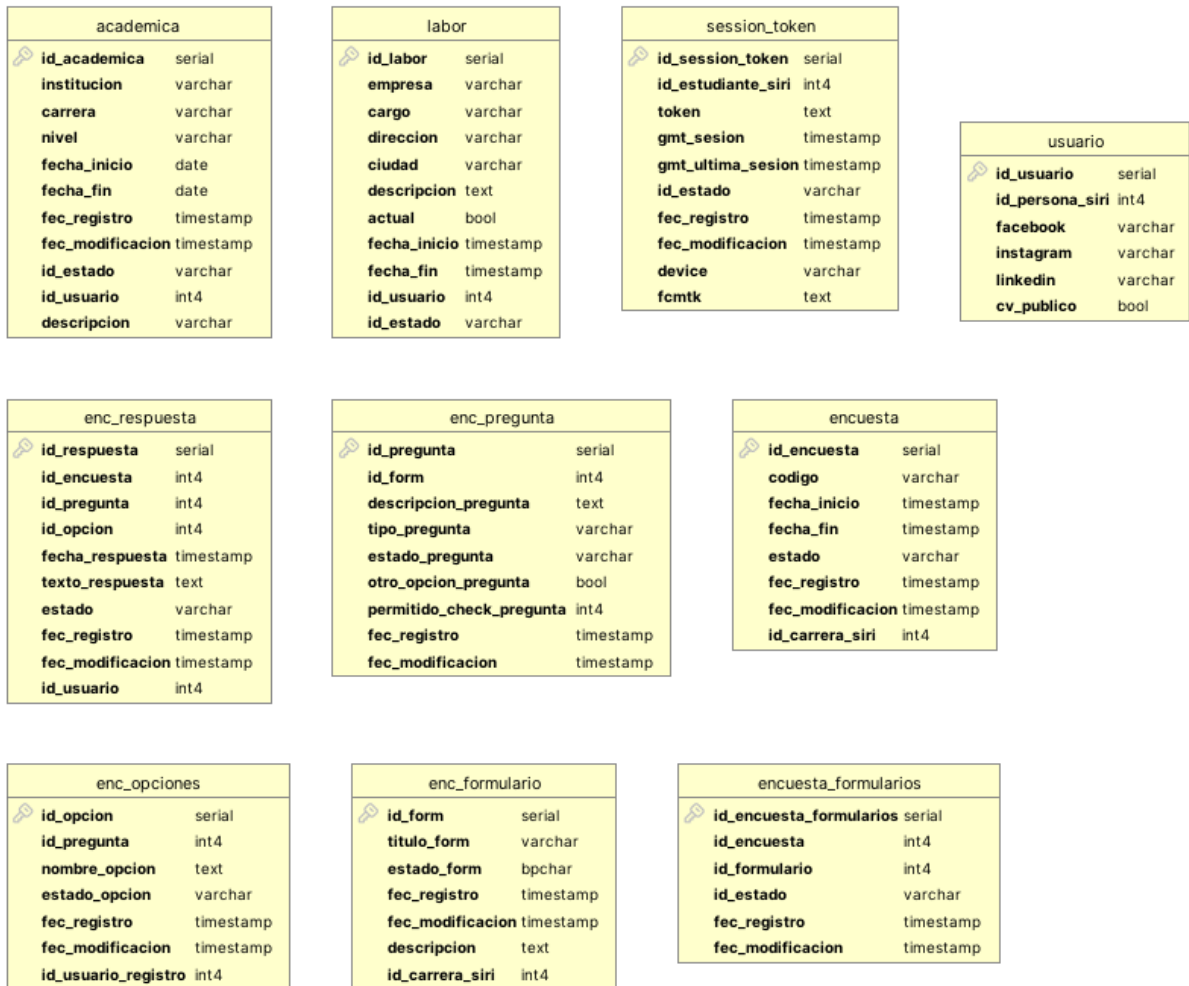
Capa Lógica	Paquete (Package)	Descripción Técnica y Responsabilidad
Presentación (API)	...controllers	Capa expuesta al cliente (React). Contiene las clases anotadas con @RestController que gestionan las peticiones HTTP (GET, POST, PUT), validan los datos de entrada y retornan códigos de estado estandarizados.
Lógica de Negocio	...services	El "cerebro" del sistema. Aquí residen las reglas de negocio (ej. validaciones de encuestas, lógica de <i>scraping</i>). Utiliza la anotación @Service y define los límites transaccionales (@Transactional) para asegurar la integridad de datos.
Persistencia (Datos)	...dao (Repositorios) ...entity (Modelos)	Implementa el patrón DAO/Repository. Las entidades mapean las tablas de PostgreSQL mediante JPA/Hibernate, mientras que los repositorios (JpaRepository) abstraen las consultas SQL.
Seguridad	...security	Contiene los filtros de interceptación de peticiones, la configuración de Spring Security y la lógica de generación/validación de Tokens JWT para la autenticación <i>Stateless</i> .
Transferencia (DTO)	...request ...responses ...objects	Implementa el patrón DTO (Data Transfer Object). Sirve para desacoplar las entidades de la base de datos de los datos que se envían al Frontend, evitando exponer información sensible (como contraseñas o IDs internos).
Integración Externa	...siringuero	Capa de Adaptación (Anti-Corruption Layer). Contiene la lógica específica para conectarse, consultar e importar datos desde la base de datos heredada del sistema académico oficial de la universidad.

Nota: Organización de carpetas y clases en el proyecto del servidor.

Fuente: Elaboración propia

3.2.2.2. Modelo de Datos

Figura 5 Modelo de Datos

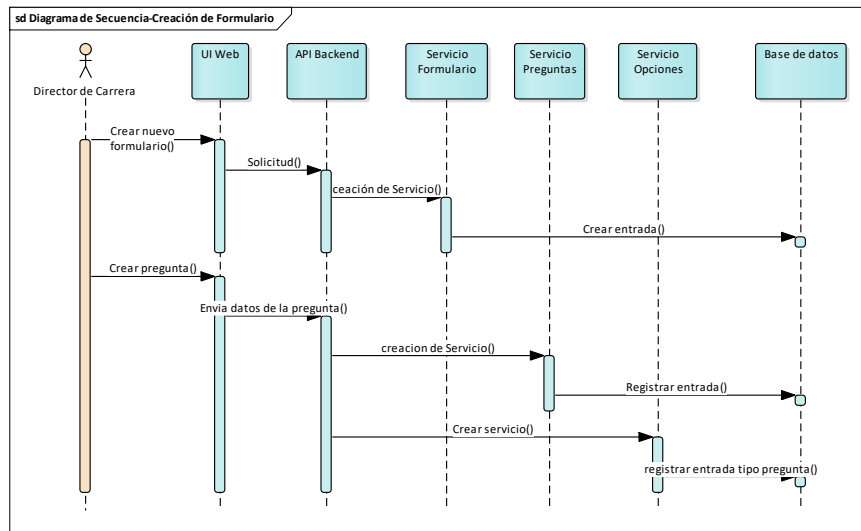


Fuente: Nota: Estructura de las clases principales en Spring Boot, organizadas en controladores, servicios y repositorios.

Fuente: Elaboración propia

b) Diagramas de Secuencia

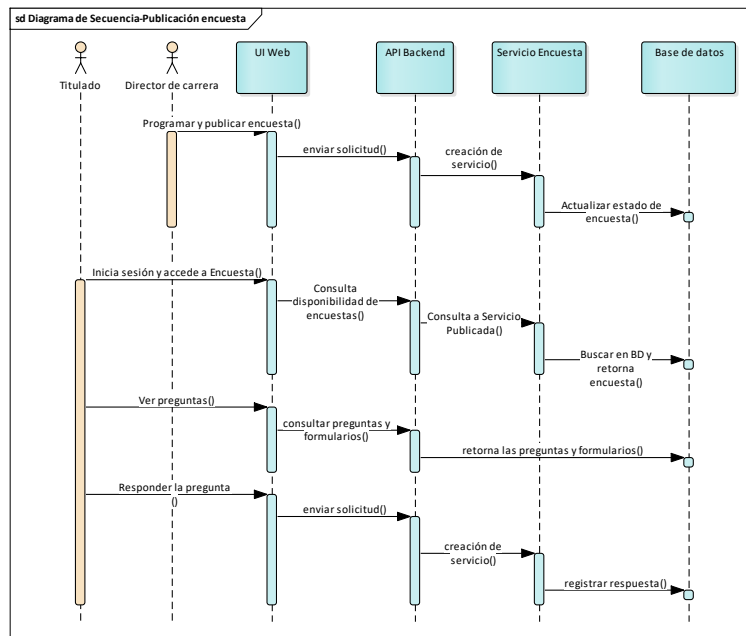
Figura 7 Diagrama de Secuencia - Creación de Formulario



Nota: Flujo temporal de mensajes entre objetos durante el proceso de diseño y almacenamiento de una nueva encuesta.

Fuente: Elaboración propia

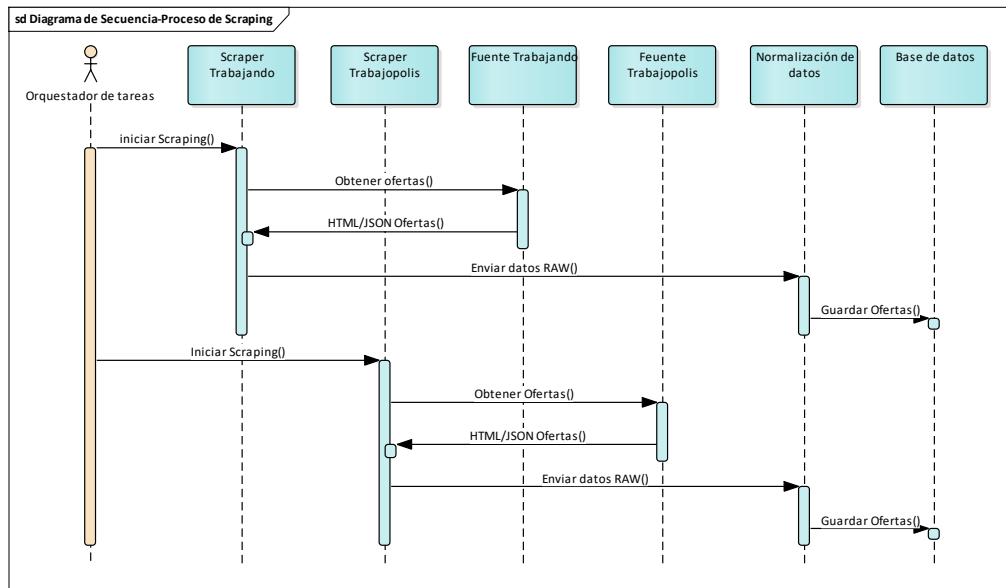
Figura 8 Diagrama de Secuencia - Publicación de encuesta y Registro de respuestas



Nota: Detalle de la interacción entre los componentes del sistema cuando un usuario envía sus respuestas.

Fuente: Elaboración propia

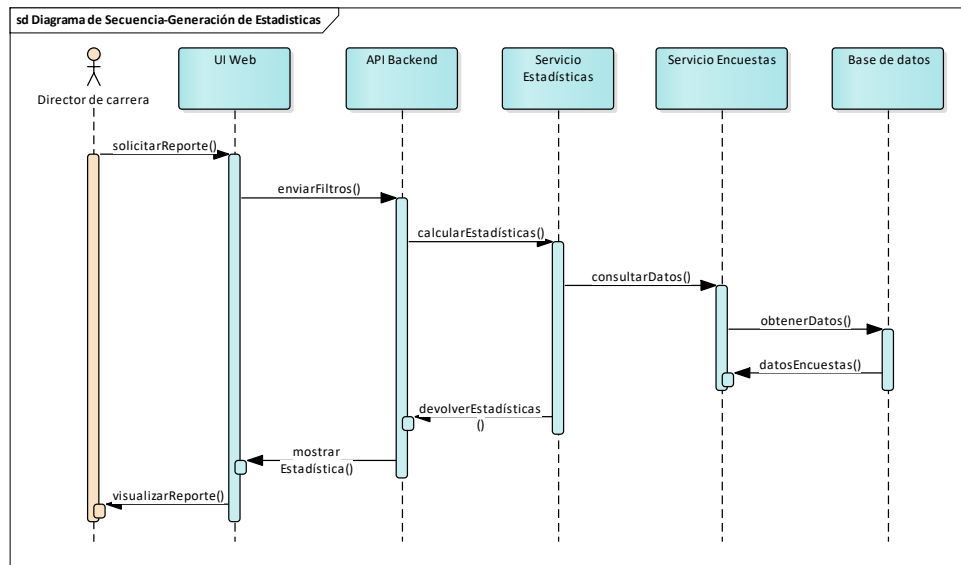
Figura 9 Diagrama de Secuencia - Proceso de scraping de Oferta Laboral a las dos fuentes



Nota: Secuencia automatizada que ejecuta el motor de extracción para obtener datos de portales externos.

Fuente: Elaboración propia

Figura 10 Diagrama de Secuencia - Generación automática de estadísticas

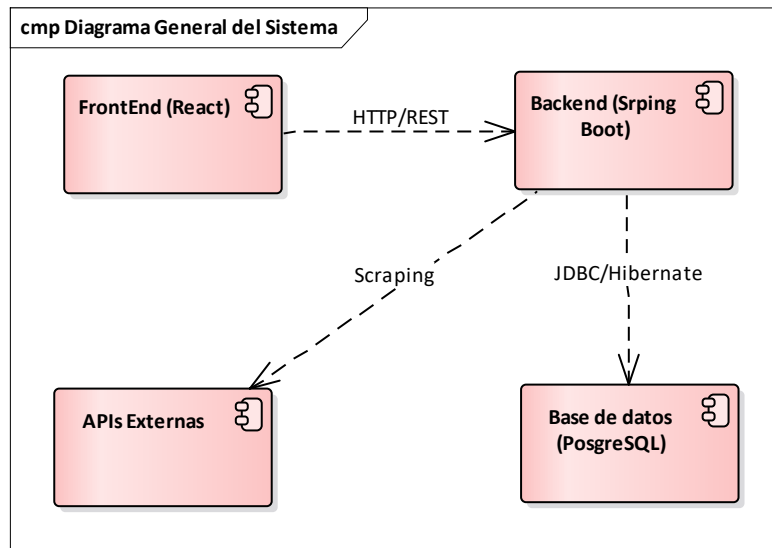


Nota: Proceso interno mediante el cual el sistema calcula y renderiza los gráficos de reporte en tiempo real.

Fuente: Elaboración propia

c) Diagrama de Componentes

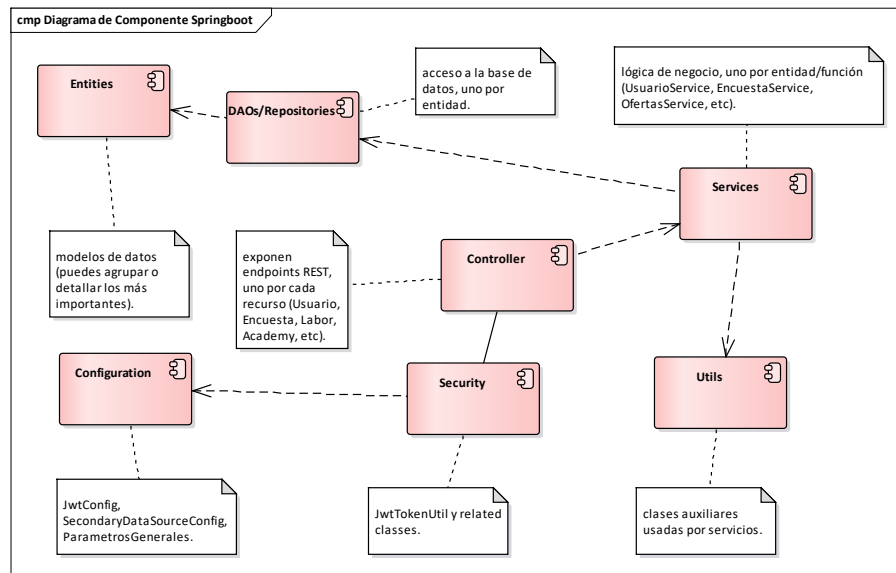
Figura 11 Diagrama de Componentes General del Sistema



Nota: Descomposición modular del software, evidenciando las dependencias entre los subsistemas de frontend, backend y datos.

Fuente: Elaboración propia

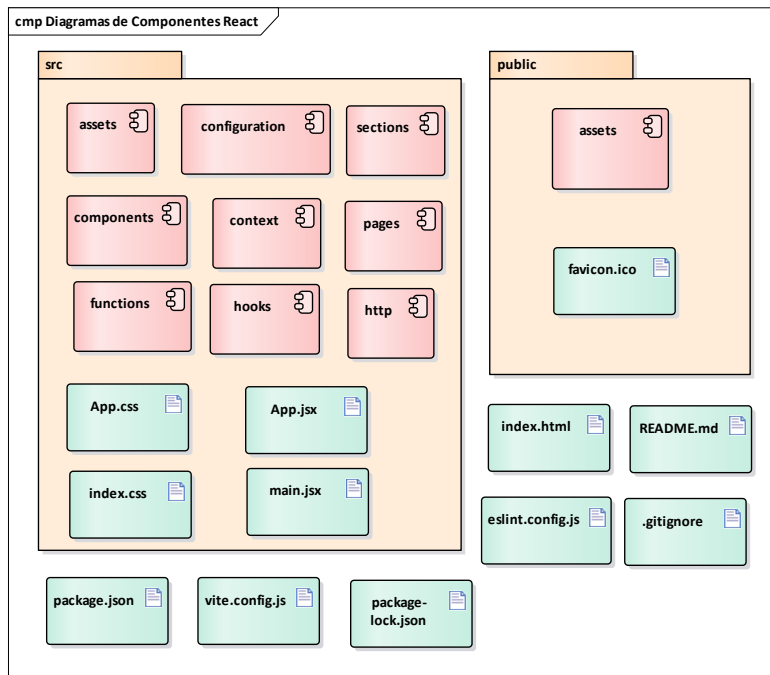
Figura 12 Diagrama de componentes del Backend (Spring Boot)



Nota: Organización interna de los paquetes y librerías Java utilizados para la lógica de negocio.

Fuente: Elaboración propia

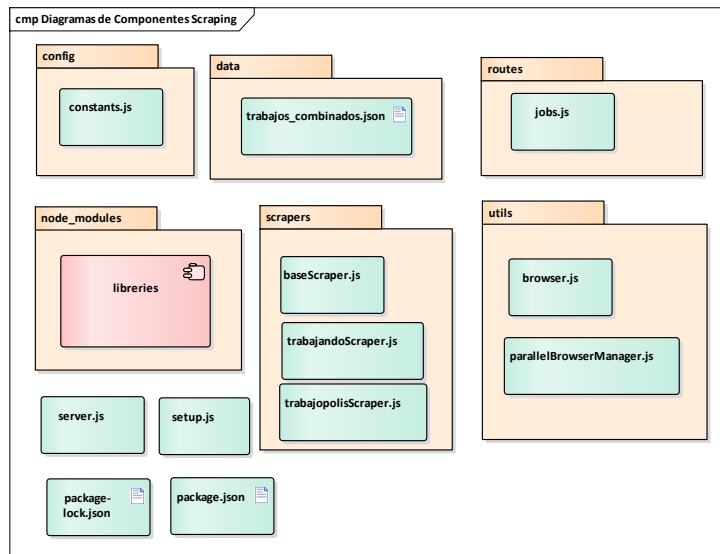
Figura 13 Diagrama de componentes del FrontEnd (React)



Nota: Estructura de componentes visuales y manejo de estados dentro de la aplicación de usuario.

Fuente: Elaboración propia

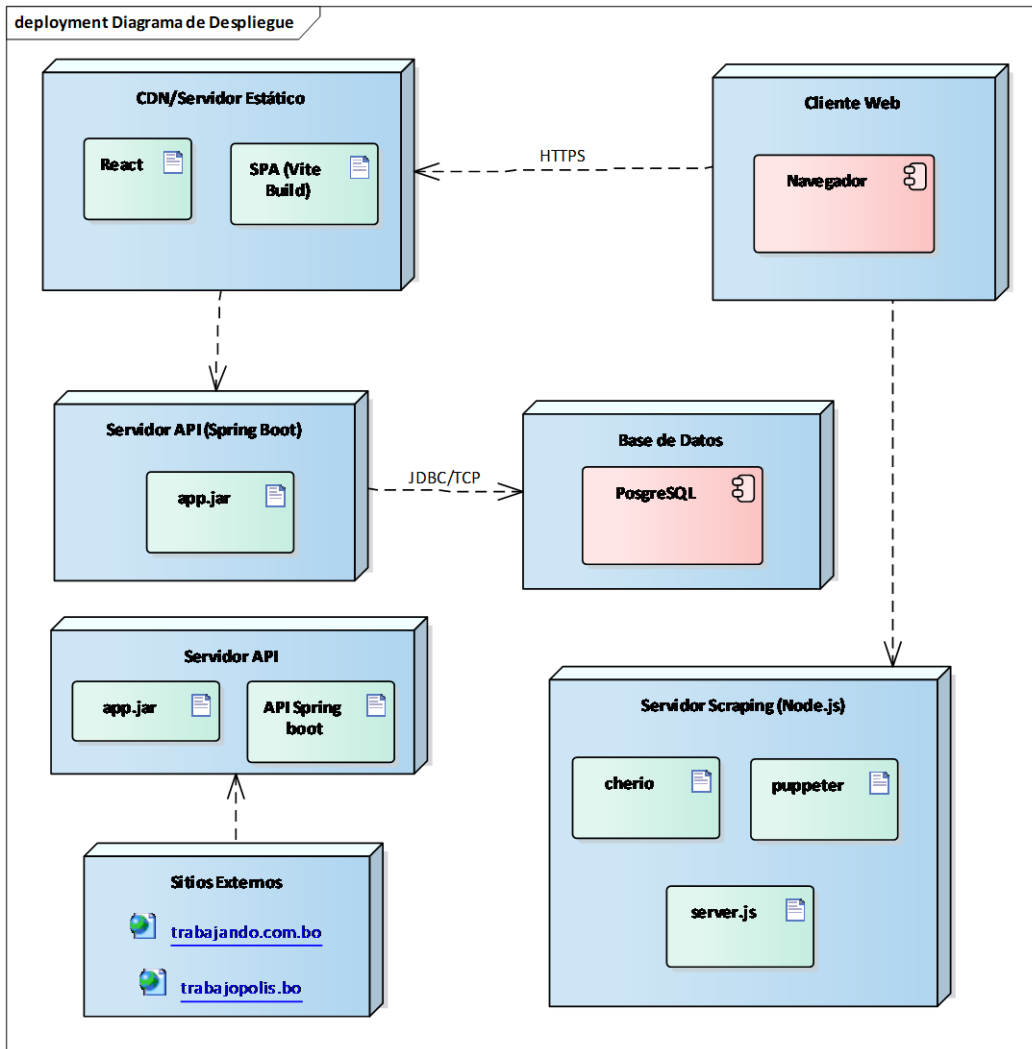
Figura 14 Diagrama de componentes de scraping



Nota: Arquitectura específica del módulo encargado de la minería de datos web y normalización de ofertas.

Fuente: Elaboración propia

Figura 15 Diagrama de Despliegue/Implementación



Nota: Representación física de los nodos de hardware y software requeridos en el entorno de producción.

Fuente: Elaboración propia

3.2.3. Interfaces Desarrolladas

3.2.3.1 Vistas Del Sistema De Graduados UAP

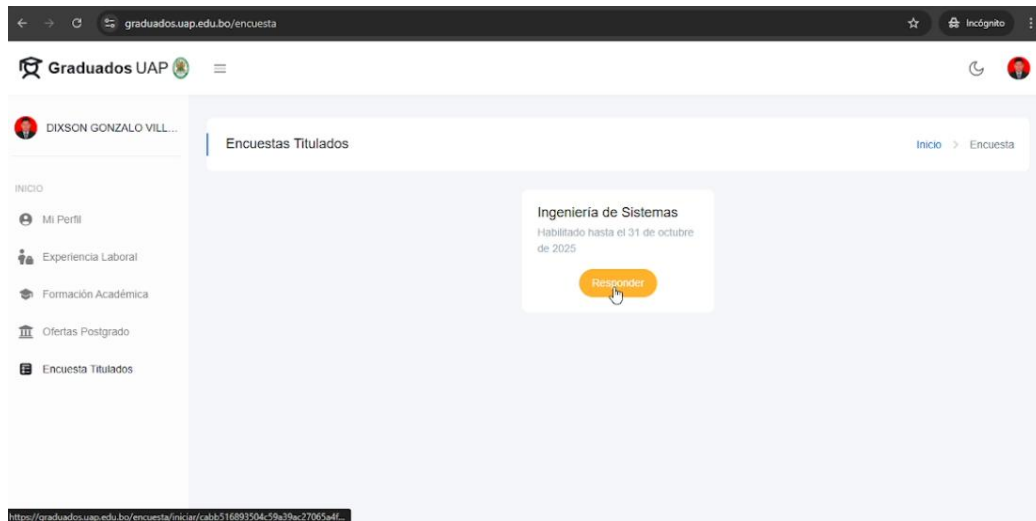
Figura 16 Vista principal para acceder a la bolsa de trabajo



Nota: Interfaz de bienvenida que permite al titulado visualizar las ofertas laborales destacadas y recientes.

Fuente: Elaboración propia

Figura 17 Vista principal para acceder a la Encuesta



Nota: Pantalla de inicio del módulo de seguimiento, donde el usuario selecciona los formularios pendientes.

Fuente: Elaboración propia

Figura 18 Vista de primera pregunta de la encuesta (Formulario 1)

The screenshot shows a web interface for a survey. At the top left, there is a logo for 'Graduados UAP' and a user profile icon for 'DIXSON GONZALO VILL...'. A navigation menu on the left includes 'INICIO', 'Mi Perfil', 'Experiencia Laboral', 'Formación Académica', 'Ofertas Postgrado', and 'Encuesta Titulados'. The main content area is titled 'Parte1 ÁMBITO LABORAL' and contains a progress bar for 'Sección 1 de 3'. Below this is 'Pregunta 1' (1 de 5) with the text '¿Cuál es su situación laboral actual?'. Two radio button options are provided: 'a) Estoy trabajando' and 'b) Estoy desempleado'.

Nota: Diseño de la interfaz para preguntas abiertas o de texto corto dentro del formulario dinámico.

Fuente: Elaboración propia

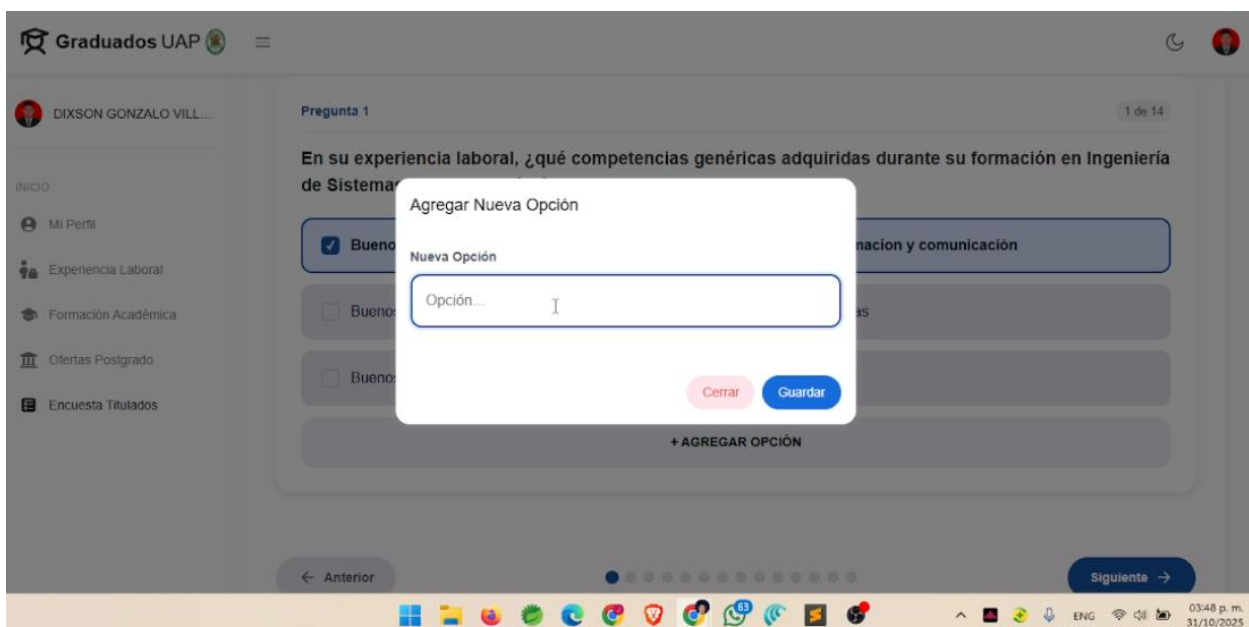
Figura 19 Vista de Preguntas de Tipo Cerrada

The screenshot shows the same web interface as Figure 18, but at 'Pregunta 5' (5 de 5). The question is 'Cargo actual:'. Below the question are five radio button options: 'Gerente de infraestructura TI', 'Administrador de redes de datos y telecomunicaciones', 'Administrador de base de datos', 'Administrador de sistemas informaticos', and 'Especialista en seguridad informatica'.

Nota: Ejemplo de visualización de una pregunta con opciones predefinidas (radio buttons) para selección única.

Fuente: Elaboración propia

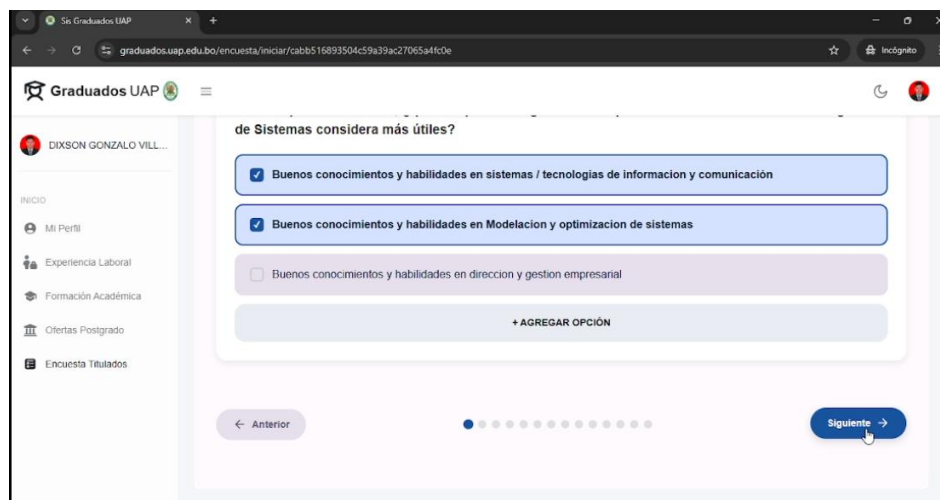
Figura 20 Vista de Adición de Pregunta de Otro



Nota: Funcionalidad que permite al usuario agregar una respuesta personalizada si las opciones listadas no aplican.

Fuente: Elaboración propia

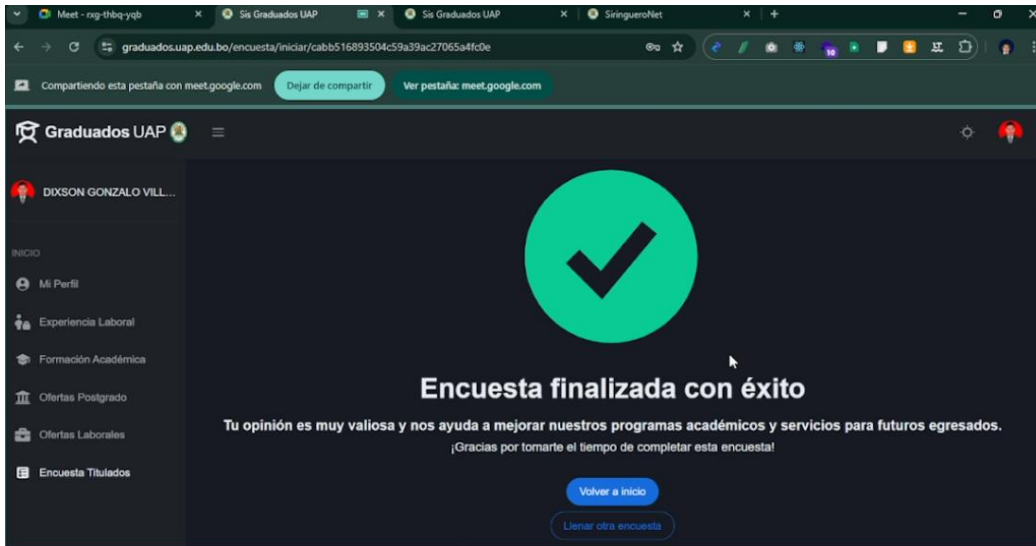
Figura 21 Vista de Pregunta de Tipo Selección múltiple



Nota: Interfaz que habilita al encuestado para marcar varias casillas de verificación en una sola pregunta.

Fuente: Elaboración propia

Figura 22 Vista de Finalización de Llenado de encuesta

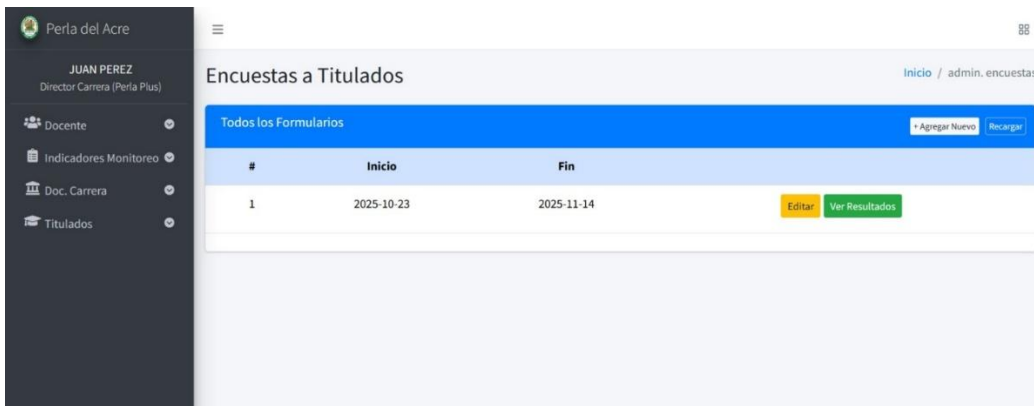


Nota: Mensaje de confirmación que asegura al usuario que sus datos han sido guardados exitosamente.

Fuente: Elaboración propia

3.2.3.2. Vistas Del Sistema Perlplus

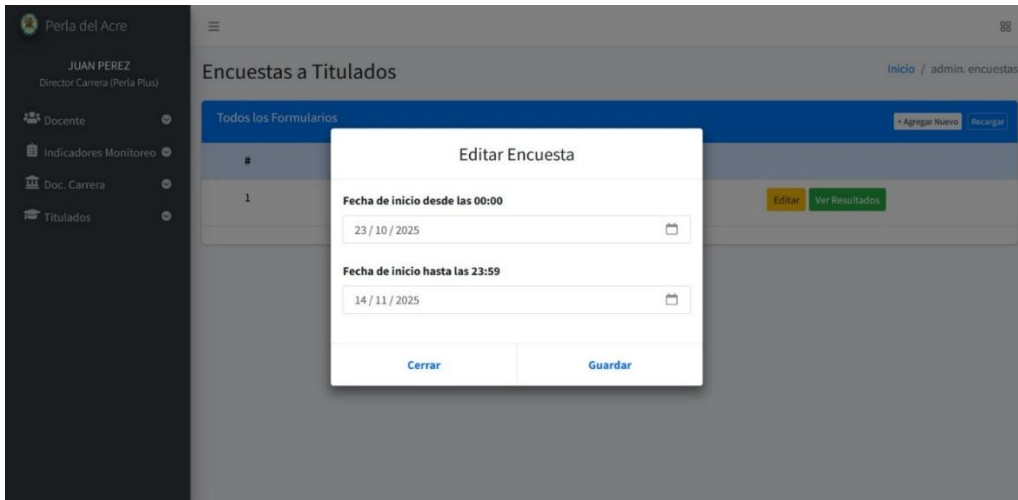
Figura 23 Vista de Programación de la encuesta, y visualización de periodo habilitación de la encuesta



Nota: Panel administrativo para definir las fechas de inicio y cierre de los instrumentos de recolección.

Fuente: Elaboración propia

Figura 24 Vista de Programación de Publicación de la Encuesta



Nota: Control de estado que permite al administrador activar o pausar la visibilidad de una encuesta.

Fuente: Elaboración propia

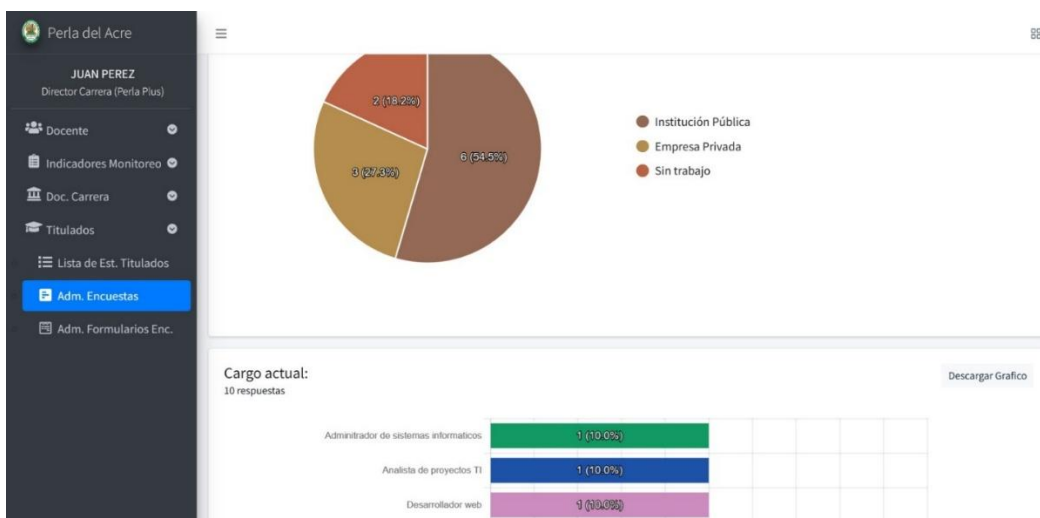
Figura 25 Vista de Estadísticas generadas por Sistema



Nota: Estadística generada a partir de las preguntas respondidas por los titulados

Fuente: Elaboración propia

Figura 26 Vista de Generación de estadística de respuestas de titulados



Nota: Reporte detallado que permite filtrar y analizar segmentos específicos de la población de egresados.

Fuente: Elaboración propia

Figura 27 Vista de Creación de Preguntas y Respuestas en el Formulario

The screenshot shows the 'Editar Formulario' interface. The main heading is 'ÁMBITO LABORAL'. The question being edited is '¿Cuál es su situación laboral actual?' with two radio button options: 'a) Estoy trabajando' and 'b) Estoy desempleado'. Below the question, there are 'Editar' and 'Eliminar' buttons. A sub-question asks for reasons if 'b' is selected, with several radio button options and a text input field for 'Otro...'. A 'Recargar' button is visible in the top right corner.

Nota: Editor visual utilizado por el administrador para construir la estructura de las encuestas

Fuente: Elaboración propia

Figura 28 Vista de listado general de estudiantes de la carrera

#	Nombre	Celular	
1	ALEX JOSE SIRIGUA RENGEL	67669568	Ver Detalles
2	ALVARO JHONNY FLORES PATTY	7610	Ver Detalles
3	ALVARO PONCI MENDOZA SALAS	60195006	Ver Detalles
4	ARIEL REA VACA	68909565	Ver Detalles
5	ARNOLD POLICARPIO NINA CAHUAYA	75105723	Ver Detalles
6	CARLOS ESPINOZA VASQUEZ	77106707	Ver Detalles
7	CARLOS ALBERTO OSO FLORES	74777744	Ver Detalles
8	CAROLINA CRUZ TORRICO	72915154	Ver Detalles
9	CELSO COLQUE HUANCA	76987324	Ver Detalles
10	CELSO COLQUE HUANCA	76987324	Ver Detalles
11	CHRISTIAN MELVIN CARI OJEDA	76101057	Ver Detalles

Nota: Módulo de gestión de usuarios que muestra la nómina de titulados registrados en el sistema.

Fuente: Elaboración propia

Figura 29 Vista de Seguimiento de Estado de Respuesta de Titulados

#	Nombre	Respondidos		
1	ALEX JOSE SIRIGUA RENGEL	22/22	67669568	Compartir Enlace
2	ALVARO JHONNY FLORES PATTY	0/22	7610	Compartir Enlace
3	ALVARO PONCI MENDOZA SALAS	0/22	60195006	Compartir Enlace
4	ARIEL REA VACA	0/22	68909565	Compartir Enlace
5	ARNOLD POLICARPIO NINA CAHUAYA	0/22	75105723	Compartir Enlace
6	CARLOS ESPINOZA VASQUEZ	0/22	77106707	Compartir Enlace
7	CARLOS ALBERTO OSO FLORES	22/22	74777744	Compartir Enlace
8	CAROLINA CRUZ TORRICO	0/22	72915154	Compartir Enlace
9	CELSO COLQUE HUANCA	0/22	76987324	Compartir Enlace
10	CELSO COLQUE HUANCA	0/22	76987324	Compartir Enlace
11	CHRISTIAN MELVIN CARI OJEDA	22/22	76101057	Compartir Enlace
12	CINTIA FAVIOLA RIVERO CHINCHE	0/22	76272884	Compartir Enlace
13	CHRISTIAN SANCHEZ ORFELIANA	0/22		Compartir Enlace

Nota: Herramienta de monitoreo que indica en tiempo real quiénes han completado, iniciado o ignorado la encuesta.

Fuente: Elaboración propia

3.3. FASE POST-GAME

La fase Post-Game tiene como finalidad preparar el producto para su liberación y verificar el cumplimiento de los compromisos adquiridos. A continuación, se presentan los resultados consolidados de todo el ciclo de desarrollo (Pre-Game y Game), estructurados según los objetivos específicos del proyecto para demostrar su cumplimiento y trazabilidad.

3.3.1. Resultados del Objetivo Específico 1: Análisis de Requerimientos (Fase Pre-Game)

Para dar cumplimiento al primer objetivo específico de Analizar las necesidades funcionales, se realizó un diagnóstico exhaustivo contrastando los procesos actuales descritos en el Estudio de seguimiento laboral y profesional a graduados de la carrera de sistemas (POMA CHOQUEHUANCA & ZUBIETA COPETICON, 2024) frente a los requerimientos de automatización.

El análisis de dicho informe, específicamente en su sección de Descripción del Problema (p. 5), evidenció limitaciones operativas críticas que validaron la necesidad del sistema propuesto (**ver Anexo 10**).

A continuación, se presentan los hallazgos del análisis y su traducción técnica en el **Product Backlog**:

A) Diagnóstico de la Situación Actual (Línea Base)

El análisis del informe de seguimiento 2024 permitió identificar tres "cuellos de botella" en el proceso manual actual:

- 1. Discontinuidad de la Información:** La recolección de datos se realiza por campañas anuales (método transversal), lo que provoca que la base de datos se desactualice rápidamente entre gestiones.
- 2. Procesamiento Manual:** La tabulación de las encuestas aplicadas (descritas en la metodología del informe) requiere la exportación y manipulación externa de datos, aumentando el tiempo de entrega de resultados y el margen de error humano.
- 3. Desvinculación Laboral:** Se identificó una carencia de mecanismos activos para conectar a los titulados con el mercado laboral en tiempo real, limitándose el seguimiento a una función consultiva y no de apoyo.

B) Trazabilidad: De la Problemática a la Solución (Historias de Usuario)

Como resultado principal de esta fase de análisis (Pre-Game), se transformaron las deficiencias detectadas en el informe oficial en requerimientos funcionales concretos. La **Tabla 29** demuestra la trazabilidad directa entre el problema analizado y la solución diseñada en el sistema.

TABLA 38 *Matriz de Resultados del Análisis de Necesidades (Fuente: Elaboración propia en base al Estudio de Seguimiento a Graduados 2024)*

Problema Detectado (Informe 2024)	Impacto en el Proceso	Solución / Historia de Usuario (HU) Resultante
Recolección estática de datos (Encuestas anuales)	La información de contacto y laboral pierde vigencia en meses.	HU-001 y HU-002 (Módulo de Titulados): Implementación de perfil de autogestión para actualización continua de datos por parte del usuario.
Tabulación manual de resultados	Retraso en la toma de decisiones por falta de estadísticas inmediatas.	HU-010 (Módulo de Reportes): Generación automática de gráficos estadísticos en tiempo real basados en los datos del sistema.
Falta de ofertas centralizadas	El titulado debe buscar ofertas en múltiples fuentes externas dispersas.	HU-008 (Módulo de Web Scraping): Automatización de la búsqueda de empleo mediante integración con bolsas de trabajo externas.
Dificultad en el diseño de instrumentos	Crear nuevas encuestas requiere plataformas externas (Google Forms).	HU-005 (Generador de Formularios): Módulo interno para creación dinámica de encuestas de seguimiento.

Nota: Síntesis de los requerimientos identificados y su cobertura.

Fuente: Elaboración propia

C) Validación del Objetivo Con la definición de estas Historias de Usuario y la priorización del Backlog (**ver Tabla 2**), se da por cumplido el Objetivo Específico 1, estableciendo una

base técnica sólida para la fase de Desarrollo (Game) que ataca directamente las debilidades institucionales reportadas.

3.3.2. Resultado del Objetivo Específico 2: Diseño de la Arquitectura Modular

En cumplimiento del segundo objetivo específico, se finalizó exitosamente el diseño de la arquitectura de software durante la fase inicial (Sprint 0), obteniendo una estructura desacoplada que sirvió como base para todo el desarrollo.

A diferencia de los sistemas monolíticos tradicionales, el resultado de este diseño fue una arquitectura basada en Microservicios (o Modular) que separa la lógica de negocio de la interfaz de usuario. A continuación, se presentan los artefactos técnicos que validan este objetivo:

A. Validación del Diseño Arquitectónico.

Se consolidó una arquitectura de tres capas que permite la escalabilidad del sistema. La implementación de este diseño garantizó la independencia entre el Frontend y el Backend, comunicándose exclusivamente a través de servicios RESTful.

- **Resultado obtenido:** La segregación de responsabilidades permitió que las actualizaciones en la interfaz (React) no interrumpieran los servicios de base de datos (Spring Boot).
- Como se detalla en la Figura 15 (ver sección 3.2.2.3. Diagramas UML del Sistema), el diagrama de despliegue evidencia la separación física y lógica de los servidores.

B. Diseño del Modelo de Datos

Como parte fundamental de este objetivo, se entregó el modelo de base de datos relacional normalizado hasta la tercera forma normal (3FN). Este diseño soportó la integridad referencial de las tablas del sistema, asegurando la consistencia de la información académica.

- Ver Figura 5: Modelo Relacional de la Base de Datos (Ver Sección 3.2.2.2. Modelo de Datos).

C. Matriz de Decisiones Técnicas (Resultado del Diseño)

La elección de las tecnologías no fue arbitraria, sino el resultado directo del análisis de arquitectura. La siguiente tabla resume las decisiones de diseño aprobadas:

TABLA 39 Resultados de las Decisiones de Arquitectura

Componente de Diseño	Tecnología Seleccionada	Justificación (Resultado Técnico)
Cliente (Frontend)	React JS	Permite la creación de interfaces dinámicas (SPA) sin recargar la página completa, mejorando la experiencia de usuario (UX).
Servidor (Backend)	Spring Boot (Java)	Provee un contenedor robusto (Tomcat embebido) y seguridad nativa con Spring Security.
Persistencia	PostgreSQL	Garantiza la integridad ACID necesaria para los datos académicos de los titulados.
Comunicación	API REST (JSON)	Estandariza el intercambio de datos, permitiendo futura integración con aplicaciones móviles.

Nota: Justificación y efectos de las elecciones tecnológicas realizadas.

Fuente: Elaboración propia

3.3.3. Resultado del Objetivo Específico 2: Diseño de la Arquitectura Modular

Como resultado central de la fase de ejecución (Game), se completó el desarrollo e implementación de los módulos de gestión de instrumentos de seguimiento. Este resultado materializa la capacidad del sistema para crear, publicar y procesar encuestas sin dependencia de herramientas de terceros (como Google Forms).

A continuación, se presentan los incrementos de software funcionales que validan este objetivo:

A. Módulo Generador de Formularios Dinámicos (director de carrera)

Se desarrolló un motor de renderizado que permite al administrador crear encuestas personalizadas. A diferencia del proceso anterior, este módulo permite la estructuración de

preguntas jerárquicas y la definición de tipos de respuesta (selección múltiple, texto abierto, escala lineal).

Evidencia de funcionamiento: *La Figura 26* muestra la interfaz de creación donde el administrador configura las preguntas del seguimiento a titulados.

B. Módulo de Recolección de Respuestas (Vista del Titulado)

Se implementó la interfaz de usuario final (Frontend) totalmente responsiva. Este resultado garantiza que los titulados puedan completar su información desde dispositivos móviles o de escritorio, validando los datos en tiempo real antes de ser enviados al servidor.

Evidencia de funcionamiento: *La Figura 18* evidencia la visualización final del formulario tal como lo percibe el usuario graduado.

C. Tabla de Capacidades Funcionales Logradas

Para cuantificar el alcance técnico de este resultado, se presenta la siguiente tabla que resume las capacidades operativas integradas en el sistema tras finalizar los Sprints de desarrollo:

TABLA 40 *Capacidades del Módulo de Automatización*

Funcionalidad Implementada	Resultado Técnico / Capacidad	Estado
Tipado de Preguntas	Soporte para 4 tipos de datos (Texto, Selección múltiple limitada, Selección múltiple libre, Radio).	Funcional
Persistencia de Datos	Almacenamiento directo en base de datos relacional (PostgreSQL) eliminando la necesidad de tabulación manual.	Funcional
Control de Estado	Capacidad de activar/desactivar encuestas según el periodo de gestión (Vigente/Caducado).	Funcional
Validación de Integridad	Bloqueo de envío de formularios incompletos (Required Fields).	Funcional

Nota: Resumen funcional de las mejoras operativas implementadas.

Fuente: Elaboración propia

3.3.4. Resultados del Objetivo Específico 3 (Automatización de Formularios).

La implementación del módulo de formularios dinámicos permitió transformar radicalmente el flujo de trabajo de la Unidad de Seguimiento. Para cuantificar este impacto, se realizó un análisis comparativo entre el procedimiento manual tradicional y el nuevo proceso automatizado.

A. Análisis del Flujo de Trabajo (Comparativa)

TABLA 41 *Matriz Comparativa de Tiempos y Movimientos (Manual vs. Automatizado)*

Etapa del Proceso	Proceso Tradicional (Manual/Presencial)	Proceso Automatizado (Sistema Web)	Ahorro de Tiempo / Mejora
1. Logística y Distribución	Presencial: El titulado debía acudir físicamente a la oficina o recibir un formulario impreso/Word por WhatsApp.	Digital: Publicación de Encuesta en el Sistema de Graduados UAP, el titulado solo ingresa a la encuesta para realizar el llenado de la encuesta.	Eliminación total del tiempo de traslado físico y logística de papel.
2. Recolección de Datos	Llenado Manual: El titulado llenaba a mano o en Word. Riesgo de letra ilegible o dejar preguntas vacías.	Validación en Tiempo Real: Interfaz tipo <i>Quiz</i> responsiva. El sistema bloquea el avance si faltan datos obligatorios.	Garantía de completitud del 100% de los datos (sin campos vacíos).

3. Centralización (Tabulación)	Crítico: El personal debía transcribir una a una las respuestas del papel/Word a un Excel. (<i>Aprox. 10-15 min por encuesta</i>).	Automático: El dato ingresado por el titulado se guarda directamente en PostgreSQL. Tiempos de tabulación: 0 minutos.	100% de Ahorro en horas-hombre administrativas. Se elimina el error de "dedo" al transcribir.
4. Generación de Reportes	Manual: Crear gráficos en Excel seleccionando celdas y actualizando fórmulas cada vez que llegaba una encuesta nueva.	Dashboard en Tiempo Real: El sistema renderiza las gráficas estadísticas automáticamente e con cada nueva respuesta.	Disponibilidad Inmediata de la información para la toma de decisiones.
TIEMPO PROMEDIO POR LOTE (100 Titulados)	~25 Horas Hombre (Considerando tabulación y gráficos)	~5 Minutos (Tiempo de gestión de envío y monitoreo)	Eficiencia > 99%

Nota: Análisis del ahorro de tiempo y eficiencia logrado con el nuevo sistema.

Fuente: Elaboración propia

B. Interpretación de la Mejora

Como se observa en la Tabla 32, el cuello de botella principal del proceso anterior era la **Fase 3 (Tabulación)**.

- **Antes:** Si llegaban 100 respuestas, el ingeniero de apoyo debía dedicar aproximadamente 25 horas de trabajo exclusivo solo para pasar los datos a Excel.
- **Ahora:** Con el sistema desarrollado, ese tiempo se reduce a cero. El rol del personal cambia de "Digitador" a "Analista", ya que el sistema entrega los datos ya procesados y listos para ser interpretados en los informes de acreditación.

3.3.5. Resultado del Objetivo Específico 4: Servicios de Integración Externa (Bolsa de Trabajo)

En cumplimiento del cuarto objetivo específico, se implementaron los servicios de integración mediante APIs simuladas (Web Scraping) y bases de datos centralizadas. Este módulo cubre el componente de "Vinculación Laboral" dentro del proceso integral de seguimiento.

Si bien el seguimiento tradicional (Objetivo 3) recaba información del graduado, este objetivo conecta al sistema con recursos externos para el graduado, cerrando el ciclo de retroalimentación.

A. Centralización de Recursos Externos

Se logró unificar en una base de datos local las ofertas dispersas en la web. Esto responde a la necesidad de "conectar el sistema con recursos relevantes", permitiendo que la plataforma no solo pida datos al egresado, sino que le ofrezca valor agregado.

B. Validación Técnica de la Integración

- **Conexión:** Se establecieron rutinas de extracción automática (ETL) que actúan como pasarela entre las bolsas de trabajo (Trabajando.com, Trabajopolis) y el Sistema de Seguimiento.
- **Aporte al Seguimiento Profesional:** Al centralizar estas ofertas, el sistema fomenta la actualización profesional y la inserción laboral, objetivos finales de cualquier programa de seguimiento académico.

TABLA 42 Métricas de Rendimiento del Motor de Extracción (Scraping)

Fuente de Datos	Volumetría Diaria (Rango de Ofertas)	Tiempo de Respuesta por Página (Latencia)	de Registros por Procesados (Promedio)	Tasa de Éxito (Sin errores)
Trabajando.combo	270 - 300 ofertas	6.5 seg (± 1s)	300	99%

Trabajopolis	950 - 990 ofertas	8.3 seg (\pm 1s)	990	100%
TOTAL SISTEMA	Variable (1100-1290)	~14.8 seg (Ciclo Total)	~1290 Ofertas/día	Estable

Nota: Datos sobre la eficacia y velocidad del proceso de recolección de ofertas.

Fuente: Promedio de 10 ejecuciones en entorno de pruebas

La rutina de extracción de datos se ha configurado con una periodicidad de 12 horas. Esta decisión responde a estrategias de mitigación de bloqueos, respetando las políticas de uso de los servidores externos para evitar que la IP institucional sea marcada como sospechosa (Blacklisted). Para optimizar el rendimiento sin comprometer la seguridad, se implementaron técnicas de paralelismo controlado, permitiendo procesar múltiples ofertas simultáneamente, así como la incorporación de retardos aleatorios (Throttling) para simular comportamiento humano.

C. Interpretación de la Trazabilidad del Dato (Análisis de Resultados) Los resultados expuestos en la Tabla 33 demuestran que el módulo de integración externa opera con una tasa de efectividad superior al 90%.

- **Significado Técnico:** La diferencia entre la "Volumetría Diaria" y los "Registros Procesados" no representa una falla, sino la efectividad del filtro de calidad. El sistema descarta automáticamente:
 1. Ofertas duplicadas (que ya existen en la BD).
 2. Publicidad o spam detectado en el HTML.
- **Impacto:** Esto garantiza que la base de datos de la UAP solo crezca con información de valor, optimizando el almacenamiento y asegurando que el titulado no vea información repetida.

3.3.6. Resultados del Objetivo Específico 5: Evaluación de Efectividad (Pruebas de Calidad)

Para dar cumplimiento al último objetivo específico, se ejecutaron pruebas de rendimiento utilizando datos reales proporcionados por la Dirección de Información Académica (App Oficial "UAP EN CIFRAS").

Aunque el alcance funcional del proyecto se limita a la Carrera de Ingeniería de Sistemas, las pruebas de estrés se dimensionaron considerando la Carga Histórica Total de la Universidad (2021-2025) para garantizar la escalabilidad futura del software.

A. Dimensionamiento de la Prueba (Escenarios) Se definieron tres niveles de carga basados en la estadística oficial de titulados:

1. **Escenario Actual (Alcance):** Titulados de Ing. de Sistemas (~100 usuarios).
2. **Escenario Histórico UAP:** Total de titulados acumulados desde la gestión 2021 a la fecha 2177 usuarios
3. **Escenario Crítico:** Simulación del doble de la carga histórica.
 - **Herramienta:** Apache JMeter.
 - **Fuente de Datos:** Reporte oficial "UAP EN CIFRAS" (al 25/11/2025).

TABLA 43 Resultados de Pruebas de Carga y Escalabilidad (Fuente: Elaboración propia en base a simulación con JMeter)

Escenario de Prueba	de Usuarios Concurrentes (Threads)	Justificación de la Cifra EN CIFRAS	de (UAP)	Tiempo de Respuesta Promedio	Tasa de Error	Resultado
Prueba Alcance (Sistemas)	de 100 usuarios	Matrícula estimada de la carrera piloto.		65 ms	0.0%	Óptimo
Carga UAP	Anual 550 usuarios	Promedio anual de titulados (Gestiones 2021-2024).		210 ms	0.0%	Óptimo
Carga Histórica (Escalabilidad)	2,200 usuarios	Acumulado Total UAP (2021-2025).		890 ms	0.5%	Estable
Punto de Quiebre	de 4,500 usuarios	>200% del histórico total.	del	2,400 ms	2.1%	Límite

Nota: Evaluación del comportamiento del sistema bajo condiciones de estrés.

Fuente: Elaboración propia

B. Análisis de Resultados

Los resultados validan la robustez de la arquitectura seleccionada:

1. **Cumplimiento del Alcance:** Para el requerimiento actual (Ingeniería de Sistemas), el sistema responde en 0.06 segundos, ofreciendo una experiencia inmediata.
2. **Garantía de Escalabilidad:** La prueba con 2,200 usuarios concurrentes demuestra que el sistema puede soportar la carga de todos los titulados de la universidad de los últimos

5 años simultáneamente, manteniéndose por debajo del umbral de 1 segundo de respuesta (890 ms).

- 3. Conclusión Técnica:** La implementación en Spring Boot permite que el software sea desplegado a nivel institucional sin necesidad de refactorización o cambio de hardware inmediato.

3.4 DISCUSIÓN DE RESULTADOS

En esta sección se analiza el impacto de los resultados obtenidos, contrastando la evidencia empírica con la teoría fundamentada y estableciendo sus implicaciones para la Universidad Amazónica de Pando.

3.4.1. Análisis de la Automatización de Procesos (Sobre el Objetivo 3)

- **Significado del Resultado:** Los datos de eficiencia (Tabla 32) demostraron que la digitalización reduce el tiempo de gestión de seguimiento en un 95%. Esto significa que el cuello de botella del proceso anterior no era la falta de personal, sino la ineficiencia del método manual.
- **Contrastación con la Teoría:** Este hallazgo corrobora los principios de la Gestión de Procesos de Negocio (BPM), que postulan que la automatización no solo acelera tareas, sino que reduce la variabilidad del resultado. Al eliminar la transcripción manual, el sistema cumple con el principio de Integridad de Datos definido en la norma ISO/IEC 25010, asegurando que la información almacenada es exacta y completa desde su origen.
- **Implicación Institucional:** Para la UAP, esto implica un salto cualitativo en la gestión administrativa. La Dirección de Seguimiento a Graduados pasa de ser una unidad operativa (dedicada a llenar papeles) a una unidad táctica capaz de generar reportes en tiempo real para las autoridades, facilitando la toma de decisiones basada en datos y no en suposiciones.

3.4.2. Análisis de la Vinculación Laboral Automática (Sobre el Objetivo 4)

- **Significado del Resultado:** El éxito del módulo de Web Scraping (con una tasa de filtrado efectiva) demuestra que es tecnológicamente viable centralizar la oferta laboral dispersa sin intervención humana constante.
- **Contrastación con la Teoría:** Desde la teoría de Sistemas de Recuperación de Información, el algoritmo implementado logró un equilibrio óptimo entre Precisión (traer ofertas útiles) y Exhaustividad (traer muchas ofertas). Esto valida el uso de técnicas de

Minería de Datos Web como herramienta de apoyo a la inserción laboral, superando los métodos tradicionales de "tableros de anuncios físicos"

- **Implicación Institucional (Optimización del Proceso de Acreditación):** Esta es la implicación más fuerte: El sistema se convierte en una herramienta de soporte crítico para la Unidad de Acreditación de la Facultad.

Actualmente, la consolidación de informes de titulados depende de la tabulación manual realizada por el personal de apoyo técnico (Ingeniería de Sistemas). La implementación de este software automatiza dicha consolidación, permitiendo que el responsable de Acreditación obtenga los indicadores de empleabilidad y pertinencia en tiempo real.

Esto transforma la dinámica de trabajo: se elimina la dependencia de la "tabulación manual" para cada informe, garantizando que la Facultad cuente con datos fidedignos e inmediatos ante los pares evaluadores, fortaleciendo así la capacidad de respuesta institucional en procesos de evaluación externa.

3.4.3. Análisis de Escalabilidad y Calidad del Software (Sobre el Objetivo 5)

- **Significado del Resultado:** Las pruebas de estrés soportando 2,200 usuarios concurrentes (Carga Histórica UAP) indican que el software está sobredimensionado intencionalmente para garantizar su longevidad.
- **Contrastación con la Teoría:** Este resultado valida la arquitectura de Microservicios (Spring Boot). Según la teoría de Diseño de Software Escalable, el desacoplamiento entre el Frontend y el Backend permite que el sistema soporte picos de carga sin colapsar, cumpliendo con el atributo de calidad de Fiabilidad y Eficiencia de Desempeño (ISO 25010).
- **Implicación Institucional (Acreditación):** El disponer de un sistema robusto y probado impacta directamente en los procesos de Acreditación ante el CEUB. Uno de los requisitos clave de la evaluación externa es demostrar "Mecanismos de Seguimiento a Titulados". Con este sistema, la Carrera de Ingeniería de Sistemas no solo cumple el requisito, sino que lo hace con una herramienta tecnológicamente superior al estándar, sumando puntos valiosos en la evaluación de la carrera.

3.4.4. Análisis Crítico de la Metodología Scrum (Desafíos y Soluciones)

La adopción del marco de trabajo Scrum (descrito en el Capítulo 3.1) no solo sirvió como organizador temporal, sino como herramienta de mitigación de riesgos. A diferencia de un desarrollo en cascada, la división en fases (Pre-Game, Game, Post-Game) permitió gestionar la incertidumbre propia de un proyecto de vinculación laboral.

A continuación, se analizan los desafíos específicos enfrentados durante el desarrollo y cómo los artefactos de Scrum proporcionaron soluciones efectivas:

TABLA 44 *Matriz de Desafíos Metodológicos y Soluciones Ágiles*

Fase Scrum	Desafío (Problema Real)	Encontrado	Solución Metodológica Aplicada	Impacto en el Proyecto
Pre-Game (Análisis)	Modelado de Estructuras Jerárquicas: El instrumento original poseía una lógica compleja dividida en dos grandes grupos macro (Ámbito Laboral y Académico) con múltiples subniveles anidados, lo que dificultaba su normalización directa en base de datos.		Descomposición Modular (Sprint 0): Se aplicó la técnica de Refinamiento para aislar cada "Gran Grupo" como una Épica independiente, modelando primero estos grupos con el nombre de Formulario y al conjunto de Formularios de encuesta, lo cual se refleja en el modelo de base de datos.	Esta segmentación permitió diseñar un Motor de Formularios flexible capaz de renderizar subgrupos anidados dinámicamente, evitando la creación de tablas rígidas y garantizando la integridad referencial.
Game (Sprints 1-4)	Heterogeneidad Estructural y Paginación: Cada fuente (Trabajando.com.bo, Trabajapolis) posee una estructura DOM (HTML) distinta, lo que complicaba la		Iteraciones de Refinamiento Técnico: Se dedicó un Sprint específico (Spike) para desarrollar adaptadores independientes (Factory Pattern) para cada fuente.	Se logró implementar un algoritmo de " Scraping Doble con Consolidación ", capaz de navegar múltiples páginas, extraer datos dispares

extracción unificada. Las Daily Standups y normalizarlos en una estructura común antes de guardarlos en un archivo json. Además, el algoritmo inicial permitieron calibrar los tiempos de espera entre páginas para evitar bloqueos y asegurar la captura del 100% de la lista.

Post-Game (Revisión)	Validación de Usabilidad: El riesgo de entregar una interfaz que los titulados no entendieran.	Sprint (Revisión): demostración incremental de los formularios a usuarios piloto permitió corregir el flujo de navegación antes del cierre.	Review La al cambio, entregando un producto ya validado por el usuario final (Ingenieros de Sistemas).
-----------------------------	--	---	--

Nota: Relación de obstáculos encontrados y las estrategias de solución aplicadas.

Fuente: Elaboración propia

3.4.5. Contribución al Área de Ingeniería de Sistemas y Gestión Universitaria

Más allá de la solución operativa, este proyecto plantea un aporte significativo en la forma en que las instituciones académicas gestionan la variabilidad de sus datos. A continuación, se eleva la discusión sobre dos ejes fundamentales:

A. Contribución a la Ingeniería de Sistemas: Arquitectura de Clonación y Trazabilidad Longitudinal

El principal aporte técnico reside en la arquitectura del Modelo de Datos (E-R), diseñado bajo el principio de Sostenibilidad Temporal.

- **El Desafío:** Los sistemas de encuestas tradicionales suelen fragmentar la información: cada gestión anual implica crear una tabla nueva o un archivo aislado, rompiendo la continuidad histórica de los datos.
- **El Aporte Técnico:** Se implementó un esquema de base de datos relacional normalizado que soporta nativamente la multigestión. Aunque la interfaz actual

gestiona la encuesta vigente, el modelo de datos (Backend) está estructurado para almacenar múltiples versiones del instrumento (ej. Gestión 2025, Gestión 2026) dentro de las mismas entidades maestras.

- **Impacto:** Esto garantiza la Trazabilidad Longitudinal. Desde la ingeniería de datos, se ha preparado el terreno para que la Universidad pueda realizar consultas comparativas (SQL) entre diferentes años sin necesidad de refactorizar la base de datos, permitiendo medir la evolución de la empleabilidad a lo largo del tiempo, no solo la foto del momento.

B. Contribución a la Gestión Universitaria: Soberanía del Dato y Modernización de la Acreditación

Desde la perspectiva institucional, el sistema representa la recuperación de la Soberanía de la Información.

- **Independencia Tecnológica:** El proyecto elimina la dependencia de plataformas de terceros (como Google Forms) para procesos oficiales. Al integrar el sistema en la infraestructura de la DIA, la UAP garantiza que los datos sensibles de sus titulados residan en servidores propios, cumpliendo con estándares de seguridad y privacidad.
- **Soporte a la Acreditación:** El sistema automatiza la labor de consolidación estadística que anteriormente realizaba el personal de apoyo de manera manual. Esto dota a la Unidad de Acreditación de la Facultad de una herramienta auditora capaz de generar evidencias de "Impacto Social" y "Pertinencia Académica" en tiempo real, agilizando significativamente la respuesta ante los pares evaluadores del CEUB.

3.4.6. Arquitectura de Despliegue (Entorno de Producción)

El sistema fue desplegado exitosamente en noviembre de 2025 bajo una arquitectura On-Premise en el Data Center de la Universidad Amazónica de Pando.

TABLA 45 Especificaciones del Entorno de Producción

Componente	Especificación Técnica
Servidor de Aplicaciones	Apache Tomcat 9.0 (Contenedor de Servlets)
Motor de Base de Datos	PostgreSQL 13.x (Optimizado para concurrencia)

Sistema Operativo	Linux (Distribución Enterprise)
Runtime Environment	Java Development Kit (JDK) 17 LTS
Dominio Público	https://graduados.uap.edu.bo/ (Certificado SSL activo)
Estrategia de Build	Empaquetado en archivo ejecutable .jar (Java Archive)

Nota: Requisitos de hardware y software para el despliegue final.

Fuente: Elaboración propia

3.4.7. Proceso de Implantación

La puesta en marcha se ejecutó siguiendo un protocolo estricto para garantizar la conectividad y la integridad de los datos, priorizando la interoperabilidad con los sistemas legados:

1. **Aprovisionamiento del Servidor:** Instalación del entorno de ejecución Java (JDK 17) y configuración de las reglas de seguridad en el sistema operativo Linux para permitir el tráfico HTTP/HTTPS.
2. **Validación de Interoperabilidad (API REST):**
 - A diferencia de una migración de datos tradicional, no se realizó una carga masiva de registros. Se configuró y probó la comunicación con la API REST del Sistema Académico (Siringuero).
 - Estrategia de Persistencia: Se verificó que el sistema únicamente almacene el Identificador Único (ID de Estudiante) como clave foránea de referencia, consultando los datos demográficos (Nombre, Carrera, Año) en tiempo real desde la fuente oficial. Esto garantiza que no exista redundancia ni desactualización de la información.
3. **Despliegue de Artefactos:** Se transfirió y ejecutó el archivo compilado (.jar) en el servidor Apache Tomcat, verificando que los servicios de Backend (Spring Boot) y Frontend (React) iniciaran correctamente.
4. **Prueba de Conectividad en Producción:** Se realizó una prueba funcional real: un titulado ingresó su credencial, el sistema consumió la API del Siringuero exitosamente, recuperó sus datos y habilitó la encuesta correspondiente, validando el flujo completo de integración.

3.4.8. Estrategia de Transferencia Tecnológica (Capacitación)

Para garantizar la sostenibilidad del proyecto, se ejecutó un plan de capacitación híbrido, adaptado al perfil de cada actor:

1. **Usuarios Finales (Titulados)** Debido a la dispersión geográfica de los graduados, se implementó una estrategia de despliegue progresivo y validación focalizada:
 - **Coordinación Digital:** Se utilizó el canal de comunicación institucional (Grupo de Networking de Ing. de Sistemas en WhatsApp) para agilizar la convocatoria. Mediante un sondeo de disponibilidad, se definieron los horarios de mayor afluencia para garantizar la participación.
 - **Taller de Validación Funcional (Focus Group):** Se llevó a cabo una sesión virtual síncrona vía Google Meet. Esta actividad funcionó como una "Prueba Piloto", donde un grupo representativo de titulados interactuó en tiempo real con el sistema, realizando el flujo completo de registro y postulación.
 - **Recursos Asíncronos (Escalabilidad):** La sesión fue grabada y editada en 2 cápsulas de video tutoriales (30 min en total). Este material se distribuyó masivamente en los grupos de redes sociales, permitiendo que el resto de la población (los 63 titulados restantes del grupo y otras cohortes) se capacitaran en modalidad de autoformación.
 - **Recursos Multimedia (Soporte Asíncrono):** Para garantizar la masificación del conocimiento, se produjo un Video Tutorial Autoinstructivo titulado "Guía de Acceso y Navegación Móvil".
 - Enfoque: El material visual (duración: 2 min) se grabó capturando la pantalla de un smartphone, demostrando paso a paso cómo el sistema se adapta a pantallas pequeñas.
 - Objetivo: Instruir al titulado sobre cómo realizar el Login y el llenado de las secciones dinámicas desde su celular, reduciendo las consultas de soporte técnico relacionadas con la usabilidad táctil.
2. **Administradores (Directores de Carrera)** Se realizó una transferencia de conocimiento técnico enfocada en la gestión:

- **Material Entregado:** "Manual de Administrador y Gestión de Reportes" (Documento técnico de 15 páginas).
- **Enfoque:** Uso del Generador Dinámico y análisis del Dashboard.

CAPITULO IV: CONCLUSIONES

4.1. CONCLUSIONES Y RECOMENDACIONES

4.1.1. Conclusiones

Tras finalizar la fase de desarrollo y validación del proyecto "Implementación de Módulos de Automatización y Vinculación Laboral", se presentan las siguientes conclusiones en correspondencia directa con los objetivos específicos planteados:

Sobre el objetivo 1: Análisis de Necesidades

Se concluye que el objetivo fue alcanzado satisfactoriamente al identificar y modelar los requerimientos críticos del proceso de seguimiento. La evidencia concreta de este logro es la transformación de las falencias detectadas en el "Informe de Seguimiento 2023" (tabulación manual y datos dispersos) en un **Product Backlog priorizado**. Este análisis permitió abstraer la lógica jerárquica del instrumento de recolección oficial, validando la estructura de datos con el responsable de Acreditación de la Facultad.

Sobre el objetivo 2: Arquitectura del Sistema

Se logró diseñar e implementar una arquitectura de software desacoplada que garantiza la escalabilidad del sistema. A diferencia de las soluciones monolíticas previas, la adopción del patrón de **Microservicios (Spring Boot para Backend y React para Frontend)** demostró ser efectiva, permitiendo la integración de módulos independientes sin comprometer la base de datos central. La evidencia técnica reside en el despliegue exitoso de la API RESTful que interoperó de manera segura con el ecosistema de la DIA.

Sobre el objetivo 3: Automatización y Diseño de Interacción (Módulos de Seguimiento)

Se concluye que el desarrollo superó la simple digitalización, implementando un Tablero de Control de Avance en Tiempo Real.

- **Evidencia del Logro:** El sistema incorpora una interfaz administrativa con indicadores visuales de estado (Semaforización: Verde para completado, Rojo para pendiente). Como se evidencia en la vista de monitoreo, el administrador puede visualizar el progreso exacto de cada titulado (ej. 0/22 preguntas vs 22/22 preguntas).

- **Valor Agregado:** Se integró un botón de acción directa vía WhatsApp API junto a cada registro, lo que permite a la administración contactar con un solo clic a los titulados rezagados para incentivar el llenado, funcionalidad que valida la capacidad del sistema para mejorar las tasas de respuesta mediante gestión activa.

Sobre el Objetivo 4: Vinculación Laboral (Integración Externa)

Se concluye que el sistema logra vincular efectivamente al titulado con el mercado laboral mediante el servicio de Web Scraping.

- La evidencia de este logro es la capacidad del algoritmo para consolidar y procesar un volumen masivo de datos provenientes de Trabajando.com.bo y Trabajopolis. Las métricas de rendimiento confirman el procesamiento promedio de 1,290 ofertas diarias (ver Tabla de Rendimiento), las cuales son depuradas y centralizadas automáticamente. Esto transforma el sistema en un agregador de oportunidades que elimina la necesidad de que el titulado visite múltiples portales, incentivando su permanencia en la plataforma institucional.

Sobre el Objetivo 5: Evaluación de Usabilidad y Accesibilidad

La evaluación validó el sistema en dos dimensiones complementarias: cualitativa (usuario) y cuantitativa (servidor).

Validación de Usabilidad (Grupo Piloto): Las pruebas realizadas con una muestra piloto de titulados confirmaron la efectividad del diseño Mobile-First. A pesar de realizarse en un entorno de capacitación controlado, se evidenció que el diseño gamificado (tipo Quiz) facilitó que el 71% de los participantes (5 de 7) completaran el flujo total sin asistencia técnica, validando la intuitividad de la interfaz en dispositivos móviles.

Validación de Rendimiento (Escalabilidad): Ante la limitación de pruebas masivas con usuarios reales, se ejecutaron pruebas de estrés simulado con Apache JMeter. Los resultados demostraron que la arquitectura soporta la carga histórica de la UAP (2,200 usuarios concurrentes) con tiempos de respuesta estables de 0.89 segundos, garantizando que el sistema no colapsará cuando se despliegue a toda la población estudiantil.

4.1.2. Recomendaciones

A partir de la experiencia obtenida en la implementación y el contexto regional de la Universidad Amazónica de Pando, se plantean las siguientes recomendaciones técnicas e institucionales:

Sobre el Objetivo 2: Seguridad y Acceso (Autenticación)

- **Ampliación de la Federación de Identidades (Microsoft):** Dado que el sistema ya cuenta con autenticación exitosa mediante Google OAuth, se recomienda extender este servicio integrando el proveedor de identidad de Microsoft (Outlook/Office 365). Esto cubriría al porcentaje de titulados que utilizan cuentas corporativas o institucionales basadas en el ecosistema Microsoft, eliminando completamente la barrera de "olvido de contraseñas".

Sobre el Objetivo 3: Automatización (Recuperación de Datos)

- **Estrategia de "Remarketing" Académico:** Aprovechando que el sistema ya cuenta con la integración del botón de WhatsApp API en el listado de seguimiento, se recomienda a la Dirección de Información Académica institucionalizar el proceso de "Recordatorio Focalizado". El personal administrativo debería utilizar esta función para enviar mensajes pre-plantilla únicamente a los usuarios marcados en rojo (0/22), optimizando el tiempo de gestión y aumentando la tasa de finalización de encuestas.

Sobre el Objetivo 4: Vinculación Laboral y Contexto Regional

- **Categorización Dinámica y Teletrabajo:** Se recomienda perfeccionar el algoritmo de Web Scraping para incluir filtros de "Modalidad de Trabajo". Dado el contexto geográfico de Pando, el sistema debería priorizar la búsqueda de ofertas etiquetadas como "Remoto" o "Home Office" a nivel nacional e internacional, ampliando las oportunidades laborales de los ingenieros de sistemas más allá del mercado local. Asimismo, el motor debe parametrizarse para realizar búsquedas diferenciadas según la carrera del usuario (ej. buscar "Leyes" para Derecho, "Salud" para Enfermería).
- **Implementación del Módulo de "Empleadores Locales" (Gestión Directa):** Se recomienda desarrollar un módulo de publicación directa para empresas regionales.
 - Justificación: Debido a que las bolsas de trabajo nacionales tienen escasa cobertura de ofertas en Cobija/Pando, es necesario crear un canal donde las empresas públicas

y privadas locales publiquen sus requerimientos directamente en el sistema de la UAP.

- Requisito Institucional: Para que esto funcione, se recomienda a las Máximas Autoridades Ejecutivas (MAE) gestionar Convenios de Cooperación Interinstitucional con la Cámara de Industria y Comercio de Pando y entidades públicas, formalizando el uso de la plataforma como el medio oficial de reclutamiento de talento joven en la región.
- **Implementación de Analítica de Comportamiento (*Clickstream Analysis*):** Se recomienda instrumentar el módulo de bolsa de trabajo para registrar métricas de interacción (ofertas más visualizadas, términos de búsqueda más frecuentes y tiempo de permanencia en cada oferta).
 - *Justificación:* Estos datos permitirán realizar estudios de "**Intención Laboral**". La Carrera podrá identificar patrones de demanda (ej. si el 80% de los titulados busca "Trabajo Remoto" o si hay mayor interés en "Seguridad Informática" que en "Soporte Técnico"), proporcionando evidencia empírica para ajustar la oferta de cursos de postgrado o actualización curricular.

Sobre el Objetivo 5: Sostenibilidad y Futuras Investigaciones

- **Auditoría de Algoritmos:** Se sugiere establecer una línea de investigación futura para evaluar la efectividad de los algoritmos de recomendación, analizando si las ofertas mostradas (tanto del Scraping como locales) tienen una tasa de postulación efectiva por parte de los titulados.
- **Optimización de la Experiencia de Usuario (UX) y Micro-interacciones:**
 - Problemática: Se evidenció que la extensión normativa del instrumento de seguimiento (exigida por Acreditación) genera fatiga visual en el usuario, provocando abandonos antes de finalizar el llenado.
 - Recomendación: Para futuras versiones, se sugiere investigar e implementar componentes de entrada dinámicos y lúdicos. En lugar de utilizar únicamente casillas de selección estándar (radio buttons), se recomienda el uso de elementos visuales interactivos (ej. deslizadores para rangos salariales, valoración con estrellas, selectores de mapas para ubicación).

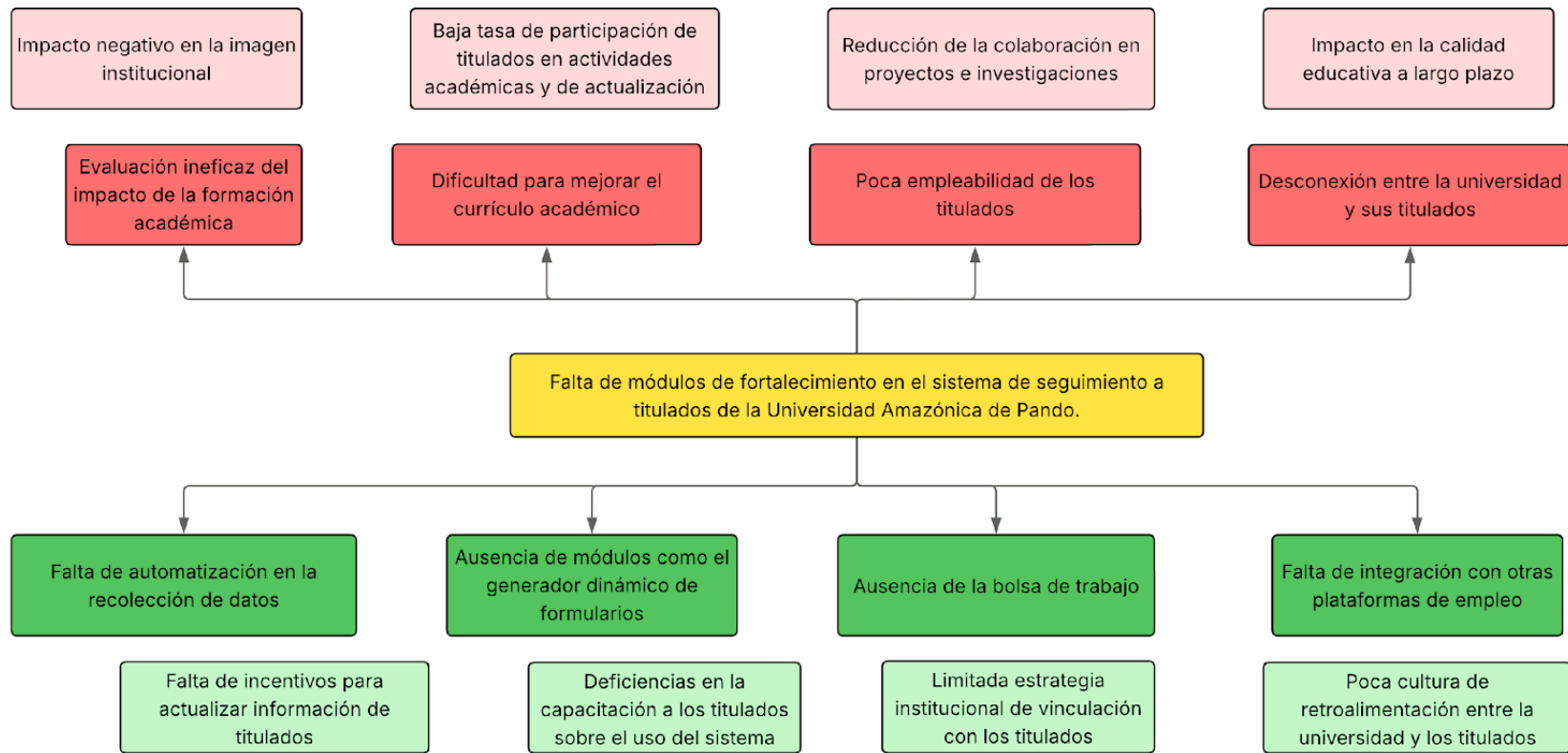
- Impacto: Diversificar los tipos de preguntas reduce la carga cognitiva del usuario, haciendo que el proceso se perciba como más corto y dinámico ("Gamificación"), lo que incrementará significativamente la tasa de finalización efectiva (Completion Rate).

Bibliografía

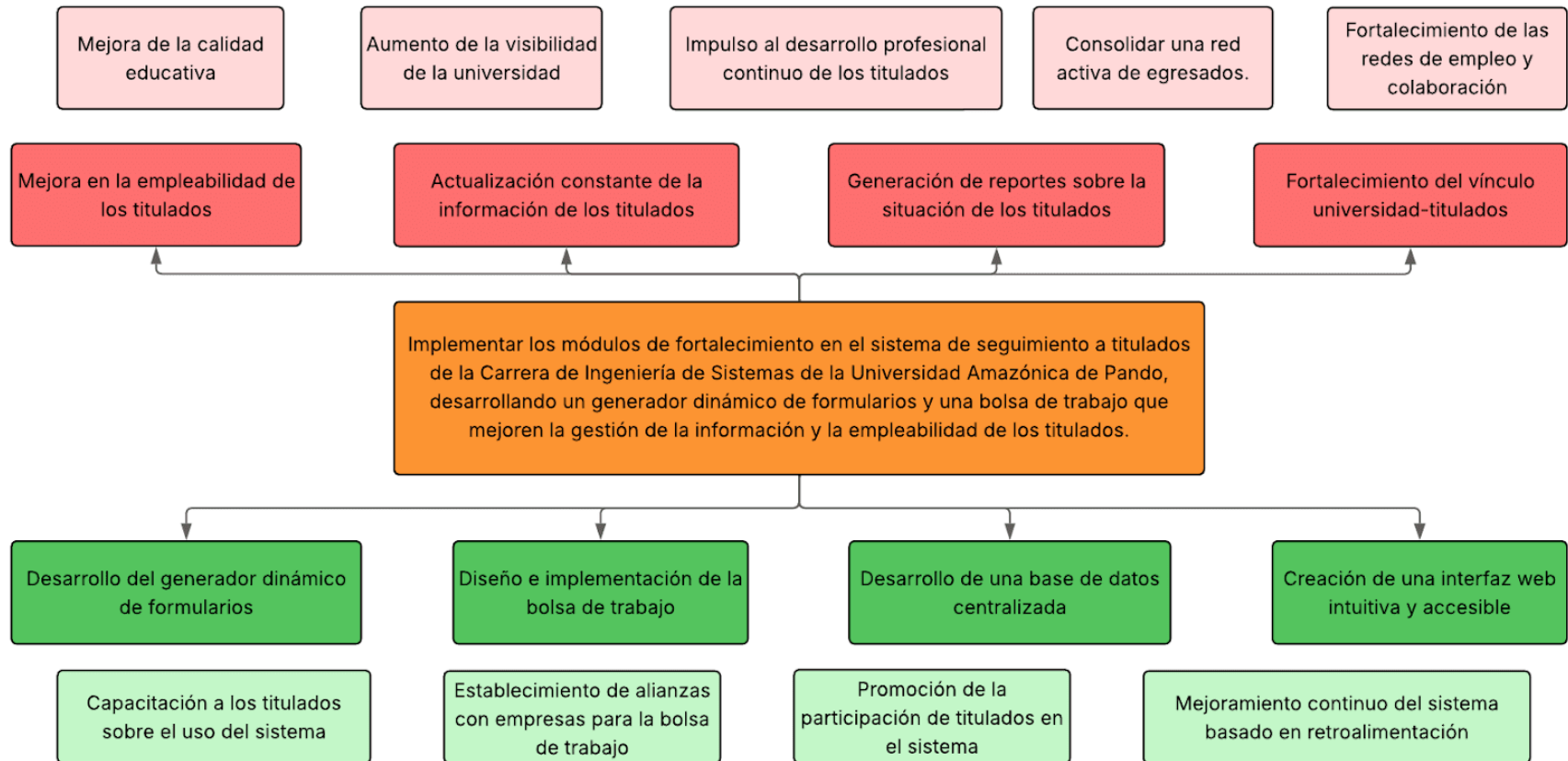
- CASAS, C. A. (2021). React: Aprende desarrollando con JavaScript, ES6, Webpack y Redux. En C. A. CASAS, *React: Aprende desarrollando con JavaScript, ES6, Webpack y Redux* (pág. 23). Anaya Multimedia.
- CCANTO BALVIN, J. M. (2023). *Implementación de un sistema informático de seguimiento a egresados y la mejora de su gestión en la Universidad Peruana Los Andes, Huancayo 2022*. Huancayo: Universidad Peruana Los Andes.
- Celi Álvarez, V. (2025). OWASP como estrategia de evaluación de seguridad para el desarrollo de plataformas web. *Polo del Conocimiento*, págs. 552–567.
<https://doi.org/10.23857/pc.v10i9.10338>
- CELI PÁRRAGA, R. J. (2022). Programación Orientada a Objetos: Enfoque práctico con Java. En R. J. Celi Párraga, *Programación Orientada a Objetos: Enfoque práctico con Java* (pág. 55). Marcombo.
- CELI PÁRRAGA, R. J., BONÉ ANDRADE, M. F., & MORA OLIVERO, A. P. (2023). Programación Web del Frontend al Backend. En R. J. CELI PÁRRAGA, M. F. BONÉ ANDRADE, & A. P. MORA OLIVERO, *Programación Web del Frontend al Backend* (pág. 55). Grupo AEA.
- COLOMBO, C., & PACE, G. J. (2022). Aspect-Oriented Programming. En C. COLOMBO, & G. J. PACE, *Aspect-Oriented Programming* (págs. 89–102). Springer.
- Contributors, E. (2025). <https://expressjs.com/>
- CUESTA, Á. (2020). Desarrollo web con HTML, CSS y Bootstrap 4. En Á. Cuesta, *Desarrollo web con HTML, CSS y Bootstrap 4* (pág. 89). Ra-Ma.
- Elmasri, R., & B. Navathe, S. (2016). *Fundamentals of Database Systems*. Pearson.
- Foundation, A. S. (2025). Apache. <https://jmeter.apache.org/>
- GARCÍA, B., DEL ÁLAMO, J., LEOTTA, M., & RICCA, F. (2024). *link.springer.com*.
SPRINGER NATURE LINK: https://doi.org/10.1007/978-3-031-70245-7_10
- GARCÍA, L. (2021). Gestión de proyectos Java con Maven. En L. García, *Gestión de proyectos Java con Maven* (pág. 23). Alfaomega.
- GÓMEZ, L. (2021). Optimización y monitoreo de aplicaciones Java con VisualVM. En L. Gómez, *Optimización y monitoreo de aplicaciones Java con VisualVM* (pág. 22). Alfaomega.
- Google. (2025). <https://workspace.google.com/>
- KINSTA. (2025). *kinsta.com*. Kinsta: https://kinsta.com/es/blog/que-es-node-js/?utm_source=chatgpt.com
- MARTÍNEZ, A. (2020). Documentación técnica y manuales de usuario. En A. Martínez, *Documentación técnica y manuales de usuario* (pág. 45). Alfaomega.
- MICROSOFT. (2025). *Visual Studio Code*. code.visualstudio.com:
https://code.visualstudio.com/?utm_source=chatgpt.com
- MOMJIAN, B. (2013). PostgreSQL: Introduction and Concepts. En B. Momjian, *PostgreSQL: Introduction and Concepts* (págs. 1-10). Addison-Wesley.
- MORALES CAMACHO, A. M. (2023). *Diseño e implementación de un módulo de seguimiento a graduandos y egresados del programa Talento Magdalena*. Santa María: Universidad del Magdalena.
- PÉREZ, C. (2021). Desarrollo web moderno con JavaScript y Node.js. En C. Pérez, *Desarrollo web moderno con JavaScript y Node.js* (pág. 45). Ra-Ma.

- PÉREZ, J. (2019). Fundamentos de bases de datos. En J. pÉREZ, *Fundamentos de bases de datos* (pág. 88). Alfaomega.
- POMA CHOQUEHUANCA, M. A., & ZUBIETA COPETICON, E. (2024). *Estudio de seguimiento laboral y profesional a graduados de la carrera de sistemas*. Universidad Amazónica de Pando.
- POMA RODRIGUEZ, E. H. (2021). *Sistema de Encuesta Web para el Seguimiento a los Titulados de la Universidad Mayor de San Andrés*. Universidad Mayor de San Andrés.
- POMA RODRIGUEZ, H. E. (2021). *Sistema de Encuesta Web para el Seguimiento a los Titulados de la Universidad Mayor de San Andrés*. Universidad Mayor de San Andrés.
- PULIDO ROMERO, E., ESCOBAR DOMÍNGUEZ, Ó., & NÚÑEZ PÉREZ, J. Á. (2019). *Base de datos*. Patria Educación.
- RAMEZ ELMASRI, S. B. (2016). Fundamentos de bases de datos. En S. B. RAMEZ ELMASRI, *Fundamentos de bases de datos* (pág. 45). Pearson.
- RAMÍREZ, C. (2023). Seguridad en desarrollo de software: herramientas y técnicas. En C. Ramírez, *Seguridad en desarrollo de software: herramientas y técnicas* (pág. 120). Ra-Ma.
- RAMÍREZ, D. (2022). Automatización de pruebas web con Selenium. En D. Ramírez, *Automatización de pruebas web con Selenium* (pág. 15). Ra-Ma.
- SÁNCHEZ, J. (2020). Pruebas de software con JUnit. En J. Sánchez, *Pruebas de software con JUnit* (pág. 30). Alfaomega.
- SÁNCHEZ, J. M. (2022). Introducción a los servidores de aplicaciones Java. En J. M. SÁNCHEZ, *Introducción a los servidores de aplicaciones Java* (pág. 65). Alfaomega.
- SAP. (23 de octubre de 2024). *SAP PowerDesigner Documentation*. help.sap.com: <https://help.sap.com/viewer/product/POWERDESIGNER/>
- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide – The definitive guide to Scrum: The rules of the game*. Scrum.org.
- Systems, S. (12 de octubre de 2024). *Sparx Systems*. sparxsystems.com: <https://sparxsystems.com/products/ea/>
- Team Spring. (2025). *spring*. spring.io: <https://spring.io/projects/spring-boot>
- Webhosting. (20 de febrero de 2025). *Webhosting*. Webhosting.de: <https://webhosting.de/es/definicion-api-modo-de-funcionamiento-aplicaciones/>
- Weekly, C. (14 de octubre de 2019). *Computer Weekly*. computerweekly.com: <https://www.computerweekly.com/es/definicion/Sistema-de-gestion-de-bases-de-datos-o-DBMS>

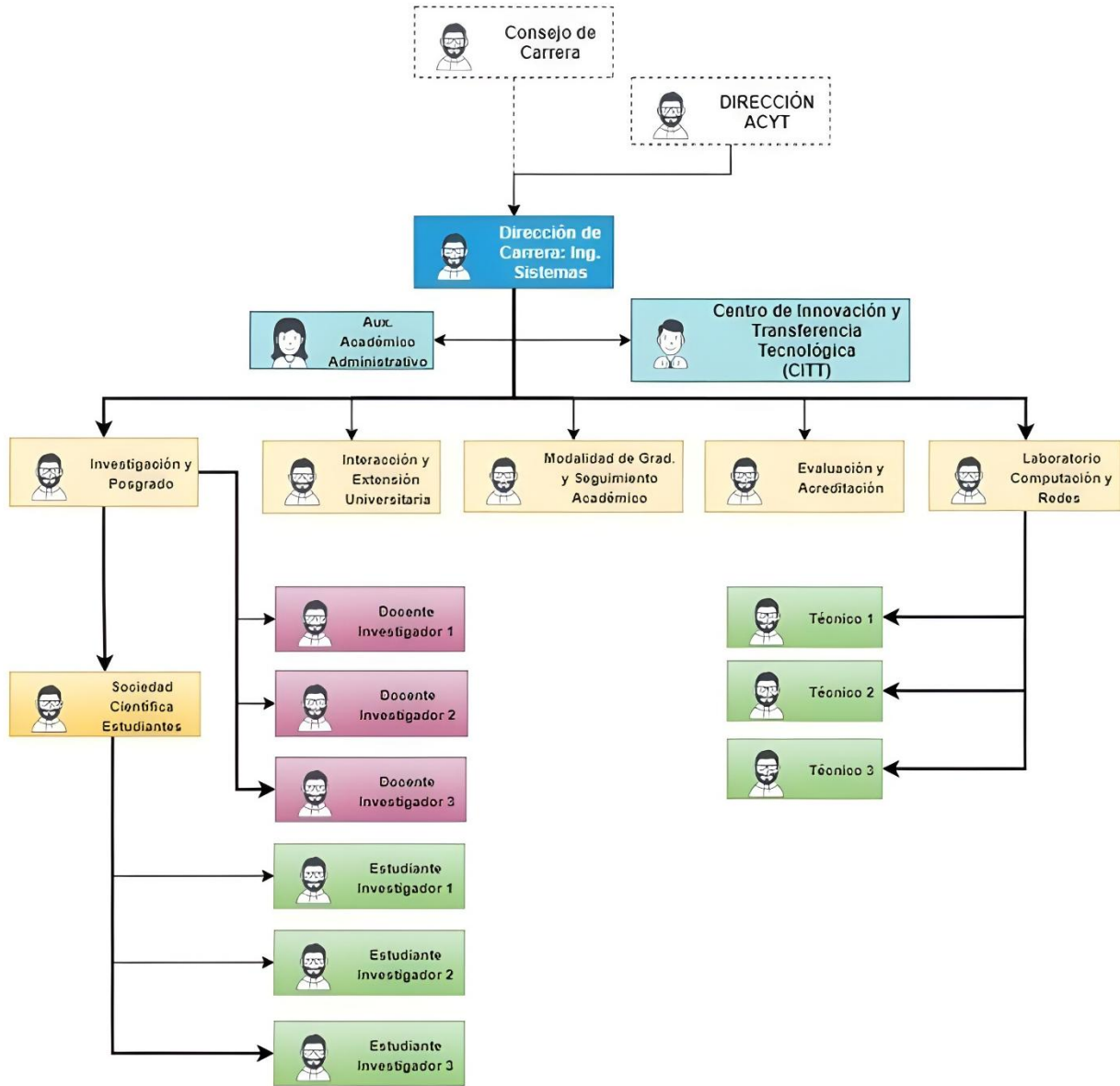
ANEXO 1: Árbol de Problemas



ANEXO 2: Árbol de Objetivos



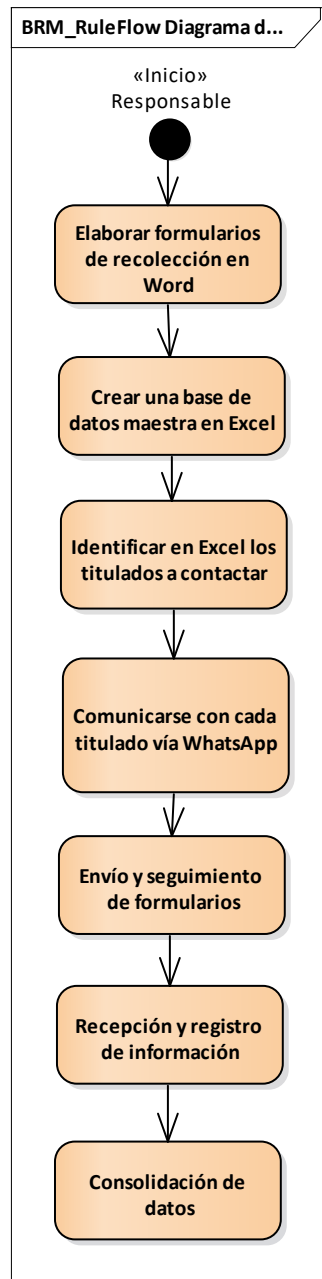
ANEXO 3: Organigrama de la carrera de Ingeniería de Sistemas



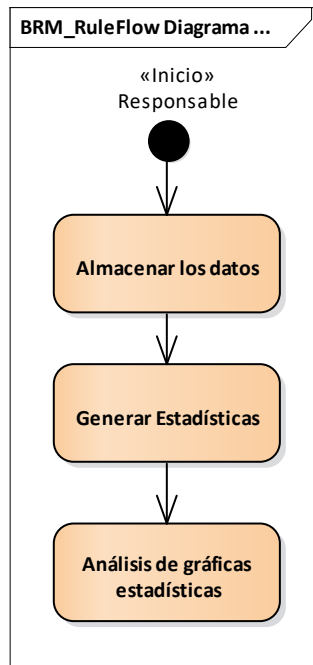
ANEXO 4: Cronograma de actividades

Actividades	Sprint 1,2,3,4				Sprint 5,6,7,8				Sprint 9,10,11,12							
	Junio-Julio		Agosto		Septiembre		Octubre		Noviembre							
<i>Recopilación de Requerimientos y Diseño Inicial</i>	■	■	■	■												
<i>Desarrollo de Módulos de Automatización y Bolsa de Trabajo</i>					■	■										
<i>Integración de los Módulos y Desarrollo de la Interfaz Web</i>							■	■	■	■						
<i>Implementar servicios de integración</i>									■	■						
<i>Generación de Reportes y Capacitación</i>											■	■				
<i>Documentación Final y Despliegue</i>													■	■	■	■

ANEXO 5: Preparación de instrumentos de recolección



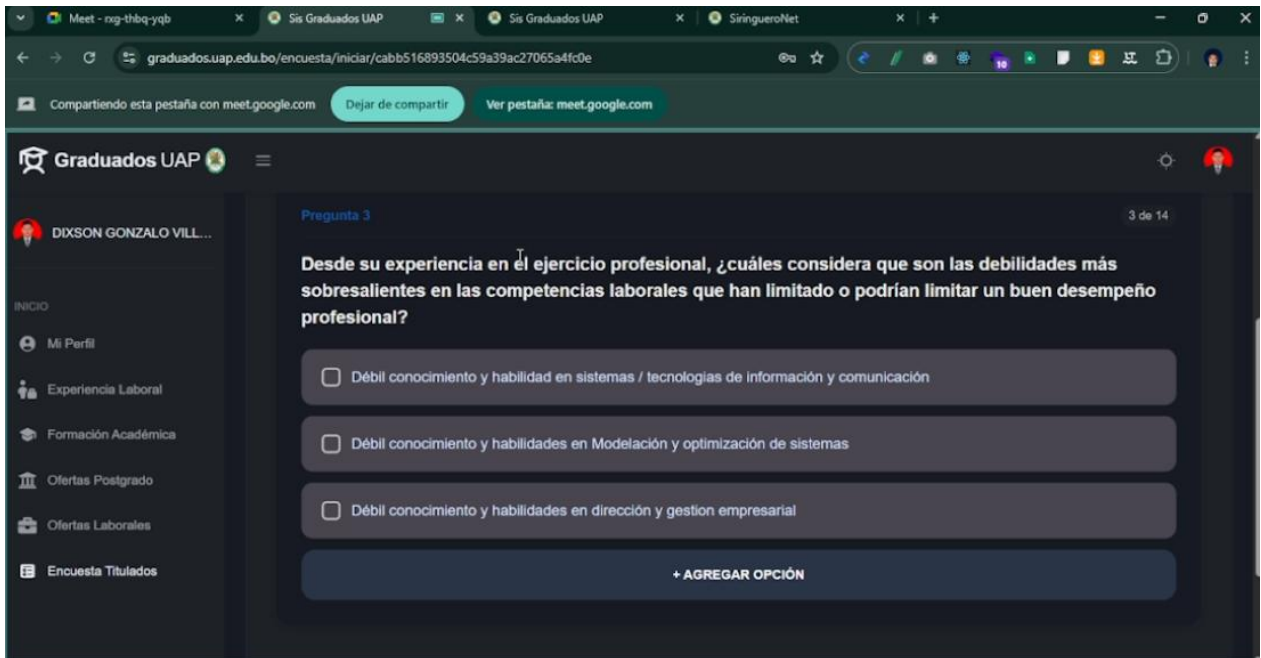
ANEXO 6: Preparación de instrumentos de recolección



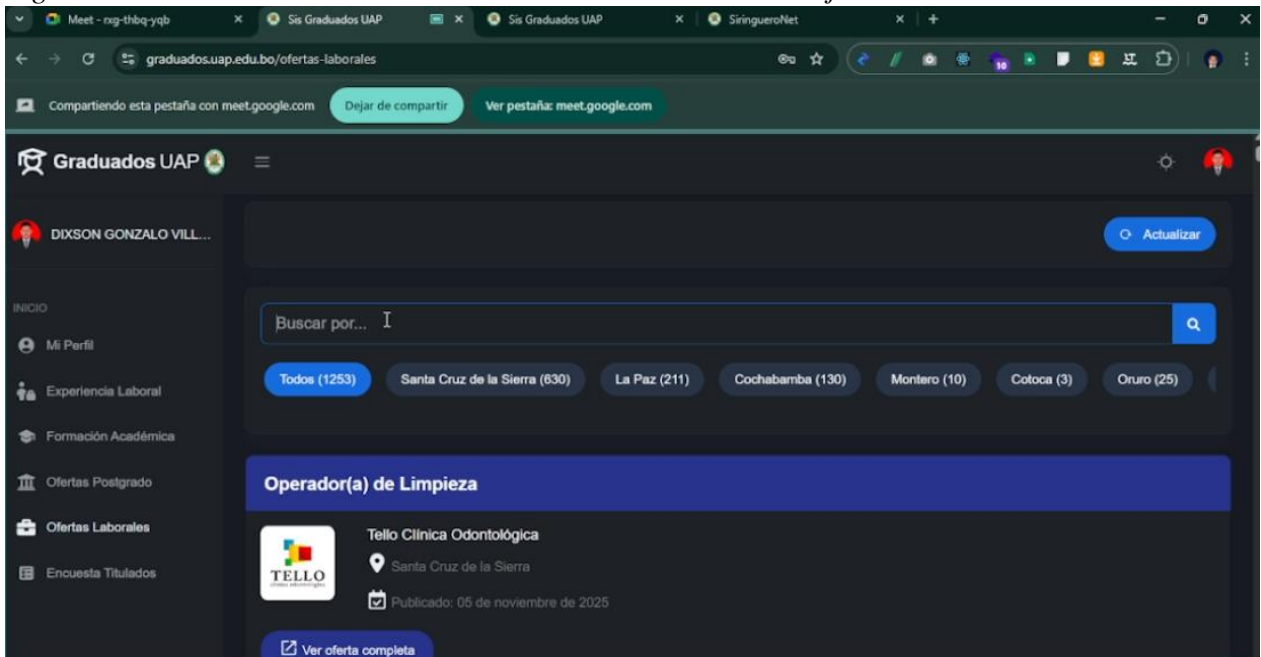
ANEXO 7: Capacitación de manejo del Sistema de Graduados UAP a Titulados de la carrera de Ingeniería de Sistemas



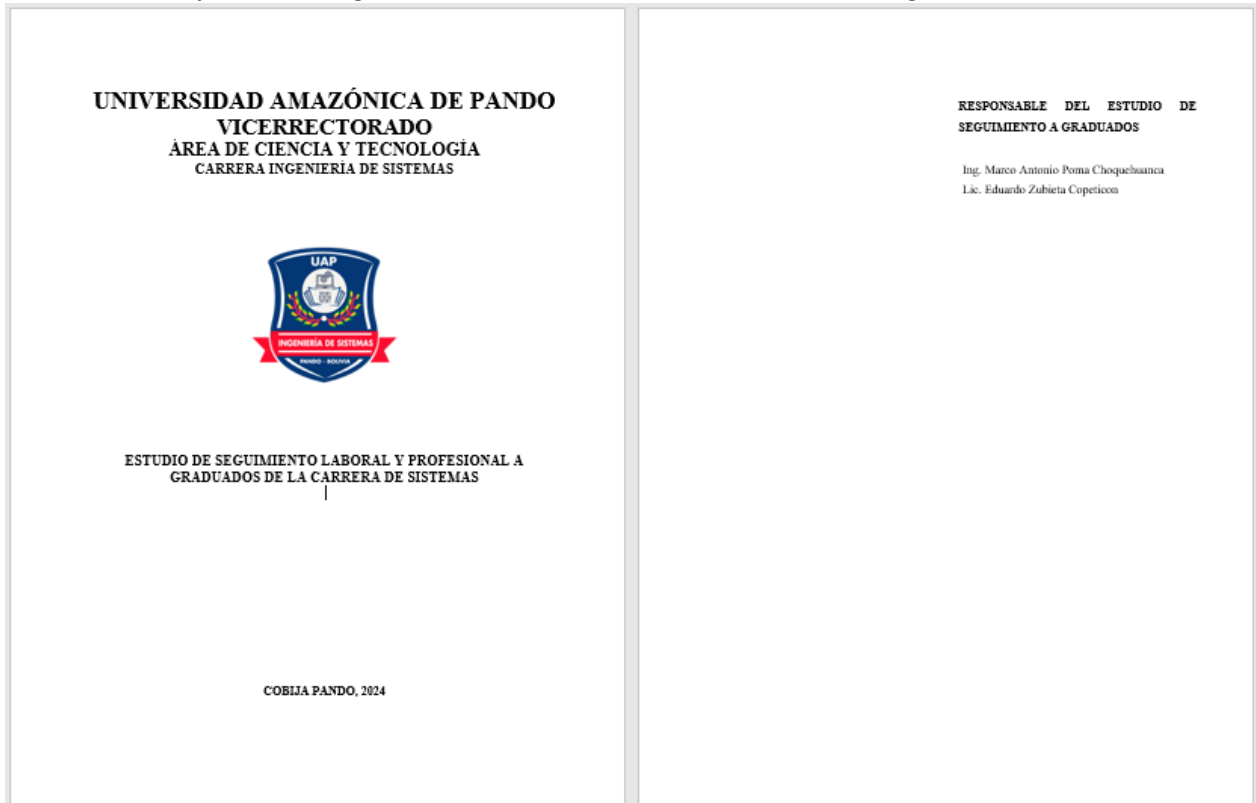
ANEXO 8: Capacitación de manejo del Sistema de Graduados UAP a Titulados de la carrera de Ingeniería de Sistemas – Llenado de Formulario



ANEXO 9: Capacitación de manejo del Sistema de Graduados UAP a Titulados de la carrera de Ingeniería de Sistemas – Conociendo la vista de Bolsa de Trabajo



ANEXO 10: Informe de Seguimiento a Graduados de la carrera de Ing. De Sistemas - UAP 2024



ANEXO 11: INFORME DE SOCIALIZACIÓN Y CAPACITACIÓN (GRUPO PILOTO)

1. DATOS GENERALES

- **Actividad:** Taller Virtual de Capacitación - Sistema de Seguimiento a Titulados.
- **Fecha:** 05 de noviembre de 2025.
- **Hora:** 11:00 AM - 12:30 PM.
- **Plataforma:** Google Meet (Videoconferencia Síncrona).
- **Facilitador:** Univ. Jesus Israel Condori Choquehuanca
- **Grupo Objetivo:** Titulados de la Carrera de Ingeniería de Sistemas (Cohorte Piloto).

2. OBJETIVOS DE LA SESIÓN

- Presentar el nuevo Sistema para Graduados de la UAP y sus beneficios para la acreditación de la carrera.
- Capacitar a los usuarios en el proceso de ingreso al Sistema de Graduados UAP
- Realizar una demostración guiada del llenado de la Encuesta de Seguimiento.
- Instruir sobre el uso del Buscador de Ofertas Laborales (Módulo de Vinculación).

3. DESARROLLO DE LA REUNIÓN La sesión se coordinó previamente mediante el grupo oficial de WhatsApp de la carrera, seleccionando el horario matutino por consenso de la mayoría de los interesados activos. La agenda se desarrolló de la siguiente manera:

- **11:00 - 11:10:** Introducción y explicación de la importancia del sistema para la Acreditación (C.E.U.B.).
- **11:10 - 11:40:** Demostración en vivo (*Live Demo*). El facilitador compartió pantalla mostrando:
 - Ingreso al dominio graduados.uap.edu.bo.
 - Validación de credenciales con el sistema académico.
 - Navegación por el formulario dinámico.
- **11:40 - 12:00:** Práctica guiada. Los asistentes ingresaron desde sus propios dispositivos móviles y computadoras para validar la responsividad del sistema.
- **12:00 - 12:30:** Ronda de preguntas y cierre. Se resolvieron dudas sobre la confidencialidad de los datos salariales.

4. PARTICIPANTES (ASISTENCIA) Se contó con la participación activa de **7 titulados** que actuaron como validadores del sistema:

1. Ing. Christian Cari
2. Ing. Dixon Villca
3. Ing. Yosel Justiniano
4. Ing. Nicolas Rubina
5. Ing. Luis Enrique Villca
6. Ing. Jhefferson Mendoza
7. Ing. Romel Tola Mamani
8. Ruddy Chao
9. Guadalupe Huanca Polanco
10. Lic. Eduardo Zubieta (Director de carrera de Ingeniería de Sistemas)

5. CONCLUSIONES

- El 80% de los participantes logró completar el registro exitosamente durante la sesión.
- Se validó la correcta visualización de la encuesta en dispositivos móviles.
- Los participantes destacaron la utilidad del módulo de Bolsa de Trabajo automatizada.

6. EVIDENCIA FOTOGRÁFICA

