



## **UNIVERSIDAD PRIVADA DE TACNA**

### **FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

**Aplicación Web Basada en IA para la Vinculación  
Laboral de Egresados de la Escuela Profesional de  
Ingeniería de Sistemas**

Curso: *Construcción de Software I*

Docente: *Ing. Alberto Johnatan Flor Rodríguez*

Integrantes:

*Agreda Ramirez, Jesús Eduardo* (2021069823)  
*Ortiz Fernandez, Ximena Andrea* (2021071080)

**Tacna – Perú  
2025**



# **Aplicación Web Basada en IA para la Vinculación Laboral de Egresados de la Escuela Profesional de Ingeniería de Sistemas**

## **Documento de Arquitectura de Software**

**Versión 4.0**



CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	Agreda Ramirez, Jesús Eduardo Ortiz Fernandez, Ximena Andrea	Agreda Ramirez, Jesús Eduardo Ortiz Fernandez, Ximena Andrea		01/05/2025	Versión Original
2.0	Agreda Ramirez, Jesús Eduardo Ortiz Fernandez, Ximena Andrea	Agreda Ramirez, Jesús Eduardo Ortiz Fernandez, Ximena Andrea		06/05/2025	Modificación Diagramas
3.0	Agreda Ramirez, Jesús Eduardo Ortiz Fernandez, Ximena Andrea	Agreda Ramirez, Jesús Eduardo Ortiz Fernandez, Ximena Andrea		15/05/2025	Revisión Final
4.0	Agreda Ramirez, Jesús Eduardo Ortiz Fernandez, Ximena Andrea	Agreda Ramirez, Jesús Eduardo Ortiz Fernandez, Ximena Andrea		26/05/2025	Revisión Unidad II

## ÍNDICE GENERAL

1. Introducción.....	5
1.1. Propósito (Diagrama 4+1).....	6
1.2. Alcance.....	8
1.3. Definiciones, Siglas y Abreviaturas.....	10
1.4. Organización del documento.....	11
2. Objetivos y Restricciones Arquitectónicas.....	11
2.1. Priorización de requerimientos.....	11
2.1.1. Requerimientos Funcionales.....	11
2.1.2. Requerimientos No Funcionales - Atributos de Calidad.....	12
2.2. Restricciones.....	12
3. Representación de la Arquitectura del Sistema.....	14
3.1. Vista de Caso de Uso.....	14
3.1.1. Diagramas de Casos de Uso - (Link a Diagrama de Caso de Uso).....	14
3.2. Vista de Lógica.....	19
3.2.1. Diagrama de Subsistemas (paquetes) - (Link al Diagrama).....	19
3.2.2. Diagrama de Secuencia (vista de diseño).....	20
3.2.3. Diagrama de Colaboración (vista de diseño).....	23
3.2.4. Diagrama de Actividades con Objetos.....	24
3.2.5. Diagrama de Clases (Link a Diagrama de Clases).....	28
3.2.6. Diagrama de Base de Datos (Link Diagrama de Base de Datos).....	29
3.3. Vista de Implementación (vista de desarrollo).....	30
3.3.1. Diagrama de arquitectura de software (paquetes) - (Link al Diagrama)...	30
3.3.2. Diagrama de arquitectura del sistema (diagrama de componentes).....	31
3.4. Vista de Procesos.....	32
3.4.1. Diagrama de procesos del sistema (diagrama de actividad).....	32
3.5. Vista de Despliegue.....	33
3.5.1. Diagrama de despliegue.....	33
4. Atributos de Calidad del Software.....	33
4.1. Escenario de Usabilidad.....	33
4.2. Escenario de Confiabilidad.....	33
4.3. Escenario de Rendimiento.....	34
4.4. Otros Escenarios.....	34



## **Documento de Arquitectura de Software**

### **1. Introducción**

El presente Documento de Arquitectura de Software (SAD) describe la arquitectura de la plataforma LinkJob, una solución web basada en inteligencia artificial destinada a optimizar la vinculación laboral entre egresados de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Privada de Tacna y empresas registradas.

La arquitectura propuesta sigue el modelo 4+1 de vistas de Kruchten, el cual permite analizar el sistema desde múltiples perspectivas complementarias: lógica, de procesos, de implementación, de despliegue y de casos de uso. Este enfoque garantiza una visión completa de los componentes técnicos y funcionales de la solución, permitiendo una comprensión integral tanto para desarrolladores como para otros interesados clave.

LinkJob ha sido concebido para atender una necesidad crítica de inserción laboral efectiva, considerando la relevancia de criterios como las habilidades técnicas, experiencia laboral, certificaciones y otros atributos clave de los egresados. La plataforma se apoya en técnicas de procesamiento de lenguaje natural y modelos de embeddings semánticos para mejorar la precisión en la recomendación de perfiles, asegurando una coincidencia significativa entre las ofertas laborales y los candidatos.

Este documento no solo describe la estructura y componentes técnicos del sistema, sino que también especifica los objetivos arquitectónicos, restricciones, escenarios de calidad y representaciones visuales necesarias para su implementación efectiva.

A través de este SAD, se pretende facilitar la toma de decisiones técnicas y estratégicas durante el ciclo de vida del proyecto, proporcionando una guía sólida para el desarrollo, evolución y mantenimiento de LinkJob.

## 1.1. Propósito (Diagrama 4+1)

La arquitectura propuesta busca satisfacer tanto los requisitos funcionales como los atributos de calidad definidos, tales como eficiencia, escalabilidad, mantenibilidad, seguridad y confiabilidad. El diseño se basa en principios de modularidad y separación de responsabilidades, con una clara distinción entre las capas del sistema (frontend, backend, servicio de IA y almacenamiento de vectores).

El sistema utiliza técnicas modernas como la búsqueda semántica con embeddings para emparejar de manera inteligente perfiles de egresados con ofertas laborales, considerando habilidades, experiencia, certificaciones, idiomas y proyectos. Los embeddings se almacenan y consultan mediante Upstash Vector, lo que permite realizar búsquedas vectoriales eficientes y en tiempo real.

### Objetivos de Diseño:

- ❖ *Eficiencia y escalabilidad:* Diseñar una plataforma capaz de soportar un volumen creciente de usuarios y datos sin degradación del rendimiento.
- ❖ *Mantenibilidad y modularidad:* Aplicar una arquitectura desacoplada que facilite futuras mejoras, pruebas y mantenimientos.
- ❖ *Seguimiento del impacto:* Permitir a los administradores hacer seguimiento a la inserción laboral de los egresados recomendados, midiendo así el impacto del sistema.
- ❖ *Captura estructurada de datos:* Registrar datos clave de los egresados de forma estructurada para su posterior análisis automático.
- ❖ *Recomendación inteligente:* Implementar un motor de emparejamiento basado en similitud semántica para mejorar la precisión de las postulaciones.
- ❖ *Tecnologías de IA eficientes:* Utilizar herramientas especializadas como Upstash Vector para asegurar velocidad y precisión en el procesamiento de datos vectoriales.
- ❖ *Seguridad de datos:* Proteger la información sensible de los egresados y empresas mediante controles de acceso, cifrado y buenas prácticas de seguridad.

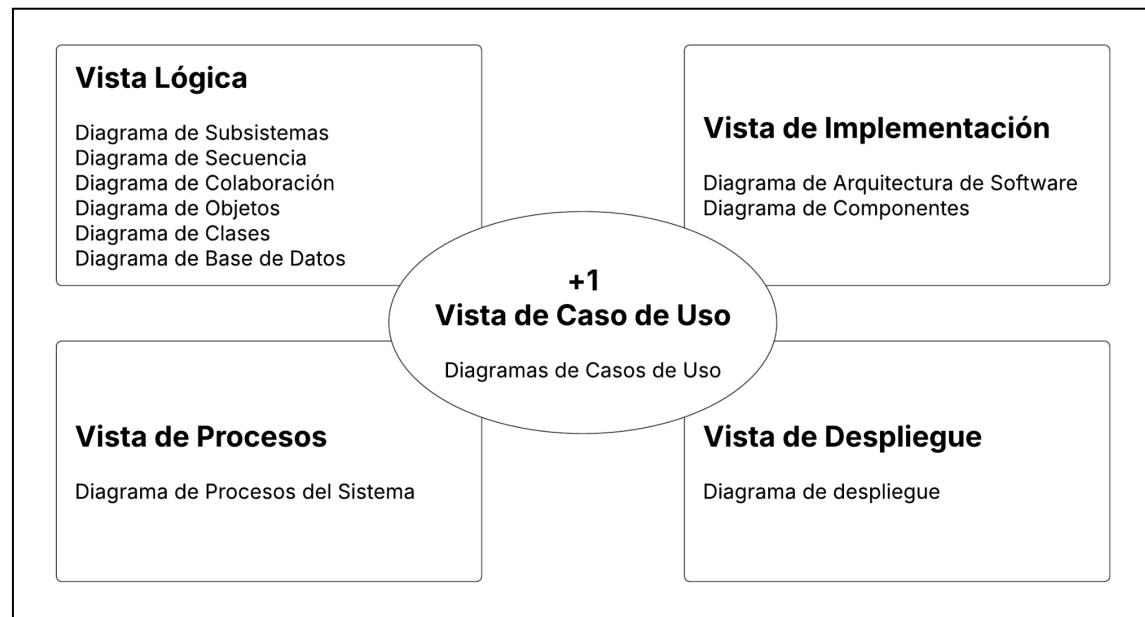
### Decisiones y Prioridades Arquitectónicas:

Durante el diseño del sistema, se establecieron las siguientes prioridades:

- ❖ *Rendimiento vs. Portabilidad:* Se priorizó el rendimiento del sistema, optando por tecnologías y servicios optimizados para IA (como Upstash Vector y bases de datos en la nube), sacrificando en parte la portabilidad total en favor de eficiencia.
- ❖ *Mantenibilidad vs. Complejidad inicial:* Se eligió una arquitectura modular más compleja inicialmente, con la finalidad de garantizar un sistema mantenable y escalable en el tiempo.
- ❖ *Seguridad vs. Accesibilidad:* Se prioriza la protección de los datos sobre la apertura del sistema, exigiendo autenticación y control de acceso en todos los puntos críticos del flujo.

### Diagrama 4+1:

El diagrama 4+1 del sistema muestra su arquitectura desde cinco perspectivas: casos de uso, lógica del negocio, estructura del código, procesos concurrentes y despliegue físico en la infraestructura.



**Diagrama 01.** Diagrama 4+1 de la aplicación web LinkJob.

*Fuente: Elaboración propia.*

## 1.2. Alcance

Este documento se centra principalmente en la descripción detallada de la vista lógica del sistema, dado que representa la estructura interna y funcional del software. También se incluyen elementos representativos de las vistas de implementación, despliegue y casos de uso, necesarias para comprender el contexto completo del sistema. La vista de procesos se menciona brevemente, al no ser prioritaria para este proyecto.

### Inclusiones

- ***Desarrollo de una Plataforma Web:***

- ❖ Creación de una interfaz de usuario intuitiva, accesible y eficiente para los administradores de la Escuela Profesional de Ingeniería de Sistemas y las empresas que deseen publicar oportunidades laborales.
- ❖ Implementación de un sistema de autenticación y registro de usuarios con roles diferenciados, donde los administradores de la Escuela gestionan el sistema, mientras que las empresas podrán enviar plazas de trabajo que requieran aprobación de los administradores antes de ser publicadas.

- ***Funcionalidades Clave:***

- ❖ Solo los administradores pueden registrar y actualizar información de empresas (nombre, RUC, teléfono, email, logo, etc.).
- ❖ Registro masivo de egresados por parte de los administradores cada 6 meses, almacenando datos clave como habilidades, experiencia, certificaciones y proyectos.
- ❖ Los administradores pueden crear y publicar plazas laborales directamente.
- ❖ Las empresas pueden registrar plazas laborales, pero estas deben ser aprobadas por los administradores antes de publicarse.
- ❖ Los administradores pueden visualizar y gestionar las recomendaciones generadas por el sistema.
- ❖ Las empresas pueden consultar el estado de sus plazas laborales enviadas (pendiente, aprobada, rechazada).

- ❖ Implementación de algoritmos de búsqueda semántica y embeddings para analizar habilidades, certificaciones, experiencia, etc. de los egresados y generar recomendaciones precisas.
- ❖ Generación de un ranking de egresados recomendados por plaza laboral, priorizando perfiles similares a los contratados por empresas del mismo sector.
- ❖ Registro de historial de contrataciones para mejorar futuras recomendaciones.
- ❖ Análisis de contrataciones previas para optimizar la precisión del sistema de recomendación.
- ❖ Generación de reportes para los administradores sobre la tasa de éxito en contrataciones, tendencias del mercado laboral y cantidad de egresados recomendados.

- **Soporte Tecnológico:**

- ❖ Integración de tecnologías de inteligencia artificial y bases de datos vectoriales para la gestión y análisis de recomendaciones.

### Exclusiones

- **Acceso Directo de los Egresados a la Plataforma:**

- ❖ Los egresados no podrán gestionar directamente su perfil ni postularse a plazas laborales; el proceso será gestionado exclusivamente por los administradores.

- **Servicios de Búsqueda de Empleo Abiertos:**

- ❖ No se incluirá un portal público de empleo para que cualquier empresa u egresado pueda registrarse y postular libremente.

- **Asesoramiento Laboral o Capacitación Adicional:**

- ❖ No se ofrecerán servicios de orientación profesional, asesoramiento en entrevistas o cursos de capacitación dentro de la plataforma.

### 1.3. Definiciones, Siglas y Abreviaturas

Término / Sigla	Definición
API	Interfaz de Programación de Aplicaciones (Application Programming Interface).
Backend	Parte del sistema que gestiona la lógica de negocio, datos y reglas.
Base de datos	Sistema que almacena de forma estructurada la información del sistema.
Caso de uso	Representación de una funcionalidad del sistema desde la perspectiva del usuario.
Diagrama de clases	Diagrama UML que muestra las clases del sistema y sus relaciones.
Diagrama de componentes	Diagrama UML que representa los componentes de software y sus interacciones.
Diagrama de despliegue	Diagrama UML que muestra la disposición física de los componentes en hardware.
Diagrama de paquetes	Diagrama UML que agrupa clases o componentes relacionados lógicamente.
Embedding	Representación vectorial densa de un texto, utilizada para análisis semántico.
Frontend	Interfaz gráfica con la que interactúan los usuarios.
LinkJob	Nombre de la aplicación web para recomendación laboral de egresados.
Postulación	Acción mediante la cual un egresado es propuesto para una oferta laboral.
Render	Plataforma de despliegue de aplicaciones web en la nube.
Upstash Vector	Servicio en la nube para almacenamiento y búsqueda de vectores.
Usuario	Persona que accede al sistema (Administrador o Empresa).

## 1.4. Organización del documento

Este documento está estructurado en varias secciones que cubren todos los aspectos clave del desarrollo y diseño de la arquitectura del sistema.

- **Sección 1:** Introducción

En esta sección se define el propósito, alcance, y las definiciones clave del sistema, proporcionando una base para el resto del documento.

- **Sección 2:** Objetivos y Restricciones Arquitectónicas

Aquí se detallan los objetivos del sistema, así como las restricciones y prioridades arquitectónicas que guiarán el diseño.

- **Sección 3:** Representación de la Arquitectura del Sistema

Esta sección cubre las diferentes vistas del sistema: caso de uso, lógica, implementación, procesos y despliegue, mediante diagramas y descripciones.

- **Sección 4:** Atributos de Calidad del Software

Se incluyen los escenarios que abordan los atributos de calidad del sistema, tales como funcionalidad, usabilidad, confiabilidad, rendimiento y mantenibilidad.

## 2. Objetivos y Restricciones Arquitectónicas

### 2.1. Priorización de requerimientos

A continuación, se presentan los requerimientos funcionales y no funcionales priorizados, lo que permitirá establecer un orden lógico de implementación de las funcionalidades y atributos de calidad del sistema.

#### 2.1.1. Requerimientos Funcionales

ID	Requerimiento	Prioridad
RF-01	Gestionar autenticación y credenciales de usuario	Alta
RF-02	Gestionar empresas	Media
RF-03	Gestionar ofertas de trabajo	Alta
RF-04	Gestionar egresados	Alta

RF-05	Recomendar egresados con IA	Alta
RF-06	Generar ranking de egresados recomendados	Alta
RF-07	Gestionar postulaciones de egresados	Alta
RF-08	Almacenar historial de contrataciones	Media
RF-09	Analizar las contrataciones previas	Media
RF-10	Visualizar reportes de empleabilidad	Media

### 2.1.2. Requerimientos No Funcionales - Atributos de Calidad

ID	Descripción	Prioridad
RNF-01	Garantizar disponibilidad del sistema ( $\geq 98\%$ ) con monitoreo del uptime	Alta
RNF-02	Seguridad en la plataforma, con pruebas de penetración y sin vulnerabilidades críticas	Alta
RNF-03	Escalabilidad horizontal del sistema, permitiendo crecimiento en usuarios sin degradación significativa del rendimiento	Media
RNF-04	Usabilidad de la plataforma, evaluada mediante encuestas de satisfacción con resultados $\geq 80\%$ en puntuación alta	Alta

### 2.2. Restricciones

Las restricciones arquitectónicas establecen condiciones fijas que deben cumplirse durante el desarrollo del sistema, limitando las decisiones de diseño y las tecnologías a emplear. Estas restricciones no son negociables y deben respetarse a lo largo de todo el ciclo de vida del proyecto. Para este sistema se han definido las siguientes:

- Lenguaje y Framework de Backend: Se emplea exclusivamente FastAPI (Python) como framework principal para la construcción del backend.
- Lenguaje y Framework de Frontend: La interfaz de usuario será desarrollada utilizando Next.js, lo cual proporciona una experiencia más ágil y moderna en el desarrollo frontend.

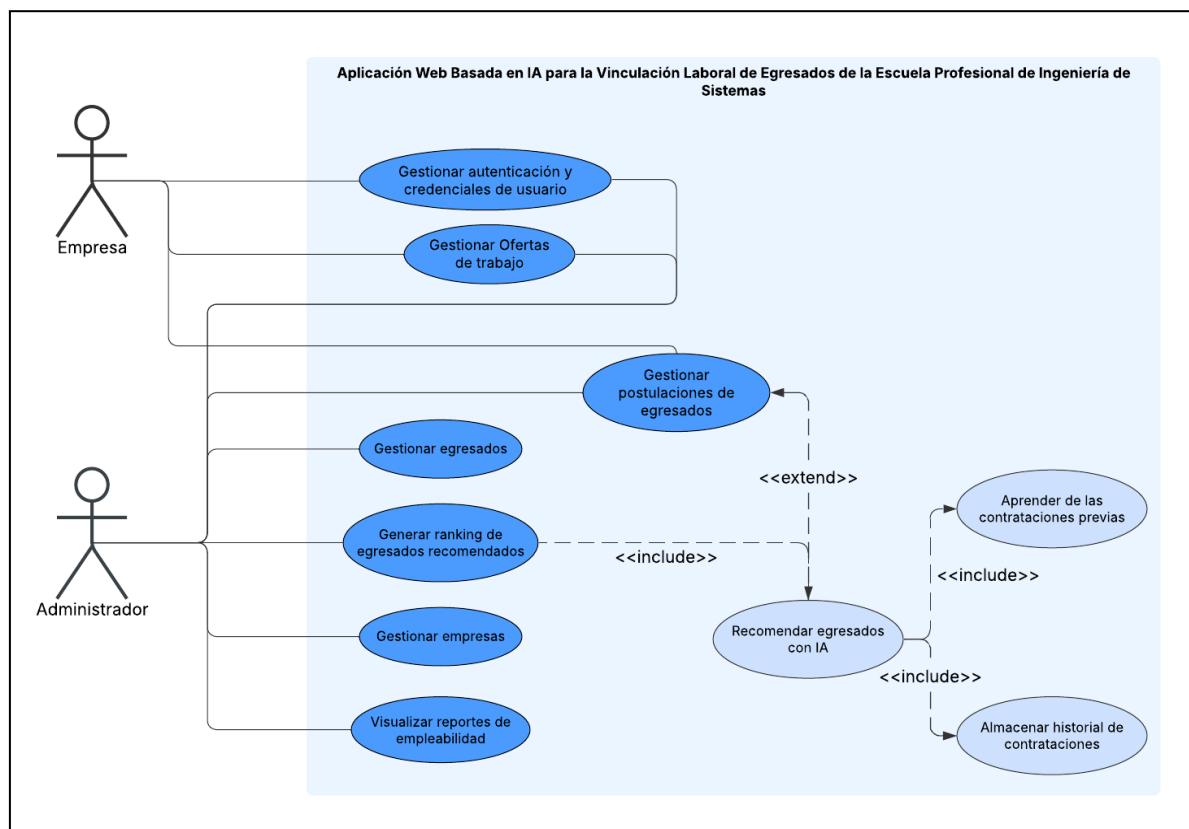
- Recuperación de contraseña: Se usará Redis como sistema de almacenamiento temporal para gestionar los tokens de recuperación de contraseña de forma segura y eficiente.
- Base de datos: Se utilizará una base de datos relacional en la nube (NeonTech) para almacenar de forma estructurada la información del sistema.
- Almacenamiento de vectores: Para la recomendación de egresados, se usará Upstash Vector para almacenar embeddings generados mediante modelos de lenguaje.
- Infraestructura: El backend será desplegado en Render, y el frontend en un servidor Debian 12, contratado a través de Elastika.
- Motor de recomendación: Se utilizará el modelo paraphrase-MiniLM-L6-v2 de sentence-transformers para generar y comparar embeddings semánticos.
- Monitoreo de disponibilidad: Se utilizará Uptime Kuma para realizar seguimiento continuo del uptime del sistema y asegurar que se mantenga por encima del 98% anual.
- Pruebas de seguridad: OWASP ZAP será utilizada para realizar pruebas de penetración automatizadas y detectar vulnerabilidades comunes en la aplicación.
- Pruebas de carga y estrés: Se utilizará Locust para simular usuarios concurrentes y evaluar el rendimiento ante incrementos de carga.

### 3. Representación de la Arquitectura del Sistema

#### 3.1. Vista de Caso de Uso

##### 3.1.1. Diagramas de Casos de Uso - ([Link a Diagrama de Caso de Uso](#))

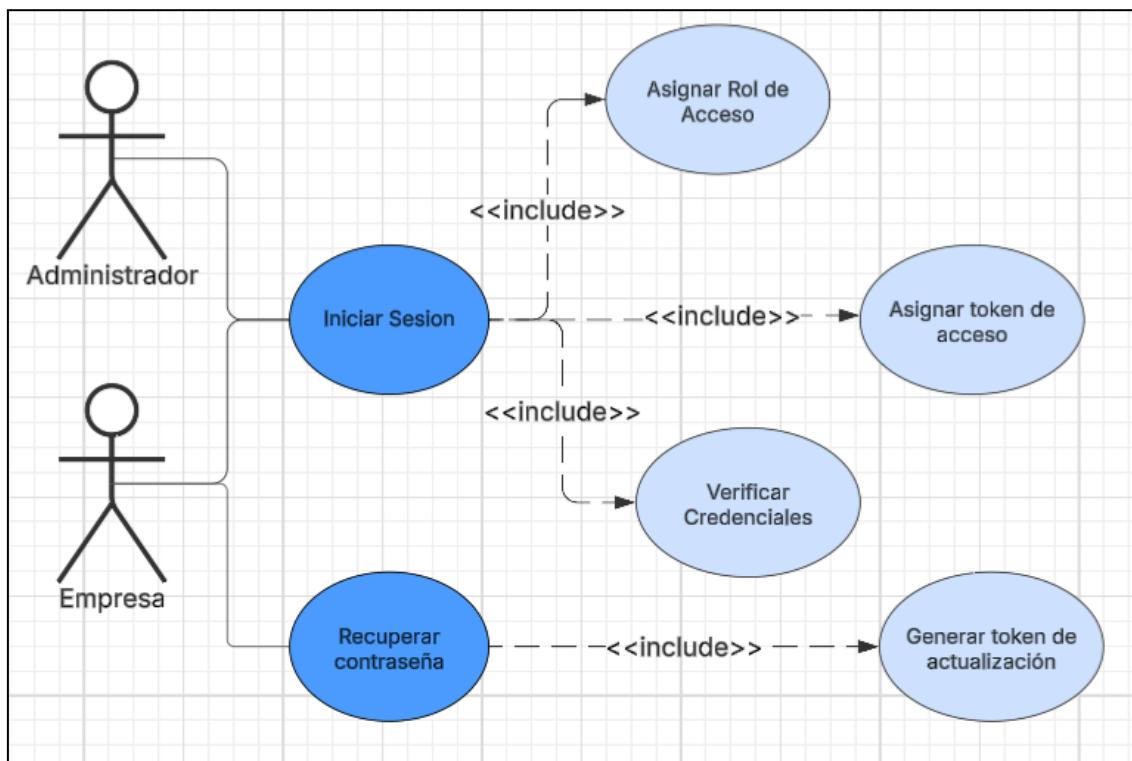
El diagrama de casos de uso de JobLink muestra las interacciones entre los diferentes actores y las funcionalidades del sistema.



**Diagrama 02.** Diagrama de Casos de Uso de la plataforma LinkJob.

**Fuente:** Elaboración propia.

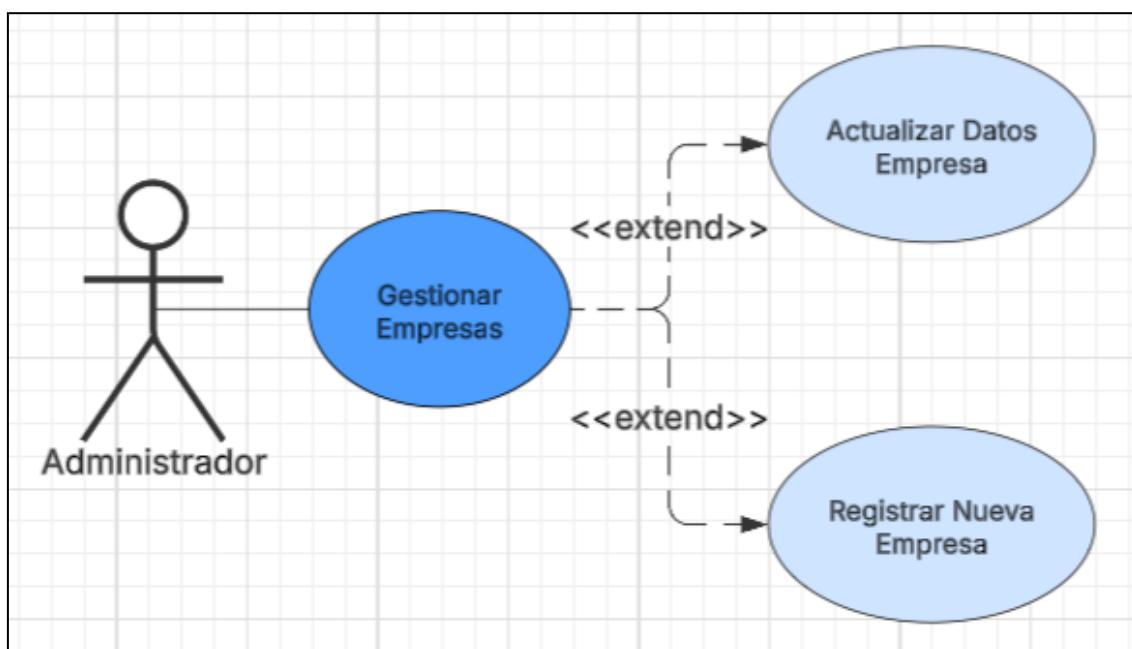
### CU - 01: "Gestionar Autenticación y Credenciales de Usuario"



**Diagrama 03.** Diagrama de Casos de Uso del CU - 01 de la plataforma LinkJob.

*Fuente:* Elaboración propia.

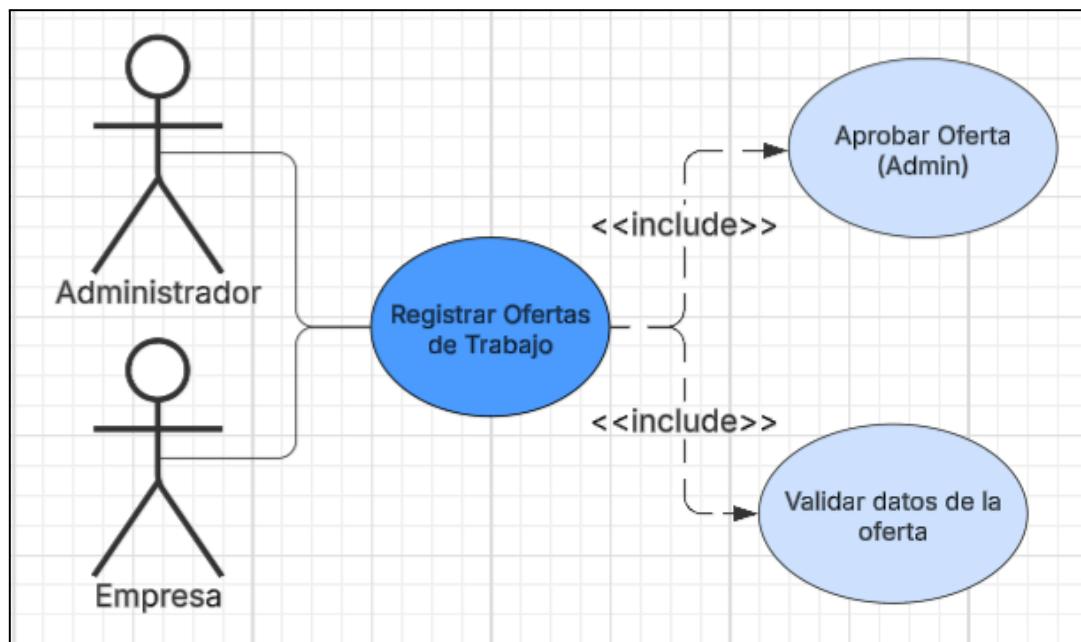
### CU - 02: "Gestionar Empresas"



**Diagrama 04.** Diagrama de Casos de Uso del CU - 02 de la plataforma LinkJob.

*Fuente:* Elaboración propia.

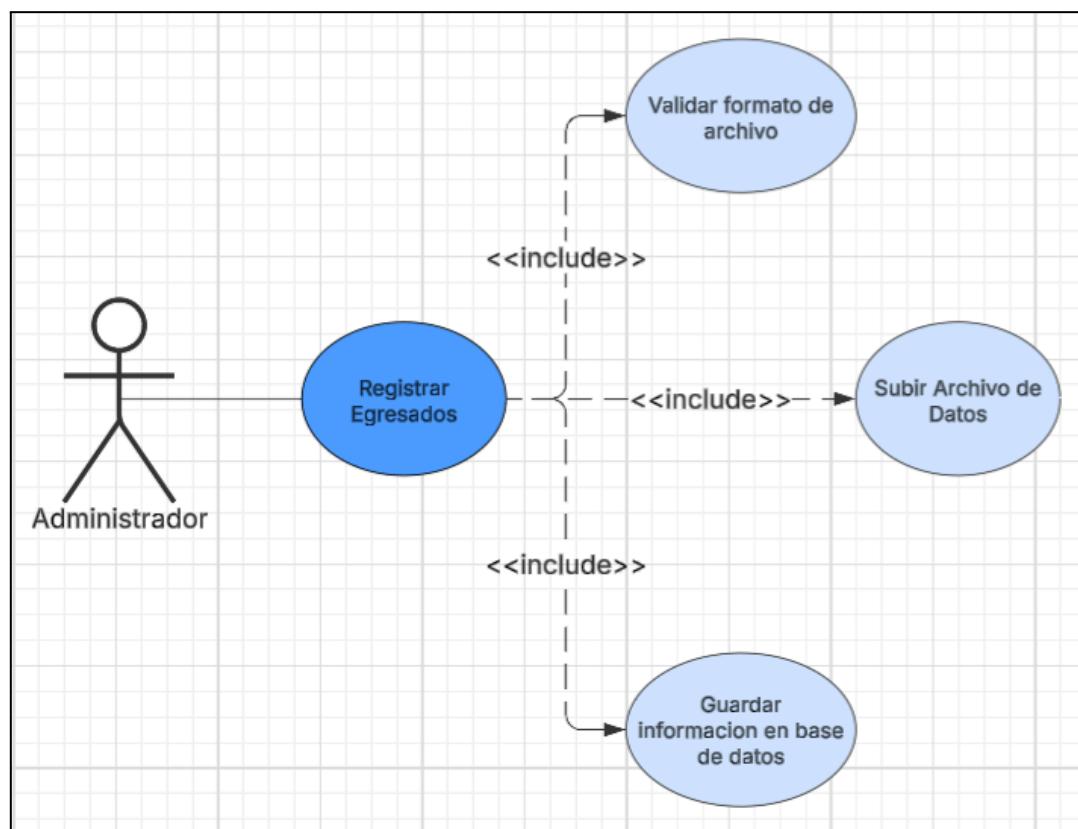
### CU - 03: “Gestionar Ofertas de Trabajo”



**Diagrama 05.** Diagrama de Casos de Uso del CU - 03 de la plataforma LinkJob.

*Fuente:* Elaboración propia.

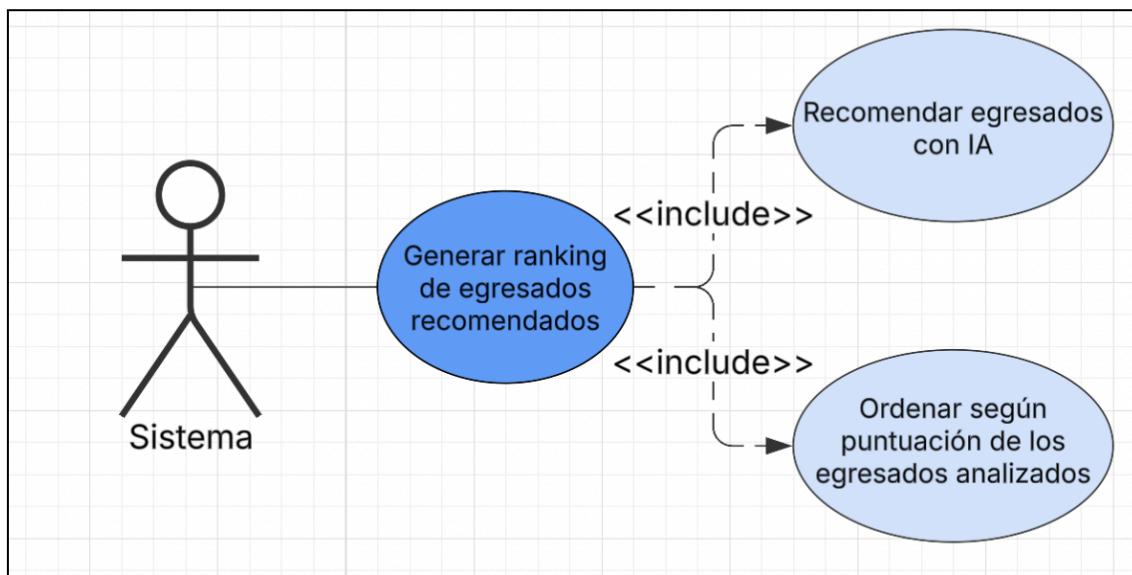
### CU - 04: “Gestionar Egresados”



**Diagrama 06.** Diagrama de Casos de Uso del CU - 04 de la plataforma LinkJob.

*Fuente:* Elaboración propia.

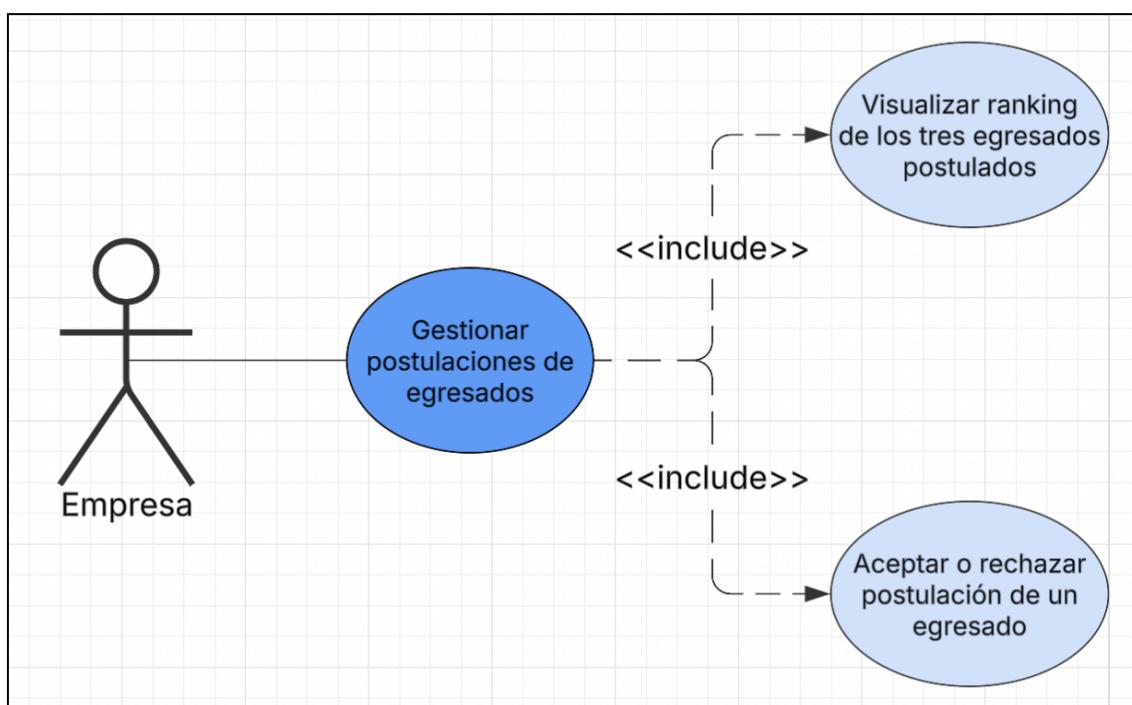
**CU - 05:** “Generar Ranking de Egresados Recomendados”



**Diagrama 07.** Diagrama de Casos de Uso del CU - 05 de la plataforma LinkJob.

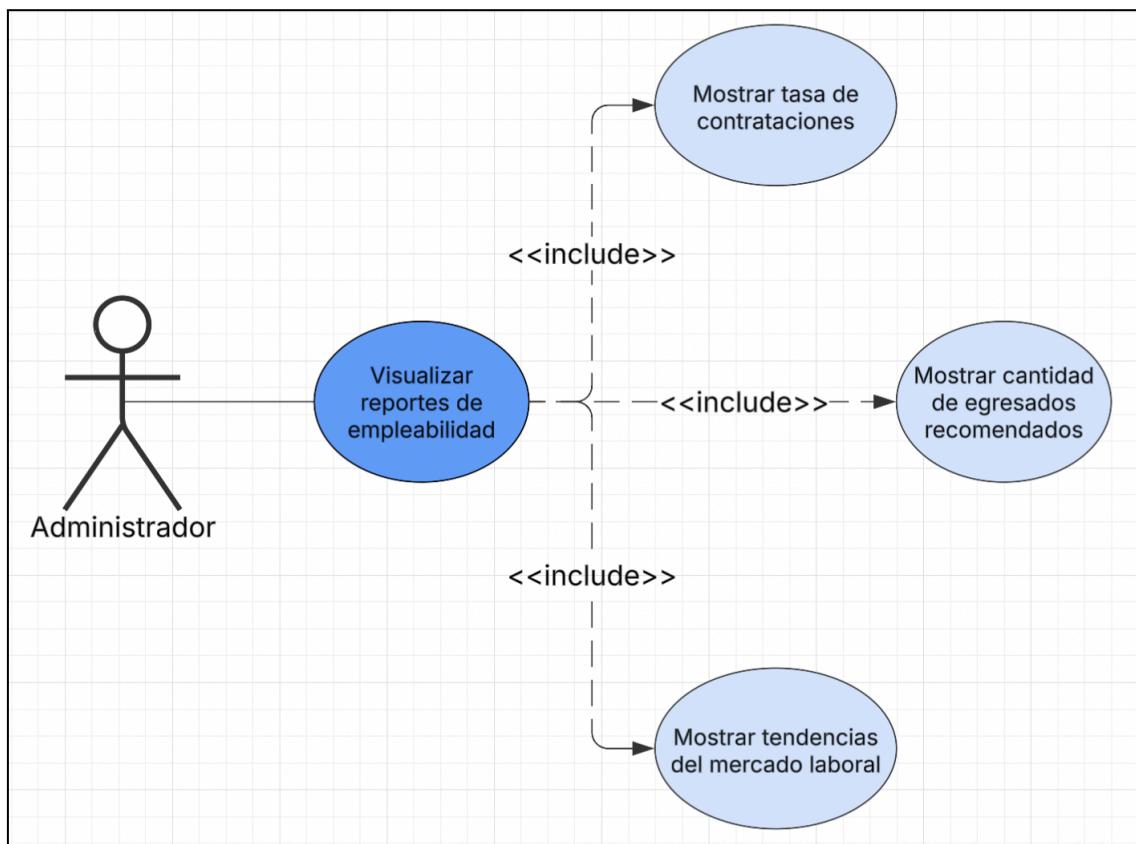
*Fuente:* Elaboración propia.

**CU - 06:** “Gestionar Postulaciones de Egresados”



**Diagrama 08.** Diagrama de Casos de Uso del CU - 06 de la plataforma LinkJob.

*Fuente:* Elaboración propia.

**CU - 07:** “Visualizar Reportes de Empleabilidad”

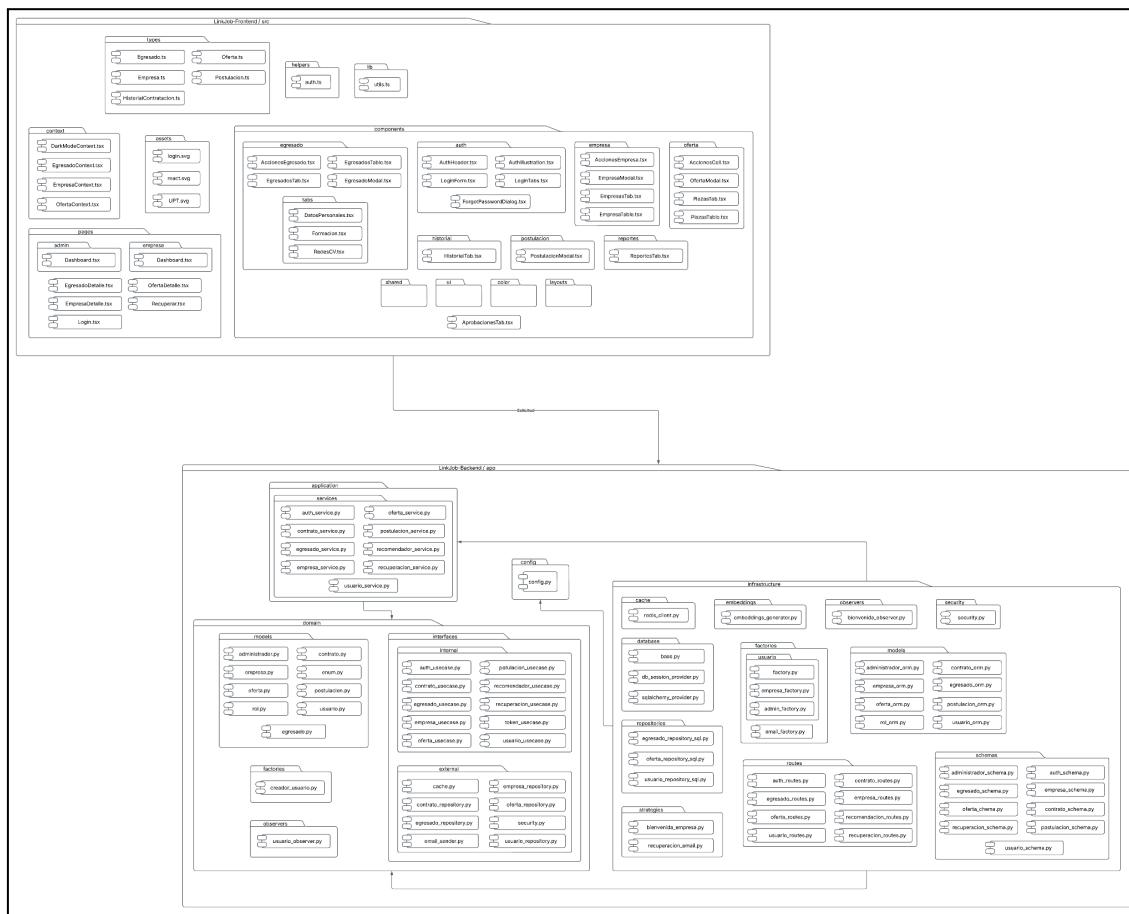
**Diagrama 09.** Diagrama de Casos de Uso del CU - 07 de la plataforma LinkJob.

*Fuente:* Elaboración propia.

## 3.2. Vista de Lógica

### 3.2.1. Diagrama de Subsistemas (paquetes) - ([Link al Diagrama](#))

El diagrama de subsistemas de JobLink muestra la organización e interacción de carpetas dentro del frontend y backend de acuerdo a las funcionalidades del sistema.

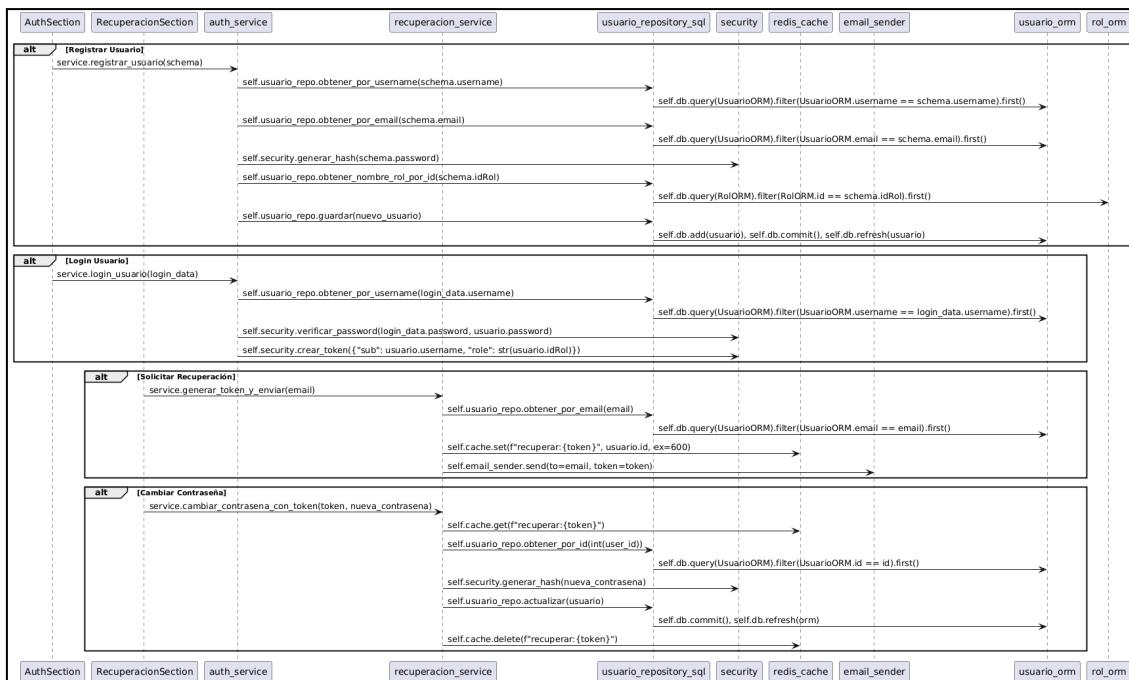


**Diagrama 10.** Diagrama de Subsistemas de la plataforma LinkJob.

**Fuente:** Elaboración propia.

### 3.2.2. Diagrama de Secuencia (vista de diseño)

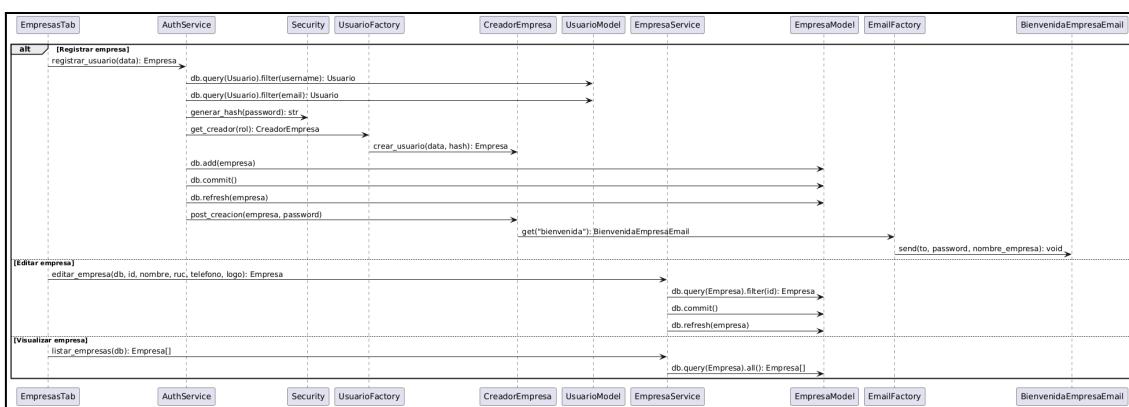
**CU - 01:** “Gestionar Autenticación y Credenciales de Usuario”



**Diagrama 11.** Diagrama de secuencia del CU - 01 de la plataforma LinkJob.

*Fuente:* Elaboración propia.

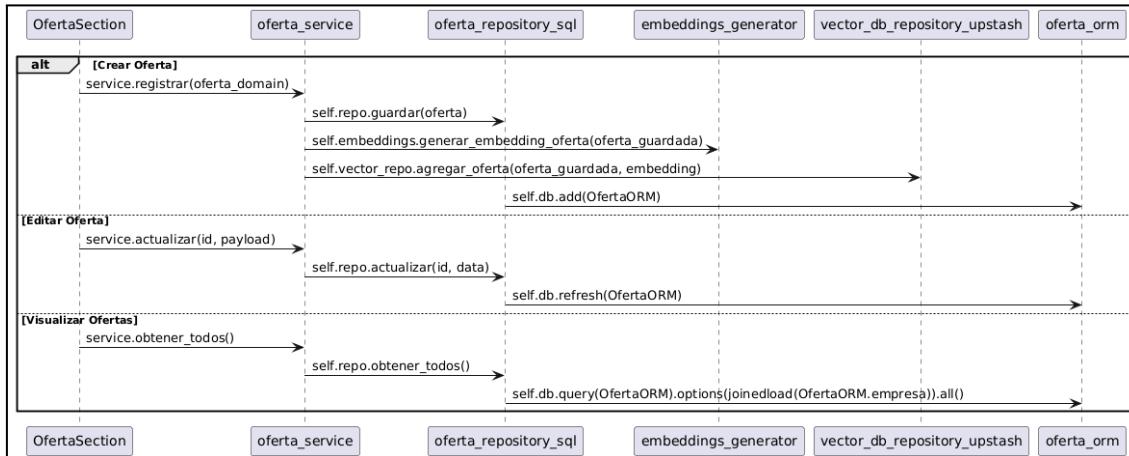
**CU - 02:** “Gestionar Empresas”



**Diagrama 12.** Diagrama de secuencia del CU - 02 de la plataforma LinkJob.

*Fuente:* Elaboración propia.

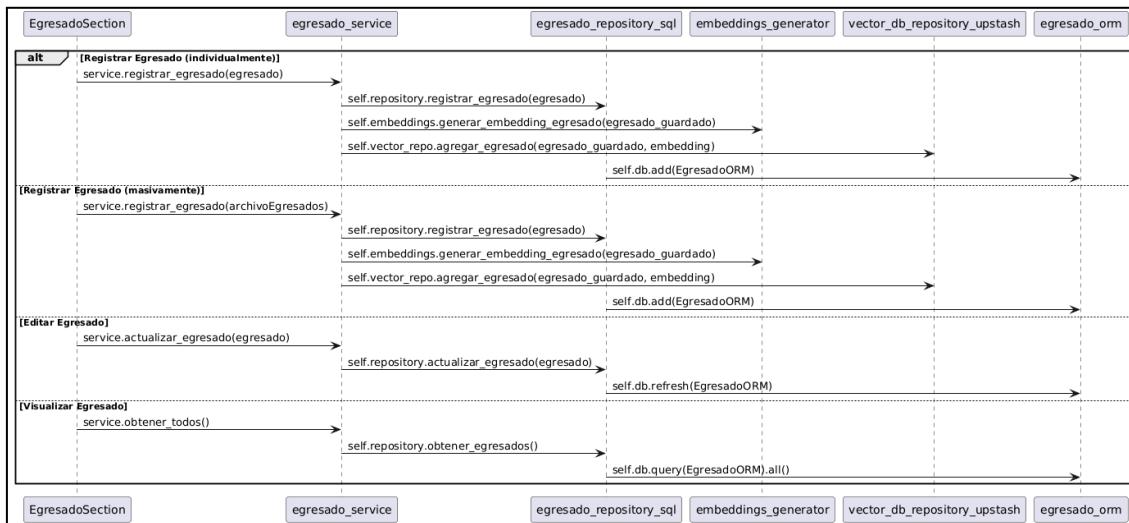
### CU - 03: "Gestionar Ofertas de Trabajo"



**Diagrama 13.** Diagrama de secuencia del CU - 03 de la plataforma LinkJob.

*Fuente:* Elaboración propia.

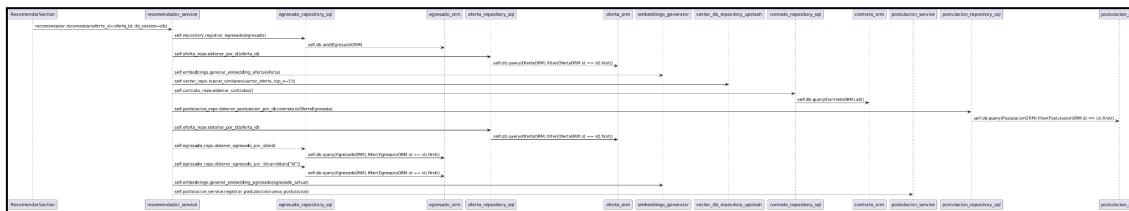
### CU - 04: "Gestionar Egresados"



**Diagrama 14.** Diagrama de secuencia del CU - 04 de la plataforma LinkJob.

*Fuente:* Elaboración propia.

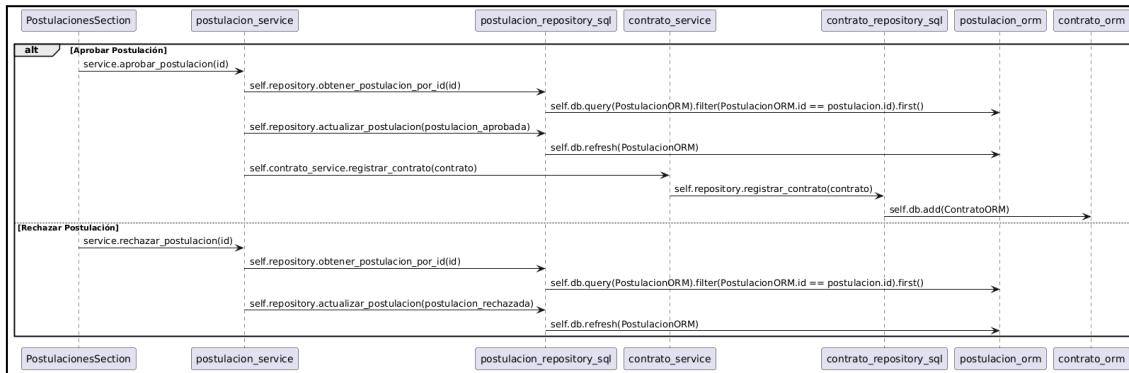
### CU - 05: "Generar Ranking de Egresados Recomendados"



**Diagrama 15.** Diagrama de secuencia del CU - 05 de la plataforma LinkJob.

*Fuente:* Elaboración propia.

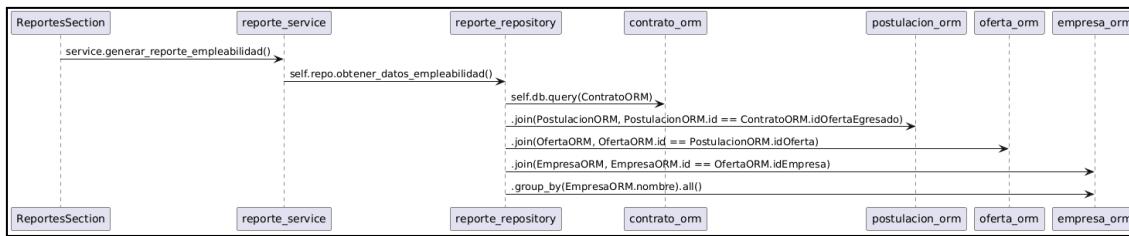
### CU - 06: “Gestionar Postulaciones de Egresados”



**Diagrama 16.** Diagrama de secuencia del CU - 06 de la plataforma LinkJob.

*Fuente:* Elaboración propia.

### CU - 07: “Visualizar Reportes de Empleabilidad”



**Diagrama 17.** Diagrama de secuencia del CU - 07 de la plataforma LinkJob.

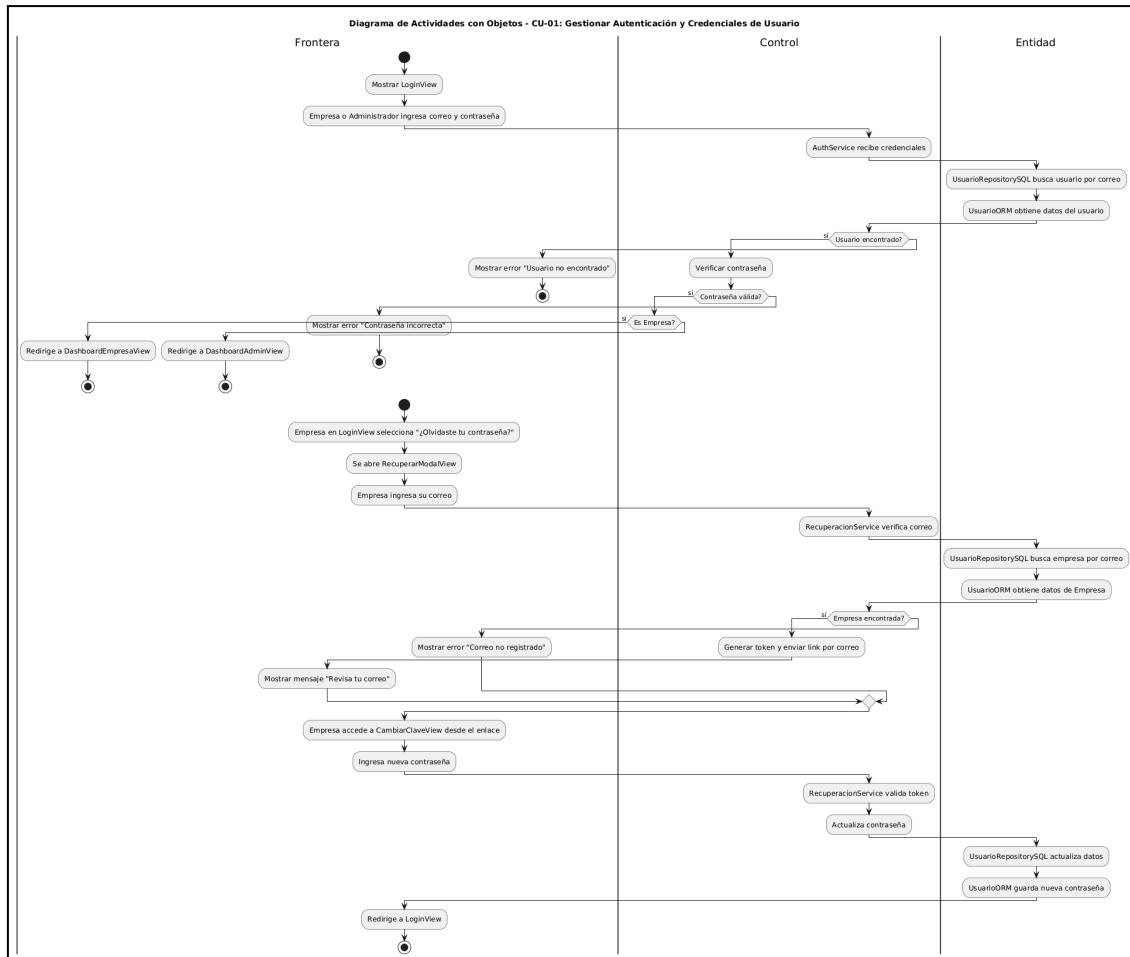
*Fuente:* Elaboración propia.

#### 3.2.3. Diagrama de Colaboración (vista de diseño)

En proceso.

### 3.2.4. Diagrama de Actividades con Objetos

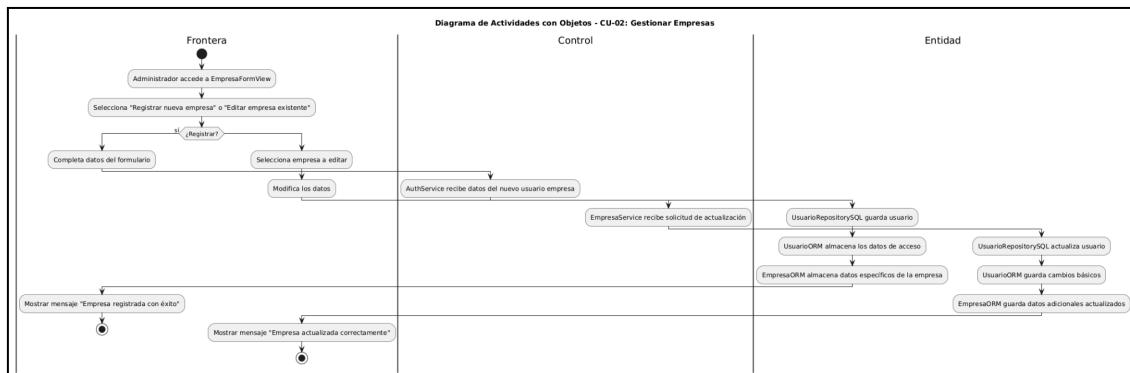
**CU - 01:** “Gestionar Autenticación y Credenciales de Usuario”



**Diagrama 19.** Diagrama de actividades del CU - 01 de la plataforma LinkJob.

*Fuente:* Elaboración propia.

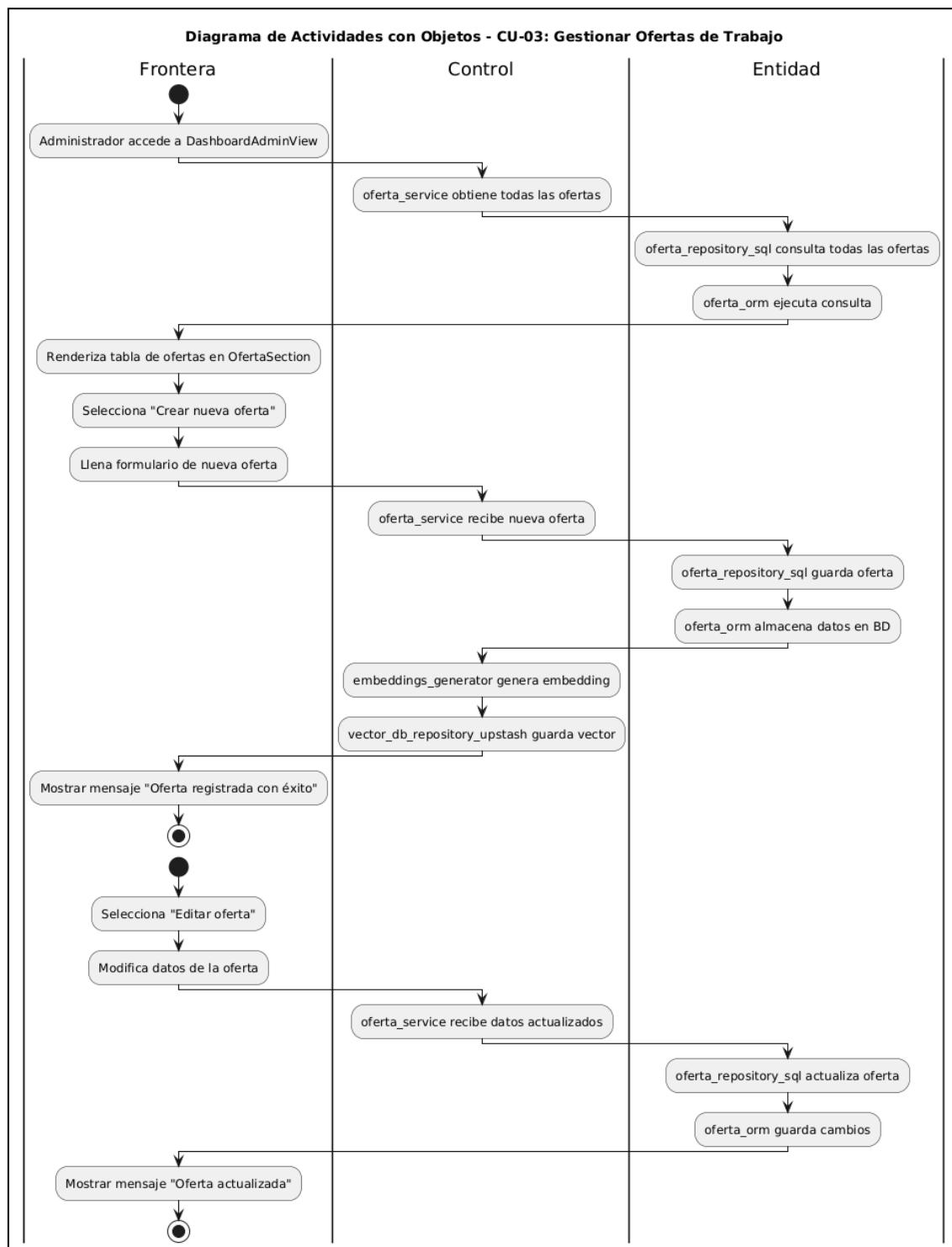
**CU - 02:** “Gestionar Empresas”



**Diagrama 20.** Diagrama de actividades del CU - 02 de la plataforma LinkJob.

*Fuente:* Elaboración propia.

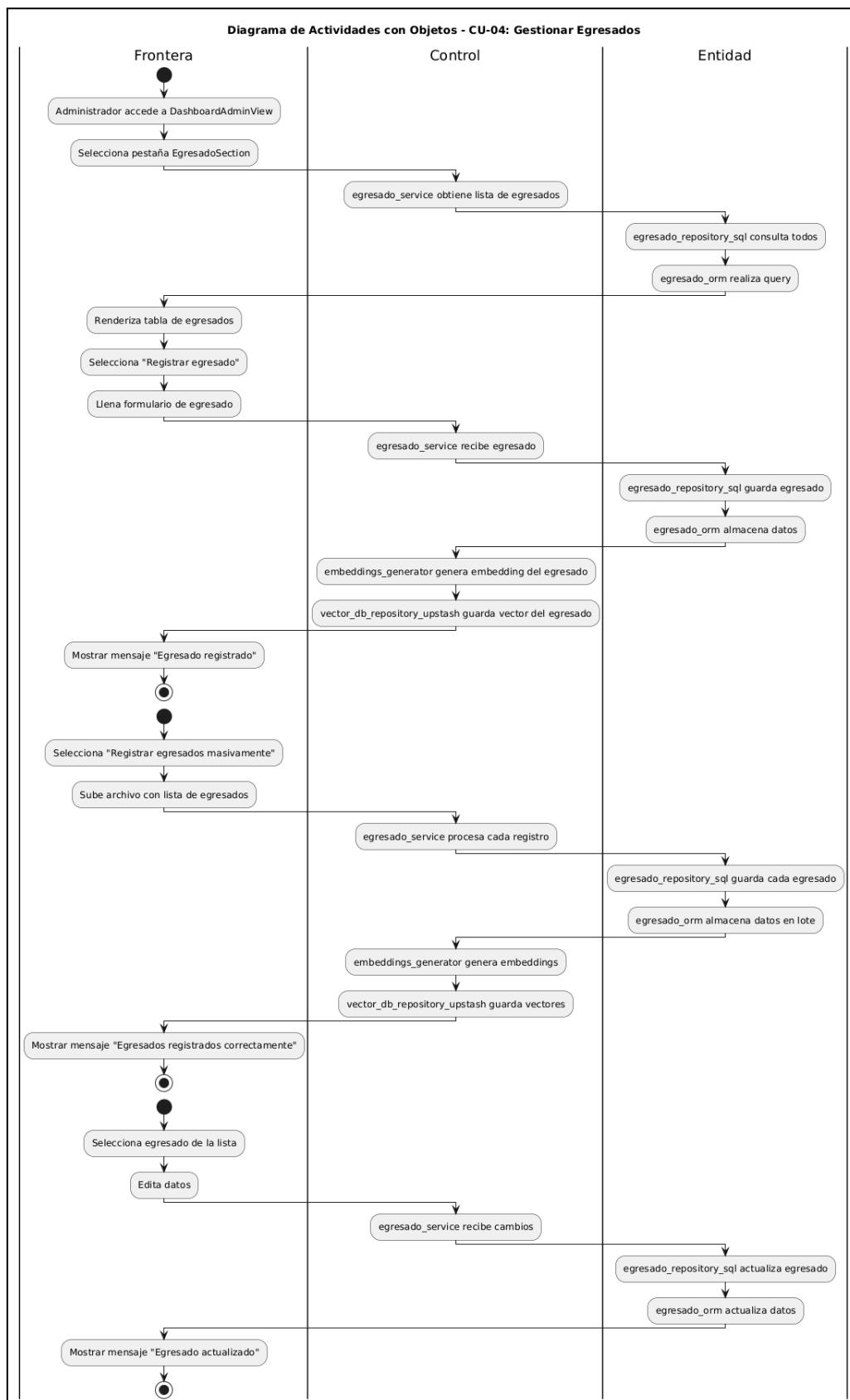
### CU - 03: "Gestionar Ofertas de Trabajo"



**Diagrama 21.** Diagrama de actividades del CU - 03 de la plataforma LinkJob.

*Fuente:* Elaboración propia.

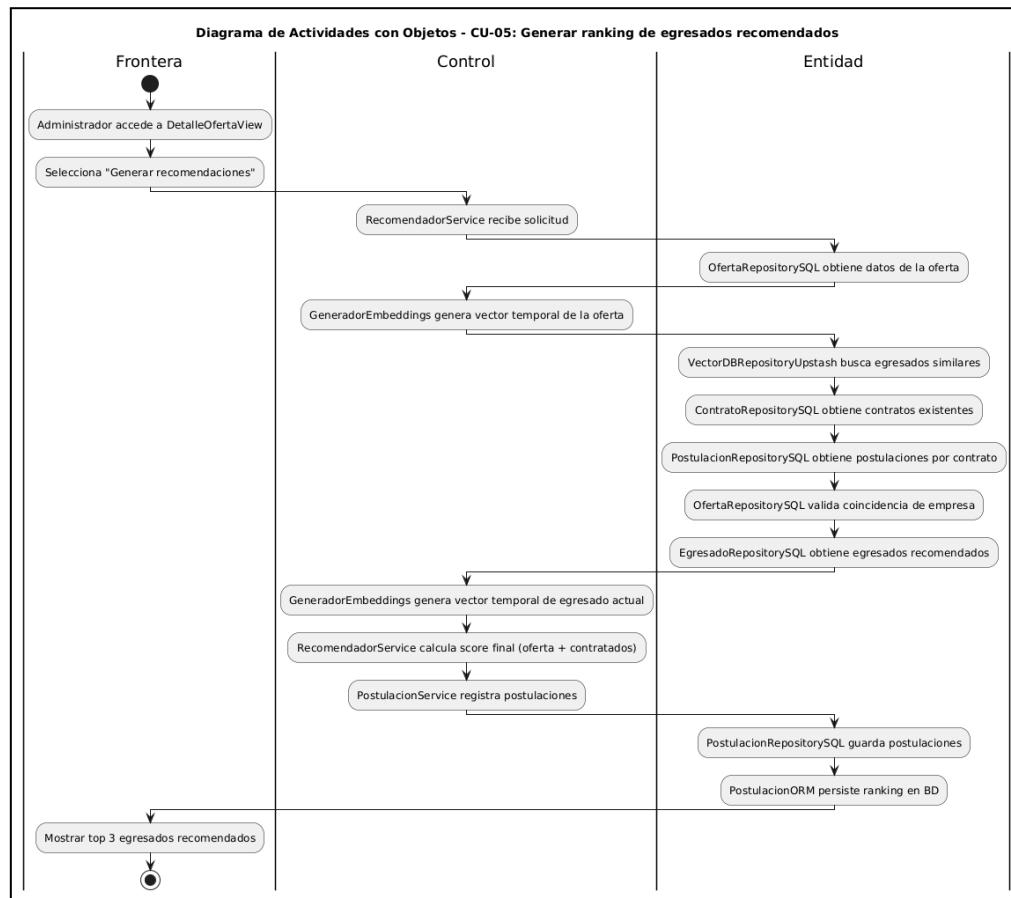
## CU - 04: "Gestionar Egresados"



**Diagrama 22.** Diagrama de actividades del CU - 04 de la plataforma LinkJob.

*Fuente:* Elaboración propia.

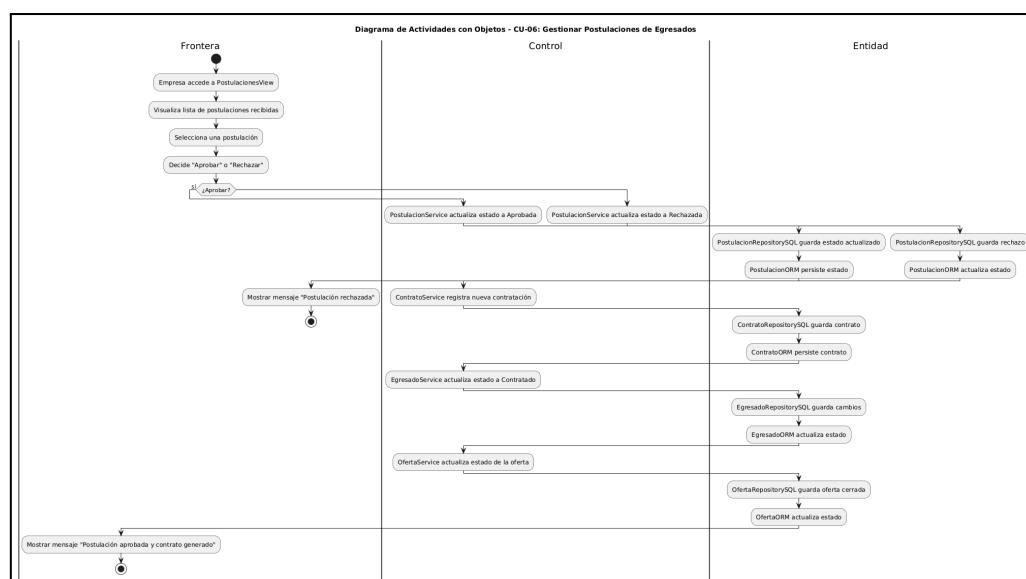
### **CU - 05: "Generar Ranking de Egresados Recomendados"**



**Diagrama 23.** Diagrama de actividades del CU - 05 de la plataforma LinkJob.

**Fuente:** Elaboración propia.

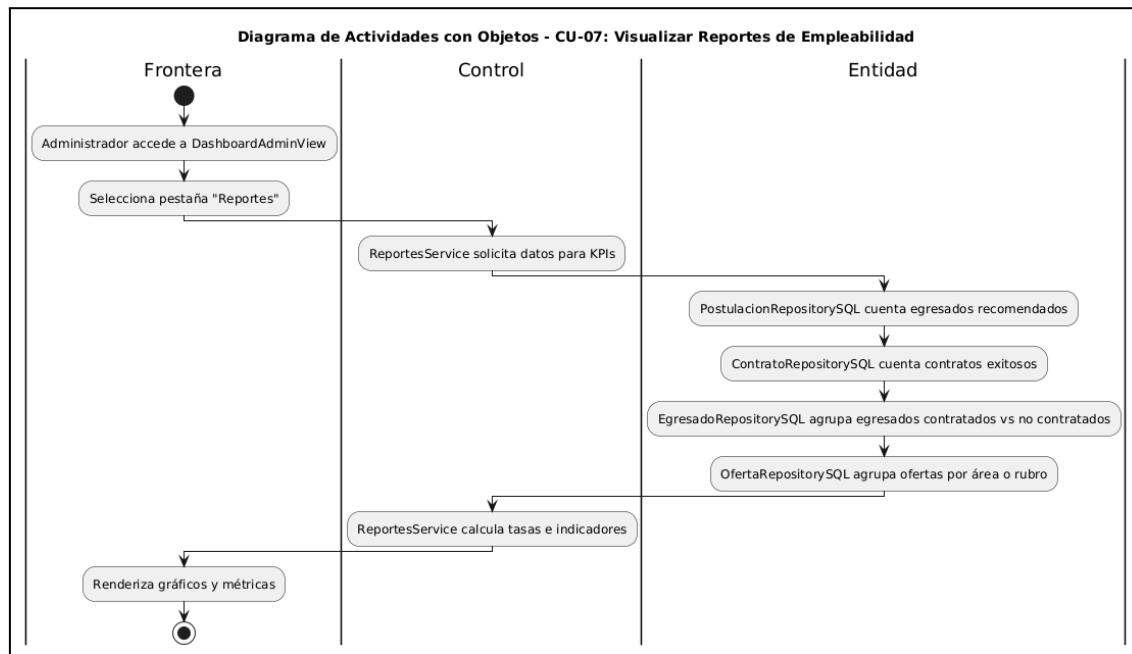
### **CU - 06: "Gestionar Postulaciones de Egresados"**



**Diagrama 24.** Diagrama de actividades del CU - 06 de la plataforma LinkJob.

**Fuente:** Elaboración propia.

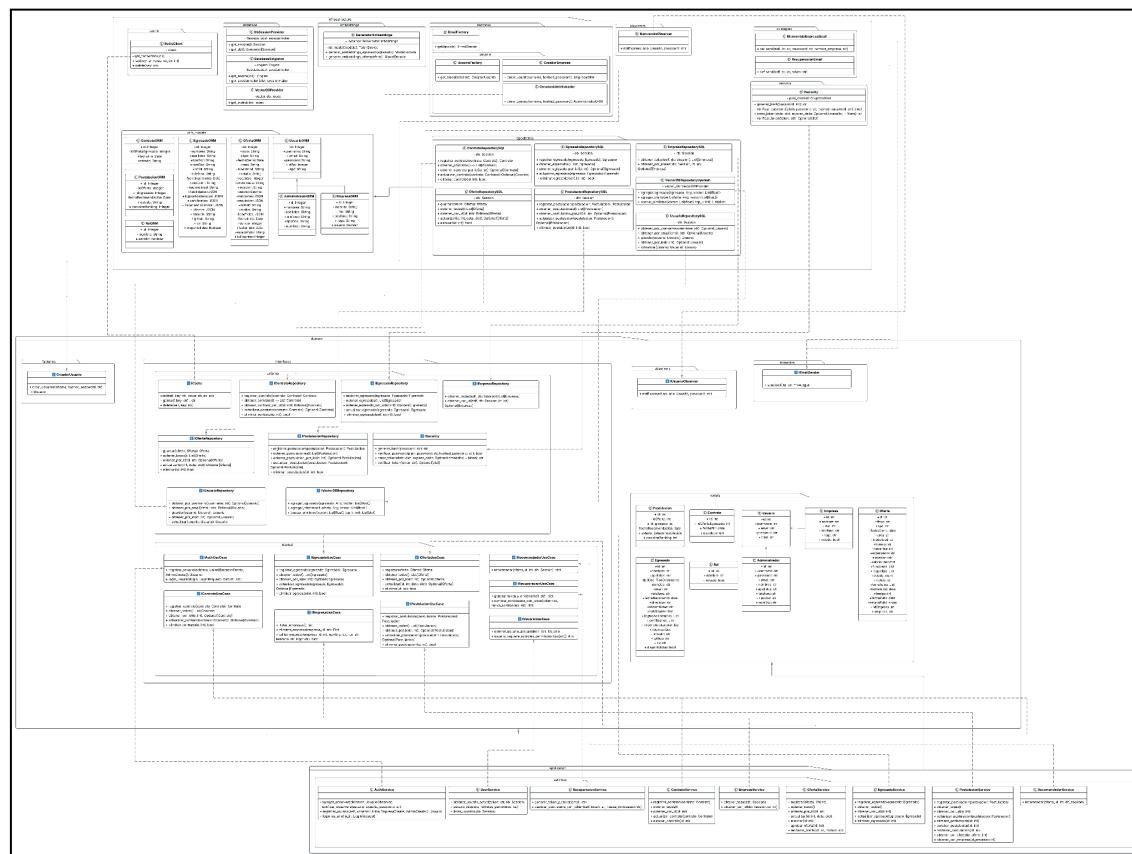
### CU - 07: "Visualizar Reportes de Empleabilidad"



**Diagrama 25.** Diagrama de actividades del CU - 07 de la plataforma LinkJob.

*Fuente:* Elaboración propia.

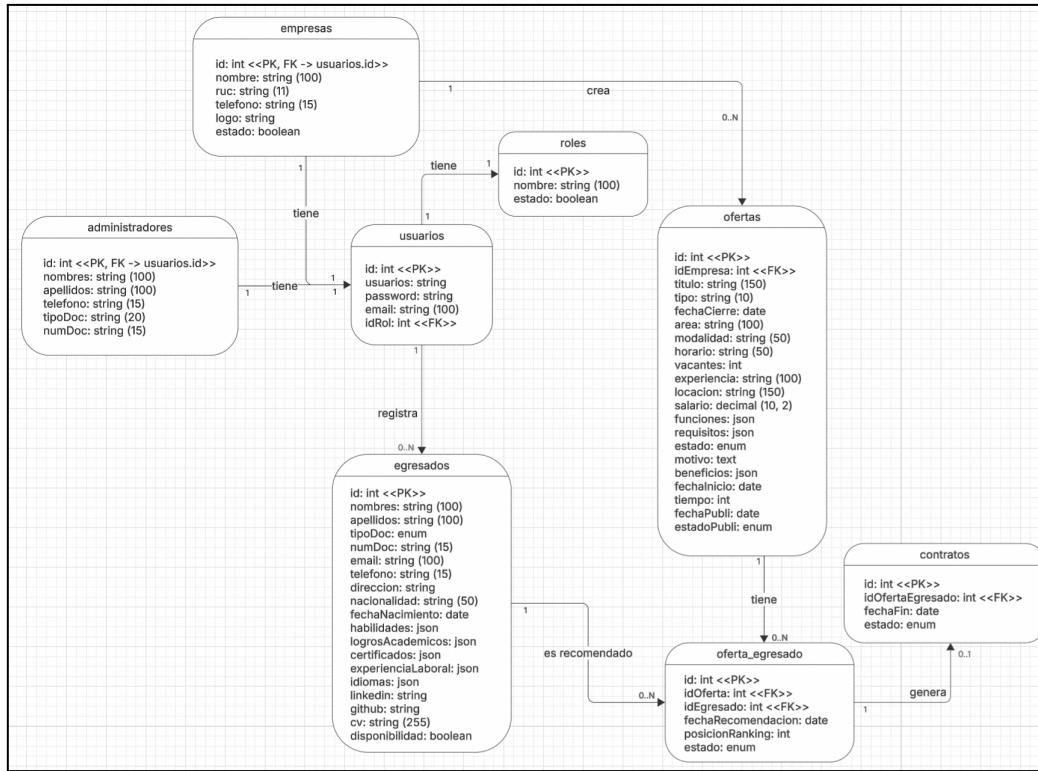
#### 3.2.5. Diagrama de Clases ([Link a Diagrama de Clases](#))



**Diagrama 26.** Diagrama de clases de la plataforma LinkJob.

*Fuente:* Elaboración propia.

### 3.2.6. Diagrama de Base de Datos ([Link Diagrama de Base de Datos](#))

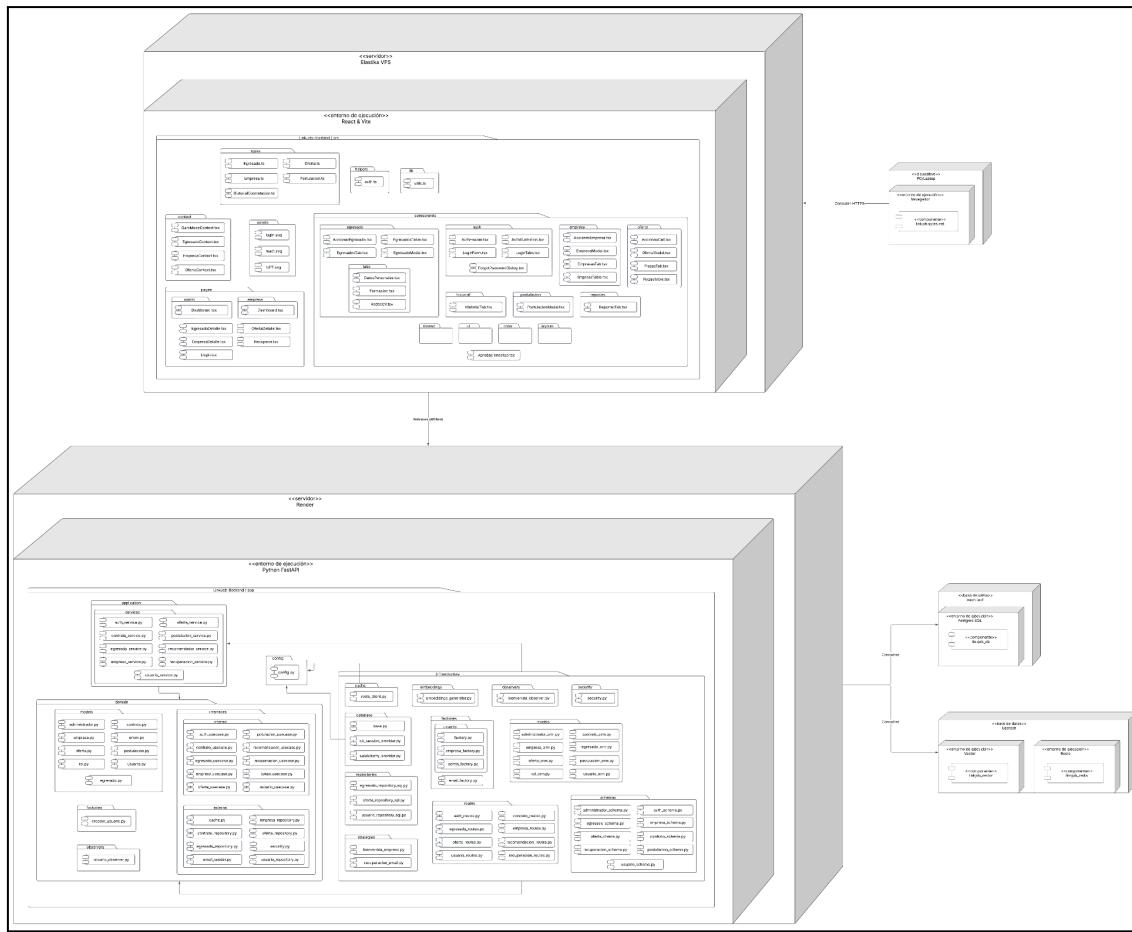


**Diagrama 27.** Diagrama de Base de Datos de la plataforma LinkJob.

**Fuente:** Elaboración propia.

### 3.3. Vista de Implementación (vista de desarrollo)

#### 3.3.1. Diagrama de arquitectura de software (paquetes) - ([Link al Diagrama](#))

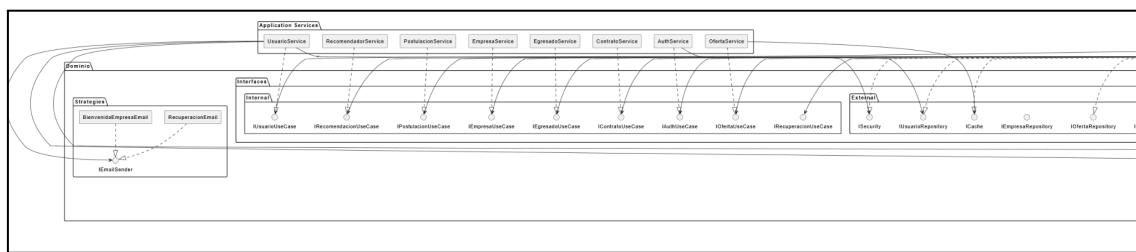


**Diagrama 28.** Diagrama de Arquitectura de Software de la plataforma LinkJob.

**Fuente:** Elaboración propia.

### 3.3.2. Diagrama de arquitectura del sistema (diagrama de componentes)

Se estructuran los principales módulos en tres capas: el Dominio, que define las entidades, interfaces de negocio y contratos de infraestructura; la capa de Servicios de Aplicación, que implementa los casos de uso; y la Infraestructura, que contiene las implementaciones concretas de los puertos externos, incluyendo repositorios, estrategias, observadores, rutas HTTP, esquemas y componentes técnicos como la base de datos y el sistema de seguridad. Las relaciones ilustran cómo los componentes interactúan de forma desacoplada mediante interfaces.



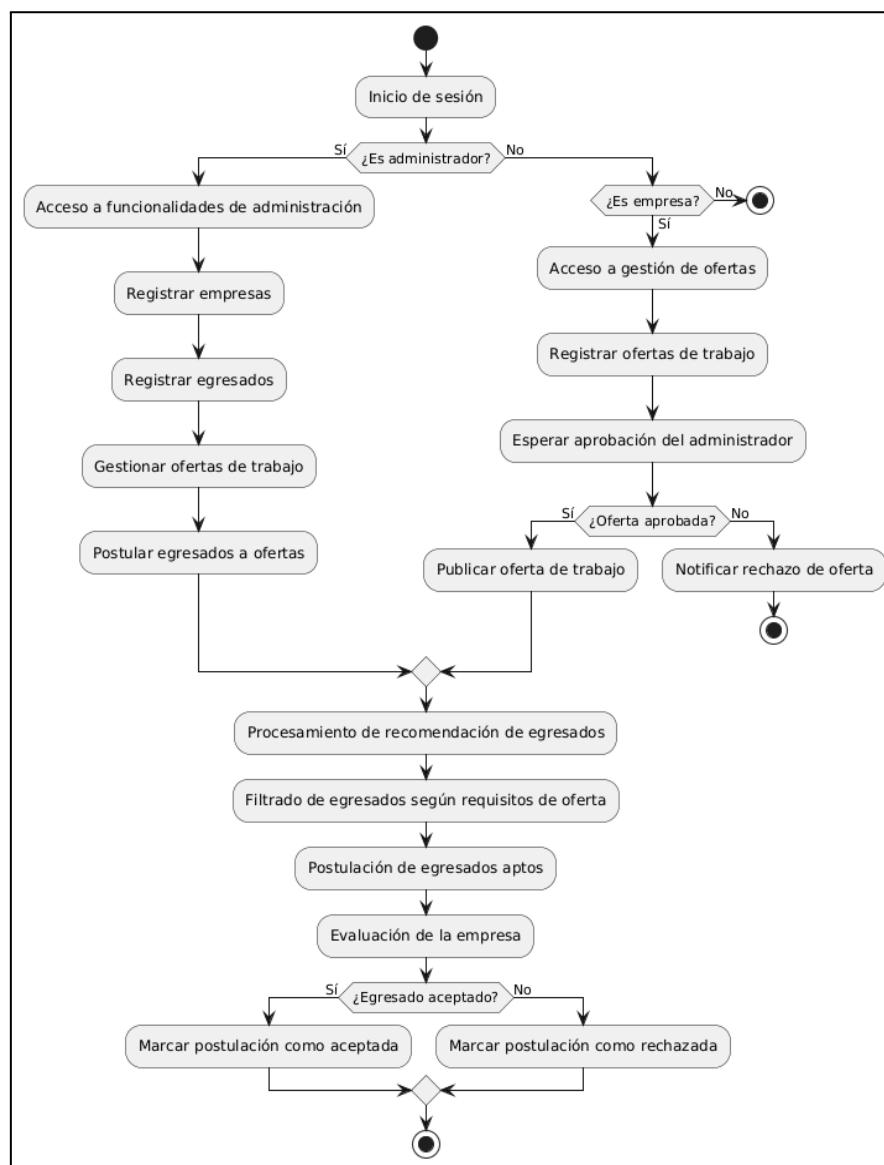
**Diagrama 29.** Diagrama de Componentes de la plataforma LinkJob.

*Fuente:* Elaboración propia.

### 3.4. Vista de Procesos

#### 3.4.1. Diagrama de procesos del sistema (diagrama de actividad)

El diagrama representa el flujo de LinkJob. Inicia con la autenticación de usuarios, diferenciando entre administradores y empresas. Los administradores gestionan empresas, egresados y plazas laborales, mientras que las empresas pueden registrar plazas, sujetas a aprobación. Una vez aprobadas, el sistema procesa la recomendación de egresados utilizando IA, evaluando habilidades y requisitos para el puesto. Finalmente, las postulaciones se marcan como aceptadas o rechazadas según la decisión de la empresa.

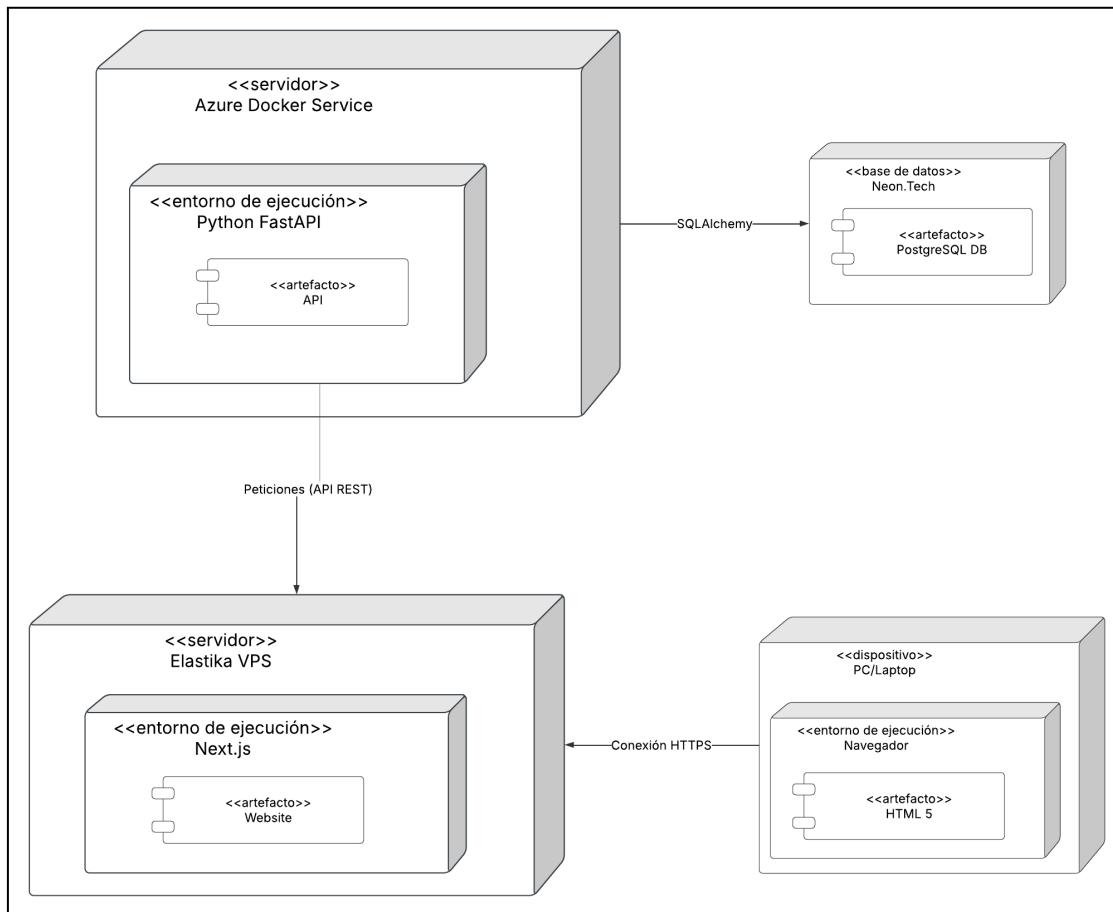


**Diagrama 30.** Diagrama de procesos del sistema de la plataforma LinkJob.

*Fuente:* Elaboración propia.

### 3.5. Vista de Despliegue

#### 3.5.1. Diagrama de despliegue



**Diagrama 31.** Diagrama de despliegue del sistema de la plataforma LinkJob.

*Fuente:* Elaboración propia.

## 4. Atributos de Calidad del Software

Los atributos de calidad (QAs) son propiedades no funcionales que definen el nivel con el que un sistema satisface expectativas relacionadas con su funcionamiento, seguridad, usabilidad, mantenimiento, rendimiento, entre otros aspectos. En el sistema LinkJob, estos atributos fueron establecidos considerando las necesidades de los stakeholders, el contexto tecnológico y las metas de escalabilidad, seguridad y experiencia de usuario de la plataforma.

### 4.1. Escenario de Usabilidad

La plataforma será diseñada para ser intuitiva tanto para administradores como para empresas. Se empleará React con Vite para construir una interfaz ágil y responsive. Se realizarán pruebas de usabilidad mediante encuestas con escala

Likert (1 a 5), donde al menos el 80% de los usuarios deberá calificar la facilidad de uso con un 4 o más. Se cuidará la coherencia visual, la accesibilidad y la navegación lógica entre módulos.

#### 4.2. Escenario de Confiabilidad

El sistema debe garantizar una disponibilidad mínima del 98% anual, controlada mediante herramientas como Uptime Kuma. Para asegurar la confiabilidad de los datos, se implementarán mecanismos de autenticación seguros, restablecimiento de contraseñas mediante Redis y cifrado de información sensible. Se realizarán pruebas de seguridad automatizadas con OWASP ZAP para garantizar al menos un 90% de cumplimiento en seguridad y asegurar la ausencia de vulnerabilidades críticas ( $CVSS \geq 7$ ) antes de cada despliegue.

#### 4.3. Escenario de Rendimiento

El sistema será capaz de escalar horizontalmente para soportar mayor carga de usuarios sin comprometer los tiempos de respuesta. Se realizarán pruebas de carga con Locust simulando un crecimiento del 50% en usuarios concurrentes, asegurando que el sistema mantenga una degradación menor al 10% en el rendimiento. El backend, desarrollado en FastAPI, permite aprovechar un alto rendimiento con bajo consumo de recursos, mientras que la base de datos PostgreSQL garantiza eficiencia en consultas complejas.

#### 4.4. Otros Escenarios

Performance: La plataforma deberá procesar eficientemente solicitudes de recomendación y búsquedas semánticas. Se usará Upstash Vector para búsquedas rápidas y escalables de similitud semántica entre egresados y ofertas. La performance se medirá en base al tiempo promedio de respuesta ante consultas intensivas, con un umbral máximo de 1 segundo por recomendación bajo condiciones normales.