



**UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO**

Facultad de Ciencias de la Computación y Diseño Digital

Carrera de Ingeniería en Software

Aplicaciones Web 'A'

# **DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN DEL PROCESO DE SELECCIÓN DE DOCENTES EN LA UTEQ**

**Proyecto de aula**

## **Autores:**

Calderón Saltos Joseph  
Herrera Barco Humberto  
Reinoso Vélez Eduardo  
Silva Triviño John

## **Docente:**

Ing. Gleiston Cicerón Guerrero Ulloa

**Quevedo - Los Ríos - Ecuador**

Noviembre 2025

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Planteamiento del problema</b>	<b>3</b>
<b>3. Objetivos</b>	<b>3</b>
3.1. Objetivo general . . . . .	3
3.2. Objetivos específicos . . . . .	3
<b>4. Justificación</b>	<b>4</b>
<b>5. Alcance del proyecto</b>	<b>4</b>
5.1. Funcionalidades del sistema . . . . .	4
5.2. Límites del sistema . . . . .	5
5.3. Restricciones tecnológicas y operativas . . . . .	5
<b>6. Marco teórico</b>	<b>6</b>
6.1. Aplicación web y arquitectura de software . . . . .	6
6.2. Interfaz de programación de aplicaciones (APIs) . . . . .	6
6.3. Sistema de gestión de bases de datos relacionales . . . . .	6
6.4. Tecnologías y frameworks seleccionados . . . . .	7
<b>7. Metodología de desarrollo e implementación</b>	<b>7</b>
7.1. Estrategia de ejecución (Sprints) . . . . .	7
7.2. Entorno tecnológico de construcción . . . . .	8
7.2.1. Desarrollo del Backend . . . . .	8
7.2.2. Construcción del Frontend . . . . .	8
7.2.3. Persistencia de Datos . . . . .	8
7.2.4. Control de Versiones y Colaboración . . . . .	9
7.3. Aseguramiento de la Calidad (QA) . . . . .	9
<b>8. Recursos necesarios</b>	<b>9</b>
8.1. Recursos de hardware . . . . .	9
8.2. Recursos de software . . . . .	9
8.3. Equipo de desarrollo . . . . .	10
<b>9. Resultados esperados</b>	<b>10</b>
<b>10. Referencias</b>	<b>11</b>
<b>11. Anexos</b>	<b>14</b>
11.1. Evidencias de las actividades realizadas por el equipo de trabajo . . . . .	14
11.2. Diseño de interfaz de usuario (mockups) . . . . .	15
11.3. Modelado de la solución . . . . .	21

## 1. Introducción

La gestión del talento humano académico es uno de los principales retos de la educación superior [1]. La selección y contratación de docentes suele desarrollarse mediante procesos extensos y poco estandarizados, lo que dificulta garantizar transparencia, equidad y trazabilidad en la evaluación [2].

En el caso de la Universidad Técnica Estatal de Quevedo (UTEQ), la contratación de docentes no titulares está regulada por la Ley Orgánica de Educación Superior (LOES) [3] y la Normativa para la selección y contratación de docentes no titulares de la institución [4]. Actualmente, el proceso se basa en la recepción de hojas de vida, certificados e informes en formato físico o digital. Esta situación complica la labor de coordinadores de carrera y decanos, quienes deben revisar expedientes y asignar puntajes manualmente, incrementando el riesgo de errores, retrasa las decisiones y dificulta la transparencia del proceso [5]. Además, la falta de estandarización limita la productividad, aumenta costos operativos y compromete la mejora continua, los cuales son elementos esenciales en instituciones públicas [6].

Diversas investigaciones han planteado soluciones relacionadas. Por ejemplo, Ccorahua et al. [7] desarrollaron un sistema web para automatizar la adjudicación docente mediante la postulación y asignación de plazas, mientras que Caiza & Guallichico [2] proponen una plataforma integral para gestionar concursos de méritos y oposición con convocatorias, evaluaciones y escalafones automatizados. Sin embargo, estas propuestas mantienen etapas manuales críticas como la validación documental y la evaluación de competencias blandas, lo que limita su alcance como soluciones completamente integrales.

En este contexto, se presenta el análisis y diseño de una propuesta orientada a mejorar la etapa de evaluación de méritos y oposición de postulantes a la cátedra en la UTEQ. La solución integra la organización digital de expedientes, la aplicación automatizada de criterios de evaluación y la generación de reportes con trazabilidad y respaldo normativo. Asimismo, incorpora un flujo estandarizado de procesamiento de datos, garantizando seguridad, transparencia y facilidad de uso, en cumplimiento de la Ley Orgánica de Protección de Datos Personales del Ecuador (LOPDP) [8].

El trabajo previo de elicitation de requisitos utilizó una metodología ágil basada en Scrum [9], adaptada para guiar el ciclo de vida de desarrollo de software. Se utilizan herramientas colaborativas para facilitar la gestión del equipo y la entrega incremental de funcionalidades. Los requisitos previamente recopilados mediante entrevistas y análisis normativo, constituyen la base para el diseño de la arquitectura y la lógica de negocio que se implementará en la solución tecnológica.

Este documento se estructura conforme a los lineamientos de una propuesta de desarrollo tecnológico: la sección 2 describe la problemática actual y la necesidad de automatización; la sección 3 establece las metas del desarrollo; y la sección 4 fundamenta la relevancia del proyecto. Posteriormente, la sección 5 delimita las funcionalidades y restricciones del sistema, mientras que las secciones 6 y ?? detallan las tecnologías (HTML5, Java) y la metodología ágil que guiarán la construcción del software. Finalmente, se presentan los recursos necesarios y los productos entregables en las secciones 8 y 9, adjuntando en los anexos los modelos de análisis y diseño preliminares.

## **2. Planteamiento del problema**

La gestión del talento humano en las instituciones de educación superior demanda procesos eficientes, transparentes y auditables. En la Universidad Técnica Estatal de Quevedo (UTEQ), el procedimiento actual para la selección y contratación de docentes no titulares se lleva a cabo mediante mecanismos manuales y canales de comunicación no integrados. La recepción de expedientes, que incluye hojas de vida, títulos académicos y certificados de experiencia, se realiza a través de medios físicos o mediante el envío de correos electrónicos dispersos, lo que impide la consolidación de un repositorio digital centralizado y seguro.

Esta situación genera una problemática multifacética que afecta tanto a la administración como a los aspirantes. En primer lugar, la falta de automatización obliga a los Coordinadores de Carrera y Decanos a realizar la validación documental y el cálculo de puntajes de manera artesanal, utilizando hojas de cálculo o formatos impresos. Este enfoque manual no solo consume una cantidad excesiva de recursos y tiempo administrativo, sino que también incrementa significativamente la probabilidad de errores humanos en la transcripción de datos y la aplicación de las fórmulas de ponderación estipuladas en la normativa institucional.

En segundo lugar, la descentralización de la información provoca una pérdida de trazabilidad durante el flujo de evaluación. Al no existir un sistema unificado, resulta complejo auditar las acciones realizadas sobre un expediente o determinar en qué etapa específica se encuentra un proceso de selección, lo que eleva el riesgo de extravío de documentación sensible.

Finalmente, se evidencia una carencia crítica de transparencia hacia los postulantes. La ausencia de una plataforma web impide que los candidatos puedan monitorear el estado de su solicitud en tiempo real o recibir notificaciones inmediatas sobre los resultados de sus evaluaciones. Esta opacidad involuntaria puede generar incertidumbre y desconfianza respecto a la equidad del concurso de méritos. Por consiguiente, es imperativo desarrollar una solución tecnológica basada en una arquitectura web que sistematice la recepción de documentos, automatice la lógica de evaluación y garantice la integridad de la información.

## **3. Objetivos**

### **3.1. Objetivo general**

Desarrollar un sistema integral para gestionar el proceso de selección docente de la UTEQ, abarcando desde la fase de convocatoria hasta la evaluación de méritos, implementando las funcionalidades necesarias para garantizar transparencia, trazabilidad, eficiencia operativa y cumplimiento del marco normativo institucional.

### **3.2. Objetivos específicos**

- Estructurar la arquitectura de software y el diseño de la base de datos relacional, integrando los modelos de análisis previamente definidos para asegurar la integridad, seguridad y persistencia de la información de los postulantes y evaluaciones.
- Implementar los módulos del sistema correspondientes a la convocatoria, postulación, verificación documental y evaluación de méritos, basados en los requisitos funcionales y no funcionales previamente definidos y validados con los actores del proceso.
- Integrar componentes de inteligencia artificial para apoyar la evaluación de méritos, mediante modelos capaces de identificar patrones, verificar coherencias, sugerir puntajes preliminares y facilitar la revisión de documentos, asegurando transparencia y evitando sesgos mediante supervisión humana constante.

## 4. Justificación

La ejecución de este proyecto es pertinente y necesaria debido a su impacto directo en la modernización de los procesos administrativos de la UTEQ, alineándose con las normativas nacionales y las tendencias tecnológicas actuales.

La implementación de una solución web representa un avance significativo respecto al manejo actual de archivos locales. El uso de tecnologías de desarrollo web modernas (HTML5, Java) permitirá crear un sistema escalable y seguro, eliminando las barreras geográficas y temporales que actualmente limitan a los evaluadores. Además, la centralización de la información en una base de datos relacional asegurará la integridad y persistencia de los datos, resolviendo los problemas de duplicidad y pérdida de documentos detectados en diagnósticos previos [7].

Para la institución, la automatización del cálculo de la matriz de méritos reducirá drásticamente la carga laboral operativa. Al liberar a los Decanos y Coordinadores de tareas repetitivas de verificación manual, se optimiza el tiempo de respuesta del proceso de contratación, un factor crítico para la eficiencia de las instituciones públicas [6]. Asimismo, el sistema proporcionará reportes estandarizados que facilitarán la toma de decisiones basada en datos reales, mejorando la gestión del talento humano [5].

Desde una perspectiva social, el proyecto fomenta la igualdad de oportunidades y la transparencia. Al proporcionar a los postulantes una herramienta clara para gestionar su participación, se fortalece la imagen institucional de la UTEQ como una entidad justa. Legalmente, el desarrollo del software es indispensable para dar cumplimiento estricto a la Ley Orgánica de Educación Superior (LOES) [3] en cuanto a concursos de merecimientos, y a la Ley Orgánica de Protección de Datos Personales (LOPD) [8], garantizando el tratamiento ético y seguro de la información confidencial de los profesionales.

## 5. Alcance del proyecto

En esta sección se define la extensión operativa y técnica de la solución tecnológica propuesta. El proyecto contempla el ciclo de desarrollo de software completo, desde el diseño hasta la implementación, detallando a continuación las funcionalidades incluidas, las exclusiones del sistema y las restricciones bajo las cuales operará la aplicación para dar cumplimiento a los objetivos planteados.

### 5.1. Funcionalidades del sistema

La aplicación web integrará los siguientes módulos operativos:

- **Módulo de gestión de convocatorias:** Permitirá a Talento Humano crear y publicar convocatorias, definir requisitos, subir bases del proceso, gestionar cronogramas y habilitar la inscripción de postulantes durante el periodo correspondiente.
- **Módulo de administración:** Permitirá al administrador del sistema registrar la terna oficial de postulantes remitida por Talento Humano y configurar los parámetros del periodo de evaluación. Permitirá gestionar cuentas de usuario, roles, permisos, restablecimiento de credenciales y parámetros técnicos generales de operación.
- **Módulo del postulante:** Proveerá una interfaz segura para que los aspirantes se registren, se inscriban en las convocatorias vigentes, carguen sus documentos habilitantes en formato PDF y consulten en tiempo real el estado de su evaluación y puntajes preliminares.
- **Módulo de evaluación y méritos asistido por IA:** Facilitará a Coordinadores y Decanos la revisión de documentos, su validación o rechazo justificado y la

asignación de puntajes según la matriz vigente. Integra herramientas de inteligencia artificial que permitirán: detectar inconsistencias o documentos faltantes, clasificar automáticamente tipos de documentos, sugerir puntajes preliminares basados en reglas y generar resúmenes o extractos de información relevante. Estas sugerencias serán supervisadas y confirmadas por los evaluadores humanos para garantizar imparcialidad y transparencia.

- **Módulo de entrevistas y oposición:** Permitirá el agendamiento de entrevistas y clases demostrativas, así como el registro de las calificaciones cualitativas obtenidas en estas fases.
- **Módulo de reportes y notificaciones:** Generará automáticamente actas de resultados individuales y consolidados para su envío al Vicerrectorado Académico, además de notificar vía correo electrónico a los postulantes sobre los avances del proceso.

## 5.2. Límites del sistema

El alcance del desarrollo se circunscribe a la gestión digital de la convocatoria, la postulación y la evaluación de méritos y aspectos de oposición. Por tanto, el sistema no incluirá:

- La contratación legal posterior del docente (firma de contratos, trámites administrativos).
- La automatización de exámenes teóricos o pruebas de conocimientos escritos.
- Procesos administrativos externos que dependan de otras áreas institucionales.

## 5.3. Restricciones tecnológicas y operativas

El desarrollo deberá adherirse a las siguientes condiciones:

- La aplicación deberá ser compatible con los navegadores web modernos (Chrome, Firefox, Edge) sin requerir instalación de complementos adicionales.
- El sistema deberá implementar un control de acceso basado en roles (RBAC) para asegurar que cada actor acceda únicamente a las funciones autorizadas.
- Se deberá garantizar la persistencia de logs de auditoría para todas las transacciones críticas (validación de documentos y asignación de notas) para asegurar la trazabilidad exigida por la normativa.
- El despliegue se realizará considerando la infraestructura tecnológica disponible en la UTEQ o servidores en la nube compatibles con los protocolos de seguridad institucional.

## 6. Marco teórico

El desarrollo de soluciones de software modernas requiere una base teórica sólida que respalde las decisiones arquitectónicas y la selección de herramientas tecnológicas. En esta sección se definen los conceptos clave que sustentan la construcción del sistema de gestión de la evaluación del mérito, incluyendo la arquitectura de software, las interfaces de programación y las tecnologías de persistencia de datos.

### 6.1. Aplicación web y arquitectura de software

Una aplicación web se define como un sistema de software distribuido al que se puede acceder a través de una red, normalmente Internet, mediante un navegador web. A diferencia de las aplicaciones de escritorio monolíticas, las aplicaciones web modernas están estructuradas en componentes que se pueden implementar de forma independiente y que interactúan entre sí para garantizar la escalabilidad y la facilidad de mantenimiento [10], [11]. La arquitectura de una aplicación web determina su estructura global y la organización de sus unidades de implementación. En los últimos años, el patrón Modelo-Vista-Controlador (MVC) y las arquitecturas por capas han seguido siendo los enfoques predominantes para garantizar la modularidad del código y separar la lógica de negocio de las cuestiones relacionadas con la interfaz de usuario [10], [12].

Para este proyecto, el sistema adopta una arquitectura que separa el frontend del backend, lo que facilita la evolución y la integración independientes. Esta separación es fundamental en entornos de desarrollo colaborativo, ya que permite organizar el código fuente en unidades modulares, lo que a su vez permite la integración de nuevas funcionalidades sin comprometer la estabilidad del sistema existente. [11], [12].

### 6.2. Interfaz de programación de aplicaciones (APIs)

La comunicación entre el cliente (frontend) y el servidor (backend) se establece mediante API RESTful. Una interfaz de programación de aplicaciones (API) sirve como un conjunto de definiciones y protocolos que permiten la interacción entre componentes de software heterogéneos. El estilo arquitectónico Representational State Transfer (REST) se ha consolidado como el estándar para el diseño de API web debido a su naturaleza sin estado y al uso eficiente del protocolo HTTP [13], [14].

Las API RESTful descomponen la funcionalidad en recursos que pueden manipularse mediante operaciones estándar (GET, POST, PUT, DELETE), lo que permite una interoperabilidad eficiente y sin estado entre el servidor y varios clientes [13], [15]. Estudios recientes destacan que el uso de arquitecturas REST no solo facilita la integración de servicios externos, sino que también garantiza que la lógica empresarial subyacente pueda ser utilizada de forma segura por diferentes plataformas, como interfaces web o móviles, al tiempo que se minimizan las vulnerabilidades [14], [15].

### 6.3. Sistema de gestión de bases de datos relacionales

La persistencia de los datos es fundamental para garantizar la integridad de la información de los solicitantes y los registros de evaluación. Se utiliza un sistema de gestión de bases de datos relacionales (RDBMS) para gestionar los datos estructurados. A pesar de la aparición de las tecnologías NoSQL para datos no estructurados, las bases de datos relacionales basadas en el lenguaje de consulta estructurado (SQL) siguen siendo la opción más sólida para los escenarios que requieren una integridad referencial y una coherencia transaccional estrictas (propiedades ACID) [16], [17].

Las revisiones sistemáticas indican que, en el caso de los sistemas administrativos y de evaluación en los que la estructura de los datos es predecible —como las relaciones entre solicitantes, métricas y puntuaciones—, los sistemas relacionales como PostgreSQL

o MySQL ofrecen un rendimiento superior en el procesamiento de consultas complejas y la evolución de esquemas en comparación con las alternativas no relacionales [16], [18]. Además, la capacidad de gestionar cambios dinámicos en los datos manteniendo al mismo tiempo estrictas restricciones de esquema es un factor crítico para la fiabilidad a largo plazo de los sistemas de gestión académica [17], [18].

#### 6.4. Tecnologías y frameworks seleccionados

La selección del stack tecnológico se fundamentó en un análisis comparativo reciente de rendimiento, seguridad y mantenibilidad, priorizando herramientas que garanticen la integridad de los datos en el proceso de evaluación de méritos.

- **Backend (Java & Spring Boot):** Para la lógica del servidor se seleccionó el lenguaje Java implementando el framework Spring Boot. Estudios comparativos recientes demuestran que Spring Boot ofrece una gestión de recursos más estable y una mejor escalabilidad en arquitecturas RESTful empresariales en comparación con alternativas como Laravel o Django, lo cual es crítico para procesar la matriz de méritos sin errores de concurrencia [19].
- **Frontend (Estándares Web Nativos):** La interfaz de usuario se construyó utilizando HTML5 y CSS3 nativos, apoyados por JavaScript estándar. El uso de estas tecnologías estándar optimiza el renderizado gráfico y la respuesta de la interfaz en prototipos de aplicaciones web, garantizando una experiencia de usuario fluida y una alta compatibilidad con navegadores modernos sin la sobrecarga de librerías adicionales [20].
- **Base de Datos (PostgreSQL):** La persistencia de datos se gestiona mediante el SGBD relacional PostgreSQL. La elección sobre MySQL se justifica en benchmarks recientes que evidencian que PostgreSQL es hasta 9 veces más eficiente en consultas complejas (como las requeridas para reportes y filtros de postulantes) y mantiene una mayor estabilidad en operaciones simultáneas de inserción y lectura, asegurando la trazabilidad del proceso [21].

### 7. Metodología de desarrollo e implementación

Para la construcción y despliegue de la solución tecnológica, se ejecutó una estrategia de desarrollo basada en la metodología ágil Scrum [9]. Este enfoque permitió abordar la codificación del sistema de manera iterativa e incremental, facilitando la entrega de módulos funcionales operativos al finalizar cada Sprint y asegurando la adaptación inmediata a los criterios de aceptación técnicos.

El ciclo de vida del desarrollo de software (SDLC) se llevó a cabo en cuatro fases de ejecución intensiva, cubriendo desde la configuración del entorno de producción hasta el despliegue final de la aplicación web.

#### 7.1. Estrategia de ejecución (Sprints)

La fase de construcción del software se organizó en cuatro sprints, durante los cuales el equipo de desarrollo transformó los requisitos en código funcional. A continuación, se detallan las actividades técnicas realizadas y los módulos implementados en cada iteración [4], [9]:

Iteración	Módulo Desarrollado	Implementación Técnica Realizada
<b>Sprint 1</b>	Arquitectura y Seguridad	Se configuró el entorno de desarrollo en IntelliJ IDEA y se implementó la base de datos relacional. Se codificó el módulo de seguridad, incluyendo el sistema de autenticación, login y el control de acceso basado en roles (RBAC) [4], [8].
<b>Sprint 2</b>	Gestión Documental	Se desarrolló la lógica del backend (Java) y las interfaces frontend (HTML5) para permitir la carga asíncrona, validación de formato y almacenamiento seguro de los documentos PDF de los postulantes [4].
<b>Sprint 3</b>	Núcleo de Evaluación	Se programó la lógica de negocio para la aplicación automática de la matriz de méritos. Además, se implementó el módulo de gestión de entrevistas, permitiendo la asignación de puntajes por oposición [3].
<b>Sprint 4</b>	Reportes y Despliegue	Se construyó el motor de generación de reportes consolidados y el sistema de notificaciones automáticas por correo electrónico. Finalmente, se realizó el despliegue de la aplicación en el servidor de producción [5], [22].

Cuadro 1: Ejecución de Sprints de Desarrollo

## 7.2. Entorno tecnológico de construcción

La implementación del sistema se realizó utilizando un stack tecnológico moderno, seleccionado por su robustez y escalabilidad, tal como se definió en la arquitectura del proyecto [23].

### 7.2.1. Desarrollo del Backend

La lógica del servidor fue codificada utilizando el lenguaje Java, aprovechando su tipado estático y capacidad de gestión de memoria para procesar la matriz de méritos de forma eficiente [24]. El desarrollo se realizó dentro del entorno integrado IntelliJ IDEA, que facilitó la depuración y compilación del código fuente.

### 7.2.2. Construcción del Frontend

Para la interfaz de usuario, se implementó una arquitectura basada en componentes utilizando HTML5, CSS3 y JavaScript. Esto aseguró que la aplicación fuera responsive y compatible con los navegadores modernos sin requerir complementos adicionales, cumpliendo con los requisitos de accesibilidad [25].

### 7.2.3. Persistencia de Datos

Se implementó y normalizó una base de datos relacional PostgreSQL para garantizar la integridad referencial de la información de los postulantes y sus evaluaciones [23]. Se diseñaron procedimientos almacenados y disparadores (triggers) para mantener la consistencia de los datos históricos.

#### 7.2.4. Control de Versiones y Colaboración

Durante todo el proceso de codificación, se utilizó Git para el control de versiones, gestionando el repositorio centralizado en GitHub [22]. Esto permitió la integración continua del código desarrollado por los distintos miembros del equipo y mantuvo un historial auditable de todos los cambios en el código fuente.

### 7.3. Aseguramiento de la Calidad (QA)

De forma paralela al desarrollo, se ejecutó una estrategia de pruebas integral para validar la robustez del software entregado [24]:

- **Pruebas Unitarias:** Se validó la lógica de los métodos individuales, especialmente las fórmulas de cálculo de méritos, asegurando que el motor de evaluación funcionara sin errores matemáticos.
- **Pruebas de Integración:** Se verificó la correcta comunicación entre el frontend, la API del backend y la base de datos, garantizando que los datos fluyeran correctamente entre las capas del sistema [23].
- **Pruebas de Usabilidad:** Se realizaron sesiones de validación de interfaz para asegurar que los flujos de postulación y evaluación fueran intuitivos para los usuarios finales [5], [25].

El resultado final de esta metodología fue una aplicación web totalmente operativa, desplegada y validada, capaz de gestionar el proceso de selección docente conforme a la normativa institucional vigente [4].

## 8. Recursos necesarios

Para la materialización de la solución tecnológica, se utilizaron recursos técnicos y humanos específicos que permitieron transformar las especificaciones de requisitos en un producto de software funcional y operativo [4].

### 8.1. Recursos de hardware

La infraestructura física utilizada para garantizar el flujo de trabajo del equipo y la operatividad del sistema incluyó:

- **Estaciones de trabajo:** Equipos de cómputo con las características técnicas necesarias (procesador y memoria RAM adecuados) que soportaron la virtualización y la ejecución simultánea de los servicios de backend y frontend sin degradación del rendimiento.
- **Infraestructura de servidor:** Un entorno de servidor configurado para el alojamiento de la base de datos y el despliegue de la aplicación web, asegurando la disponibilidad requerida durante las pruebas y la puesta en producción [24].

### 8.2. Recursos de software

Se empleó un entorno tecnológico basado en estándares de la industria, alineado estrictamente con la arquitectura de software definida en la metodología:

- **Entorno de Desarrollo e Implementación:**
  - *IDE:* IntelliJ IDEA para la codificación, depuración y compilación de la lógica del servidor.

- **Lenguajes y Tecnologías:** Java para el desarrollo del backend; HTML5, CSS3 y JavaScript para la construcción de la interfaz de usuario.
- **Gestión de Datos:** Sistema de Gestión de Bases de Datos (SGBD) relacional PostgreSQL para garantizar la persistencia e integridad de la información de los postulantes [23].
- **Gestión de la Configuración:** Uso de Git y la plataforma GitHub para el control de versiones del código fuente, lo que facilitó la integración continua y el trabajo colaborativo [22].

### 8.3. Equipo de desarrollo

El proyecto fue ejecutado por un equipo técnico multidisciplinario organizado bajo el marco de trabajo Scrum [9]. Este equipo demostró las competencias necesarias en desarrollo web *Full-Stack* cubriendo los roles técnicos definidos:

- **Backend developer:** Fue el responsable de la lógica del servidor, la implementación de la matriz de méritos y la seguridad de los datos.
- **Frontend developer:** Se encargó de la construcción de las interfaces en HTML5, asegurando la usabilidad y el diseño responsivo.
- **Administrador de base datos (DBA):** Responsable del diseño, implementación y optimización del esquema de base de datos relacional.
- **QA tester:** Encargado de la ejecución del plan de pruebas integrales para garantizar la calidad del software entregado [26].

## 9. Resultados esperados

La ejecución de este proyecto de desarrollo culminó con la entrega de un producto tecnológico que resolvió la problemática de gestión manual detectada inicialmente. Los resultados tangibles obtenidos al finalizar el periodo de desarrollo fueron:

1. **Sistema Web desplegado:** Se entregó una aplicación accesible vía navegador que integró exitosamente los módulos de administración, postulación y evaluación. El sistema quedó totalmente operativo y configurado en el entorno tecnológico de la UTEQ, cumpliendo con los requerimientos de accesibilidad [24].
2. **Automatización del proceso:** Se implementó y probó un motor lógico capaz de realizar el cálculo de la matriz de méritos sin intervención manual. Esto eliminó el margen de error humano en la asignación de puntajes, resolviendo los problemas de inconsistencia reportados en el análisis previo [5].
3. **Repositorio de código fuente:** Se realizó la entrega del proyecto completo desarrollado en IntelliJ IDEA, debidamente documentado y versionado. El repositorio quedó listo para su mantenimiento y escalabilidad futura por parte del área de TI de la institución [22].
4. **Documentación técnica y de usuario:** Se generaron y entregaron los manuales necesarios para el uso y soporte del sistema:
  - Manual de despliegue e instalación del sistema en servidores de producción.
  - Guías de usuario interactivas diseñadas específicamente para los roles de Administradores, Evaluadores y Postulantes.

5. **Base de datos consolidada:** Se entregó un esquema de base de datos funcional que centraliza históricamente la información de los procesos de selección. Esta estructura garantiza la integridad de los datos y facilita la ejecución de futuras auditorías conforme a la normativa legal vigente [8], [23].

## 10. Referencias

### Referencias

- [1] J. E. Pinargote Párraga y M. E. Pico Macías, “Modelo de Gestión de Talento Humano como factor del desarrollo en centros de educación superior: revisión bibliográfica,” *RECIMUNDO*, vol. 7, n.º 2, págs. 117-131, jun. de 2023, ISSN: 2588-073X. DOI: 10.26820/recimundo/7.(2).jun.2023.117-131. dirección: <https://recimundo.com/index.php/es/article/view/2032>.
- [2] E. S. Caiza Chulde y G. B. Guallichico Chingo, “Sistema web para gestionar la necesidad de contratación de docentes en la Universidad Iberoamericana del Ecuador,” Trabajo de titulación, Universidad Iberoamericana del Ecuador, 2023. dirección: <http://repositorio.unibe.edu.ec/bitstream/handle/123456789/552/CAIZA%20CHULDE%20ERICK%20STEVEN%20y%20GONZALO%20BLADIMIR%20GUALLICHICO%20CHINGO.pdf?sequence=1&isAllowed=y>.
- [3] A. N. del Ecuador, *Ley Orgánica de Educación Superior*, es, Reformada por Ley 0 publicada en Registro Oficial Suplemento 297 de 2 de agosto de 2018 y otras reformas posteriores., 12 de oct. de 2010. dirección: <https://www.ces.gob.ec/documentos/Normativa/LOES.pdf>.
- [4] Universidad Técnica Estatal de Quevedo. “Normativa para la Selección y Contratación de Docentes No Titulares en la UTEQ.” dirección: [https://www.uteq.edu.ec/assets/docs/reglm-norm/n\\_contrato\\_doc.pdf](https://www.uteq.edu.ec/assets/docs/reglm-norm/n_contrato_doc.pdf).
- [5] Coordinadora de Carrera de Ingeniería de Software y Decano de la Facultad de Ciencias de la Computación y Diseño Digital, *Entrevista sobre el proceso actual de selección de docentes en la UTEQ*, Quevedo, 27 de junio, 2025. dirección: <https://drive.google.com/drive/folders/1Yk6xr2z4rvlni6R0D0nVzK58f01mC0aj>.
- [6] C. Ramos Estrada, J. A. López Lemus y R. Navarrete Reynoso, “Sistema integral para la gestión de los procesos institucionales: caso de la Universidad de Guanajuato, México,” *Transdigital*, vol. 6, n.º 11, e431, 2025. DOI: <https://doi.org/10.56162/transdigital431>.
- [7] A. J. Ccorahua-Mamani, Y. Y. Vargas-Ocola, N. A. Gallegos-Ramos y E. G. Estrada-Araoz, “Implementación de un sistema web para optimizar el proceso de adjudicación docente en la Unidad de Gestión Educativa de Tambopata,” *Revista Amazonía Digital*, vol. 3, n.º 2, e287, 2024. DOI: 10.55873/rad.v3i2.287. dirección: <https://revistas.unamad.edu.pe/index.php/rad/article/view/287>.
- [8] Asamblea Nacional del Ecuador, *Ley Orgánica de Protección de Datos Personales*, Registro Oficial Suplemento 459, Normativa vigente a partir del 26 de mayo de 2021, mayo de 2021. dirección: <https://www регистрация официальный.gob.ec/index.php/registro-oficial-web>.
- [9] K. Schwaber y J. Sutherland. “La Guía de Scrum,” Scrum.org. dirección: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>.

- [10] G. C. Amorim, M. T. Valente y R. Terra, “Micro-frontend architecture in software development: A systematic mapping study,” en *Proceedings of the 27th International Conference on Enterprise Information Systems (ICEIS)*, vol. 1, 2025, págs. 120-132. DOI: 10.5220/0013195800003929.
- [11] Y. Mo y B. Li, “ArchiWeb: A web platform for AI-driven early-stage architectural design,” *Frontiers of Architectural Research*, vol. 14, n.º 6, págs. 1551-1566, 2025, ISSN: 2095-2635. DOI: <https://doi.org/10.1016/j.foar.2025.06.002>. dirección: <https://www.sciencedirect.com/science/article/pii/S2095263525000809>.
- [12] O. Söderman, “Comparison of web application architecture styles,” Tesis de mtría., Aalto University, 2025. dirección: <https://aalto.doc.aalto.fi/server/api/core/bitstreams/4e1121c4-6409-4032-91aa-381b7b17096b/content>.
- [13] A. Ehsan, M. A. M. Abuhalqa, C. Catal y D. Mishra, “RESTful API testing methodologies: Rationale, challenges, and solution directions,” *Applied Sciences*, vol. 12, n.º 9, pág. 4369, 2022. dirección: <https://doi.org/10.3390/app12094369>.
- [14] F. Tanveer, F. Iradat, W. Iqbal y A. Ahmad, “Towards Secure APIs: A Survey on RESTful API Vulnerability Detection,” *Computers, Materials and Continua*, vol. 84, n.º 3, págs. 4223-4257, 2025, ISSN: 1546-2218. DOI: <https://doi.org/10.32604/cmc.2025.067536>. dirección: <https://www.sciencedirect.com/science/article/pii/S1546221825007040>.
- [15] M. Szewczyk, “Performance comparison of development frameworks in the context of REST API creation,” *Journal of Computer Sciences Institute*, vol. 34, págs. 12-18, 2025. DOI: <https://doi.org/10.35784/jcsi.7041>.
- [16] T. Taipalus, “Database management system performance comparisons: A systematic literature review,” *Journal of Systems and Software*, vol. 208, pág. 111872, 2024, ISSN: 0164-1212. DOI: <https://doi.org/10.1016/j.jss.2023.111872>. dirección: <https://www.sciencedirect.com/science/article/pii/S0164121223002674>.
- [17] Z. Brahmia, F. Grandi y B. Oliboni, “A Literature Review on Schema Evolution in Databases,” *COMPUTING OPEN*, vol. 02, n.º 2430001, págs. 1-54, 2024. DOI: [10.1142/s2972370124300012](https://doi.org/10.1142/s2972370124300012).
- [18] W. Khan, T. Kumar, C. Zhang, K. Raj, A. M. Roy y B. Luo, “SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review,” *Big Data and Cognitive Computing*, vol. 7, mayo de 2023. DOI: [10.3390/bdcc7020097](https://doi.org/10.3390/bdcc7020097).
- [19] M. Szewczyk y M. Skublewska-Paszkowska, “Performance comparison of development frameworks in selected environments in REST API architecture,” *Journal of Computer Sciences Institute*, vol. 34, págs. 121-128, 2025. DOI: [10.35784/jcsi.7041](https://doi.org/10.35784/jcsi.7041).
- [20] M. Almalki y N. R. Aljohani, “The Role of HTML5 and CSS3 in Creating Optimized Graphic Prototype Websites and Application Interfaces,” *NeuroQuantology*, vol. 20, n.º 12, págs. 4522-4536, 2022. dirección: [https://www.researchgate.net/publication/387238322\\_The\\_Role\\_of\\_HTML5\\_and\\_CSS3\\_in\\_Creating\\_Optimized\\_Graphic\\_Prototype\\_Wbsites\\_and\\_Application\\_Interfaces](https://www.researchgate.net/publication/387238322_The_Role_of_HTML5_and_CSS3_in_Creating_Optimized_Graphic_Prototype_Wbsites_and_Application_Interfaces).
- [21] M. M. Al-Shammary y S. A. Al-Hajri, “A Performance Benchmark for the PostgreSQL and MySQL Databases,” *Future Internet*, vol. 16, n.º 10, pág. 382, 2024. DOI: [10.3390/fi16100382](https://doi.org/10.3390/fi16100382).
- [22] B. Beyer, C. Jones, J. Petoff y N. R. Murphy, *Site Reliability Engineering: How Google Runs Production Systems*. O'Reilly Media, 2016. dirección: [http://repo.darmajaya.ac.id/4636/1/Site%20Reliability%20Engineering\\_%20How%20Google%20Runs%20Production%20Systems%20%28%20PDFDrive%20%29.pdf](http://repo.darmajaya.ac.id/4636/1/Site%20Reliability%20Engineering_%20How%20Google%20Runs%20Production%20Systems%20%28%20PDFDrive%20%29.pdf).

- [23] M. Kleppmann, *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*, 1st. Sebastopol, CA: O'Reilly Media, Inc., 2017, Accedido en: 1 de diciembre de 2025, ISBN: 978-1-449-37332-0. dirección: [https://unidel.edu.ng/focelibrary/books/Designing%20Data-Intensive%20Applications%20The%20Big%20Ideas%20Behind%20Reliable,%20Scalable,%20and%20Maintainable%20Systems%20by%20Martin%20Kleppmann%20\(z-lib.org\).pdf](https://unidel.edu.ng/focelibrary/books/Designing%20Data-Intensive%20Applications%20The%20Big%20Ideas%20Behind%20Reliable,%20Scalable,%20and%20Maintainable%20Systems%20by%20Martin%20Kleppmann%20(z-lib.org).pdf).
- [24] K. Hernández Rueda, M. P. Martínez Vargas y M. D. Casillas, “Evaluación del rendimiento de una aplicación web,” *South Florida Journal of Development*, vol. 3, n.º 1, págs. 445-457, 2022. DOI: <https://doi.org/10.46932/sfjdv3n1-034>.
- [25] Figma, Inc., *Figma*, Utilizado para el diseño de prototipos de interfaz, 2023. dirección: <https://www.figma.com>.
- [26] G. Parra-Quero, O. Palma-Urdaneta, M. E. Torres-Samuel y F. Durán-Garrido, “Caracterización de buenas prácticas en la elicitation de requisitos de software referidas en el estándar ISO/IEC/IEEE 29148,” *Publicaciones en Ciencias y Tecnología*, vol. 14, n.º 2, págs. 91-99, jul. de 2020, ISSN: 1856-8890. DOI: <http://doi.org/10.13140/RG.2.2.35706.82889>. dirección: <https://revistas.uclave.org/index.php/pcyt>.
- [27] K. Abad, J. P. Carvallo y J. P. Carvallo, “Descubriendo patrones de modelos de contexto basados en i\*,” *Maskana*, vol. 6, n.º Supl. Págs. 87-98, dic. de 2015. dirección: <https://publicaciones.ucuenca.edu.ec/ojs/index.php/maskana/article/view/701>.

## 11. Anexos

### 11.1. Evidencias de las actividades realizadas por el equipo de trabajo

Los audios correspondientes a las entrevistas con las autoridades académicas y la validación de los requisitos se encuentran disponibles en el siguiente enlace:

- Carpeta de audios de entrevistas

La documentación requerida para la etapa de selección docente está disponible en el siguiente enlace:

- Carpeta de documentos de selección docente

#### Diagramas SD/SR:

Estos diagramas elaborados en StarUML bajo la notación i\*\*, complementan el análisis de requisitos [27]. El Diagrama de Dependencia Estratégica (SD) muestra las relaciones entre los actores identificados y las dependencias que existen entre ellos para alcanzar sus metas. Por su parte, el Diagrama de Justificación Estratégica (SR) detalla cómo dichas metas, tareas y recursos se vinculan y contribuyen al cumplimiento de los objetivos del sistema.

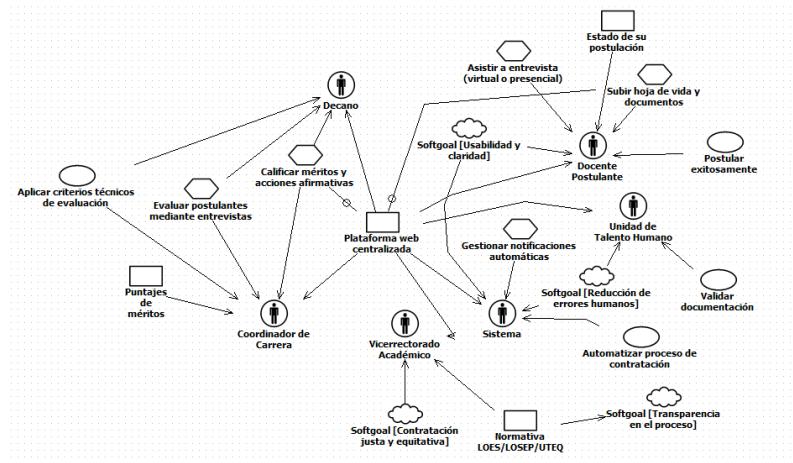


Figura 1: Diagrama SD (Strategic Dependency)

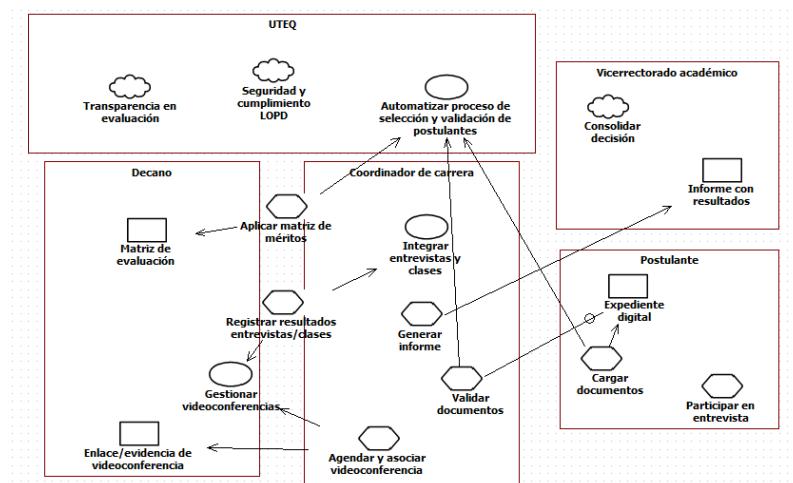


Figura 2: Diagrama SR (Strategic Rationale)

## 11.2. Diseño de interfaz de usuario (mockups)

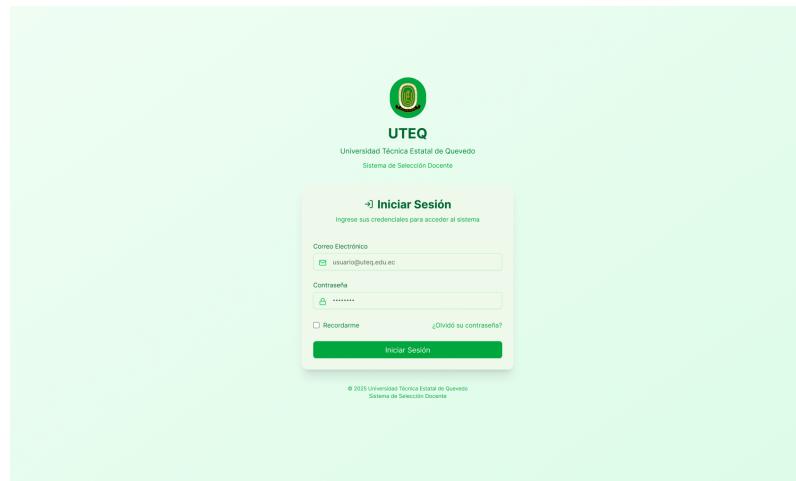


Figura 3: Interfaz de Inicio de Sesión

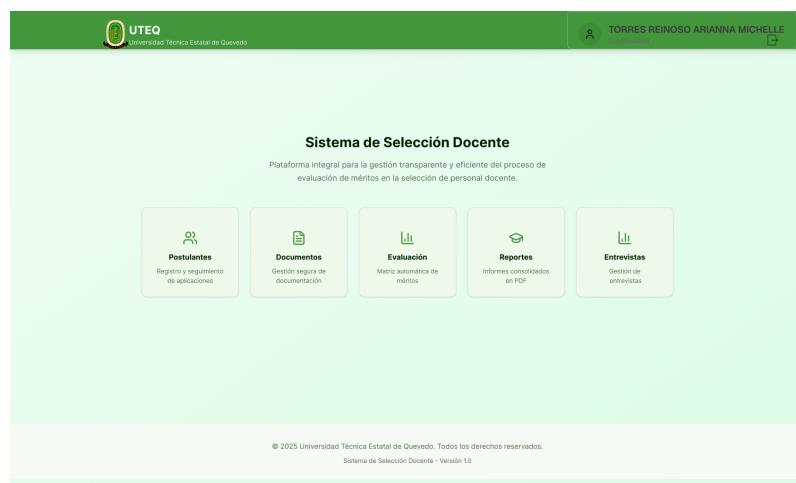


Figura 4: Interfaz de Menú Principal

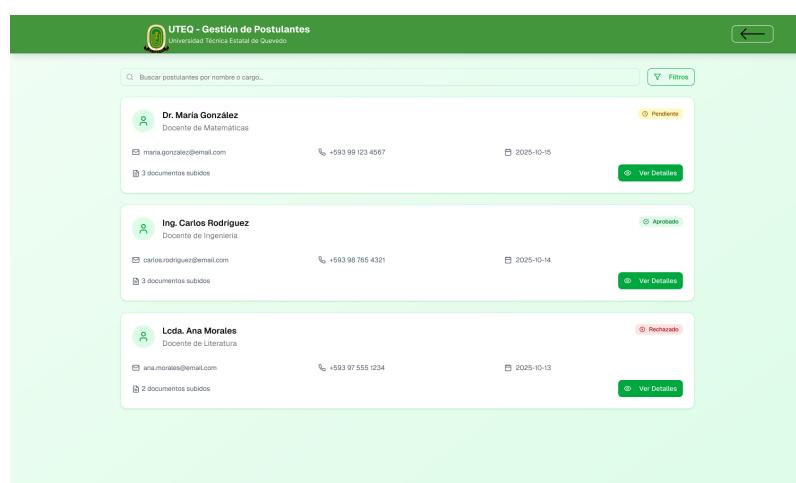


Figura 5: Interfaz de Gestión de Postulantes

The screenshot shows a list of uploaded documents for Dr. María González:

- CV Actualizado - Dr. María González**: Awaiting review (Pending). File size: 2.3 MB.
- Título de Grado - Dr. María González**: Approved. File size: 1.8 MB.
- Certificados - Dr. María González**: Pending review. File size: 3.2 MB.
- CV Actualizado - Ing. Carlos Rodríguez**: Approved. File size: 1.9 MB.
- Másteria - Ing. Carlos Rodríguez**: Approved. File size: 2.8 MB.

Figura 6: Interfaz de Gestión de Documentos

The screenshot shows the ranking of applicants based on their scores:

Postulante	Puntuación	Opciones
Dr. María González Docente de Matemáticas	92.5 puntos	Ver Detalle
Ing. Carlos Rodríguez Docente de Ingeniería	87.3 puntos	Ver Detalle
Dr. Luis Herrera Docente de Química	78.9 puntos	Ver Detalle

Figura 7: Interfaz de Gestión de Evaluaciones

The screenshot shows the selection matrix for non-titular professor positions at the Universidad Técnica Estatal de Quevedo:

UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO - CUADRO DE SELECCIÓN DE PROFESOR NO TITULAR			
PARÁMETROS	HEREDIA CASTRO MIGUEL ALEJANDRO Ingeniero en Informática	RODRIGUEZ VEGA ANDRES FERNANDO Ingeniero en Sistemas	MARTINEZ SOLIS CAROLINA ELIZABETH Ingeniera en Sistemas
<b>A) Título de cuarto nivel</b> Maestría vinculada al campo de empleo de la asignatura <b>MÁXIMO 20 PUNTOS</b>	20	20	20
<b>B) Experiencia en docencia, investigación, vinculación o gestión</b> Experiencia universitaria o posdoctoral <b>MÁXIMO 10 PUNTOS</b>	7	0	0
<b>C) Publicaciones</b> En el año año del concurso <b>MÁXIMO 6 PUNTOS</b>	20	6	4
- Libros (2 puntos por libro)	0	0	0
- Artículos en revistas indexadas regionales (2 puntos)	12	6	4
- Artículos con factor de impacto (4 puntos)	8	0	0

Figura 8: Interfaz de Gestión Evaluaciones 2

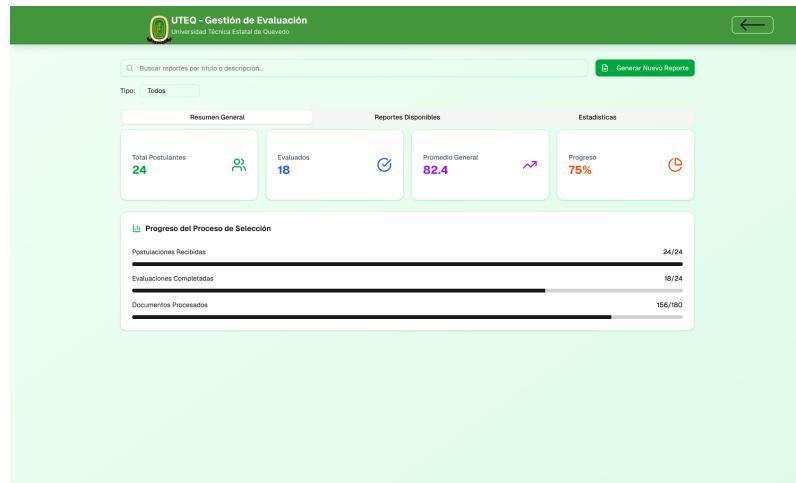


Figura 9: Interfaz de Gestión de Reportes

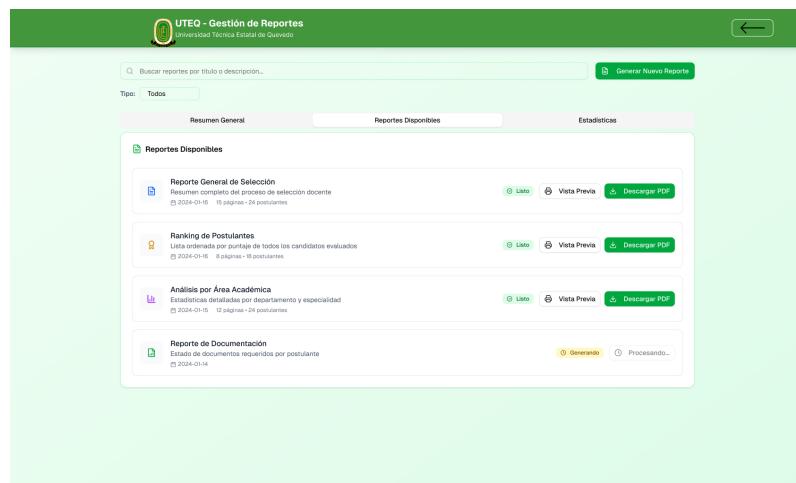


Figura 10: Interfaz de Gestión de Reportes 2

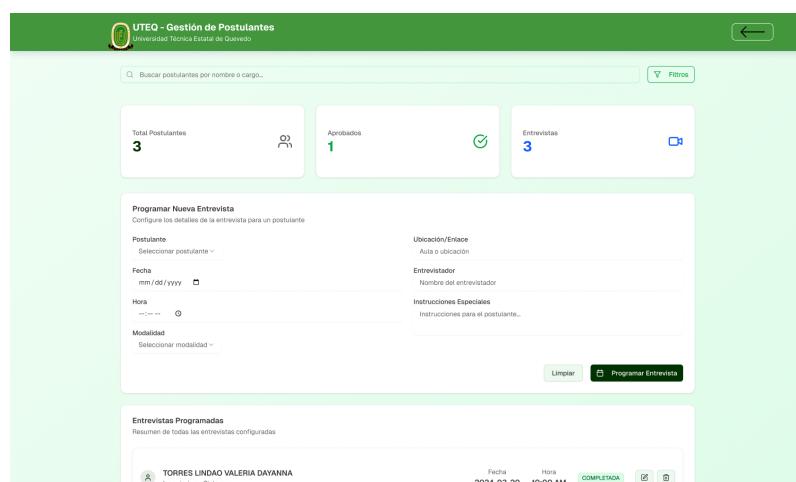


Figura 11: Interfaz de Gestión de Entrevista

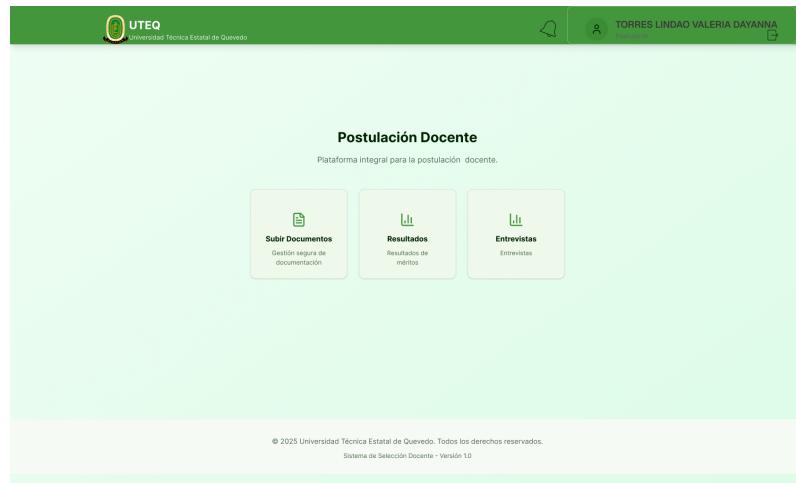


Figura 12: Interfaz de Menú vista postulante

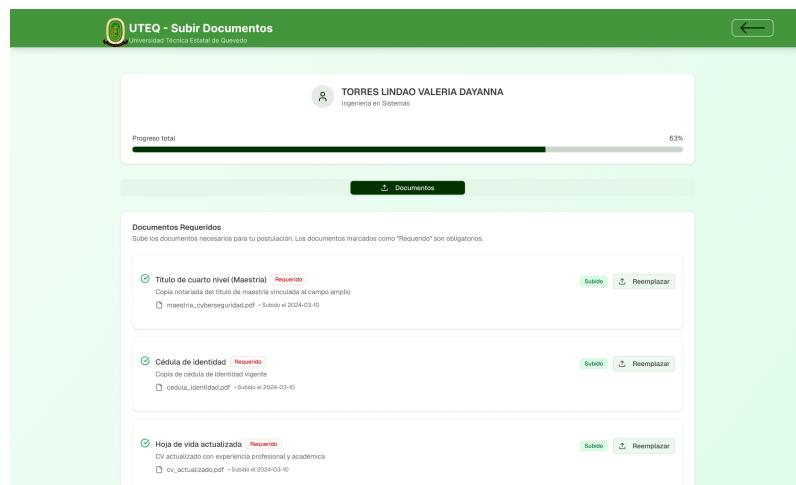


Figura 13: Interfaz de Subir documentos postulante

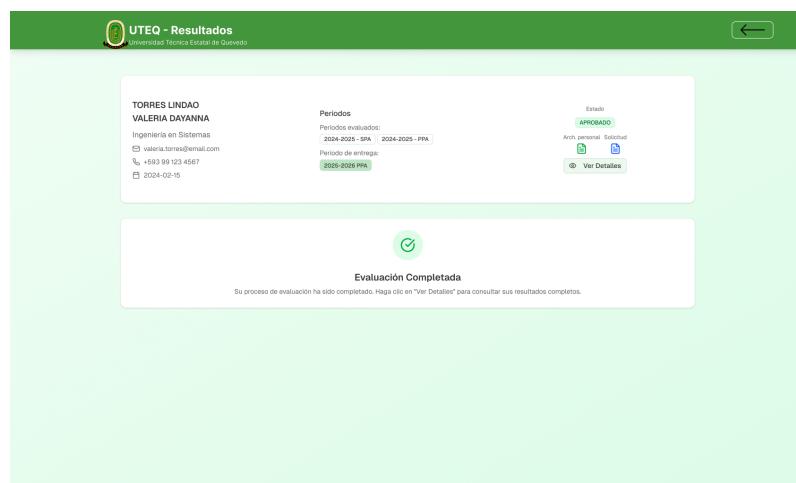


Figura 14: Interfaz de Resultados

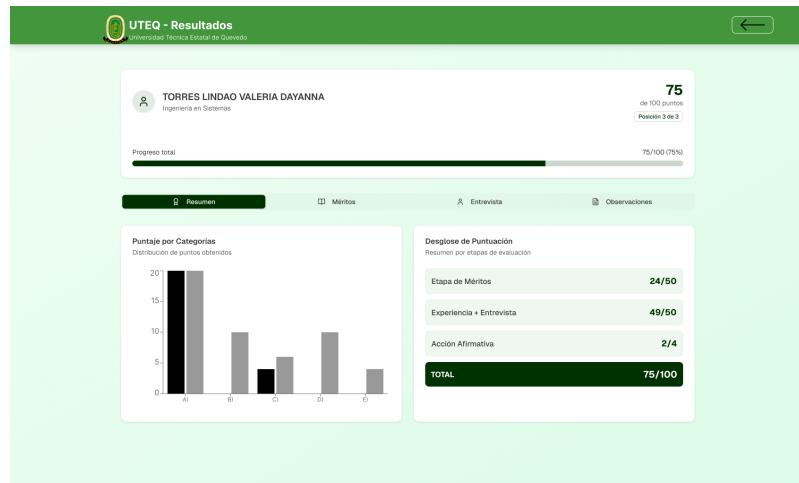


Figura 15: Interfaz de Resultados 1

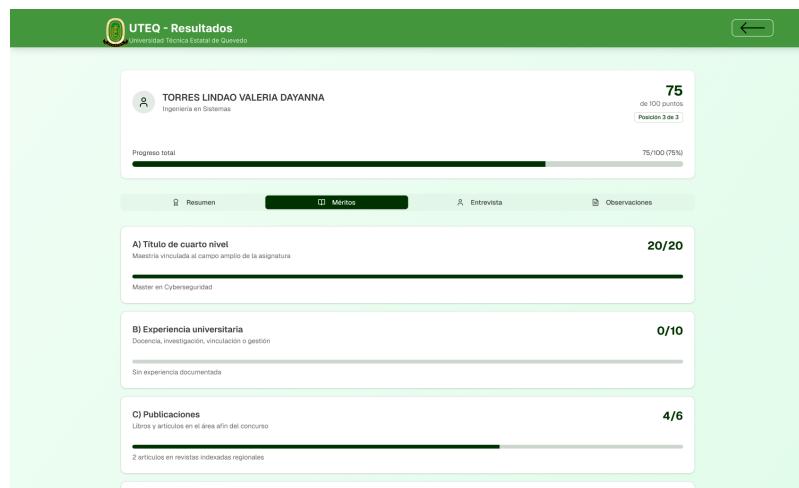


Figura 16: Interfaz de Resultados 2

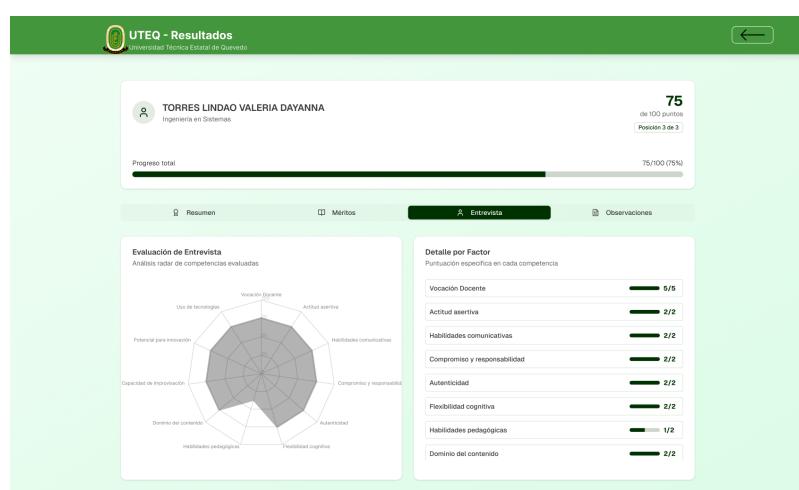


Figura 17: Interfaz de Resultados 3

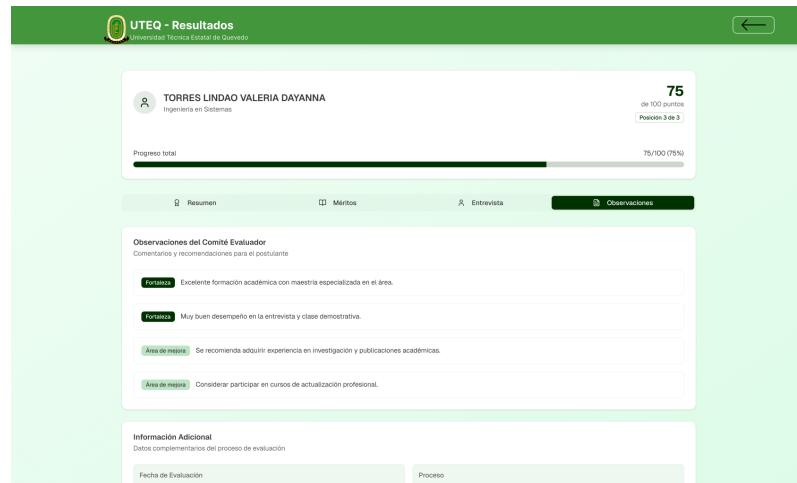


Figura 18: Interfaz de Resultados 4

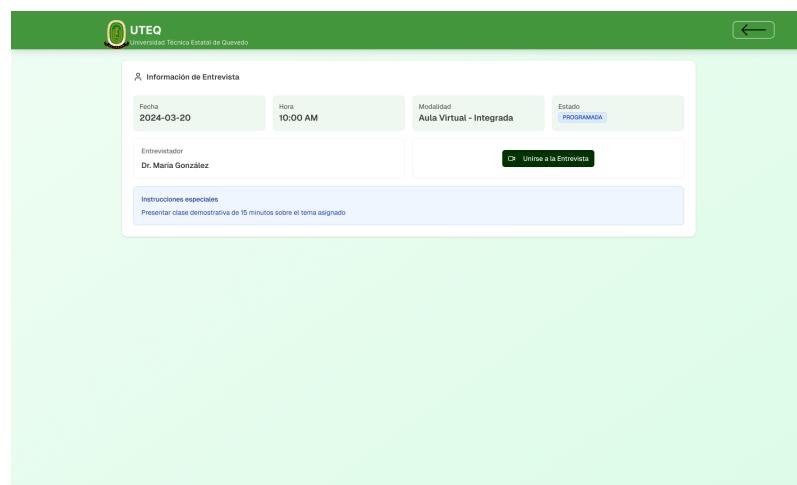


Figura 19: Interfaz de Entrevista de postulante

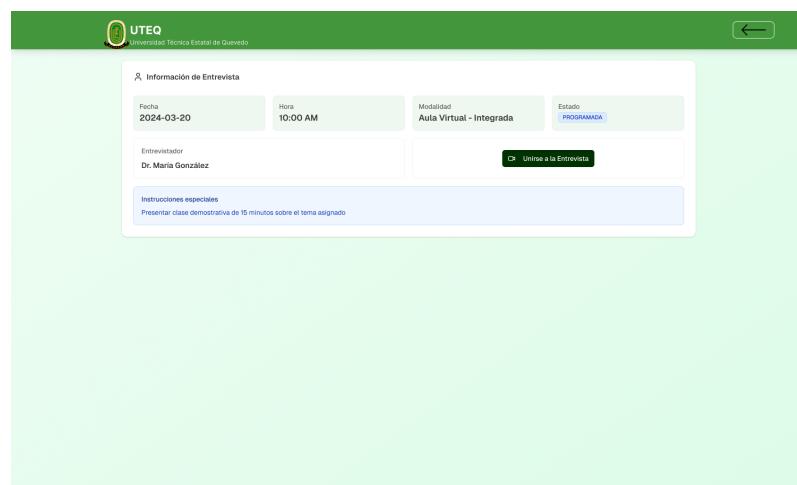


Figura 20: Interfaz de Entrevista de postulante 1

### 11.3. Modelado de la solución

Los artefactos de modelado presentados a continuación son el resultado de la ejecución de sprints iterativos durante la fase de análisis de la solución, donde cada sprint se enfocó en desarrollar tipos específicos de diagramas para los diferentes módulos del sistema, de acuerdo con la metodología Scrum.

#### Diagramas de casos de uso

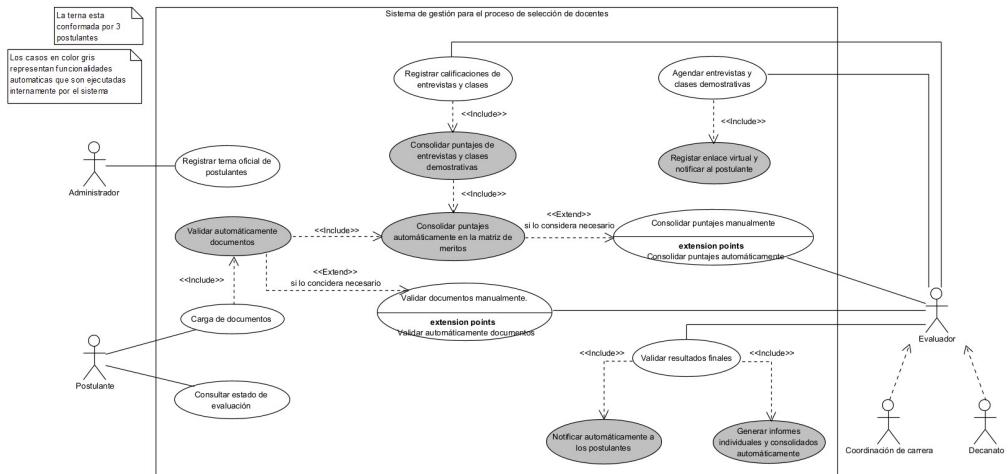


Figura 21: Diagrama de casos de uso general del sistema

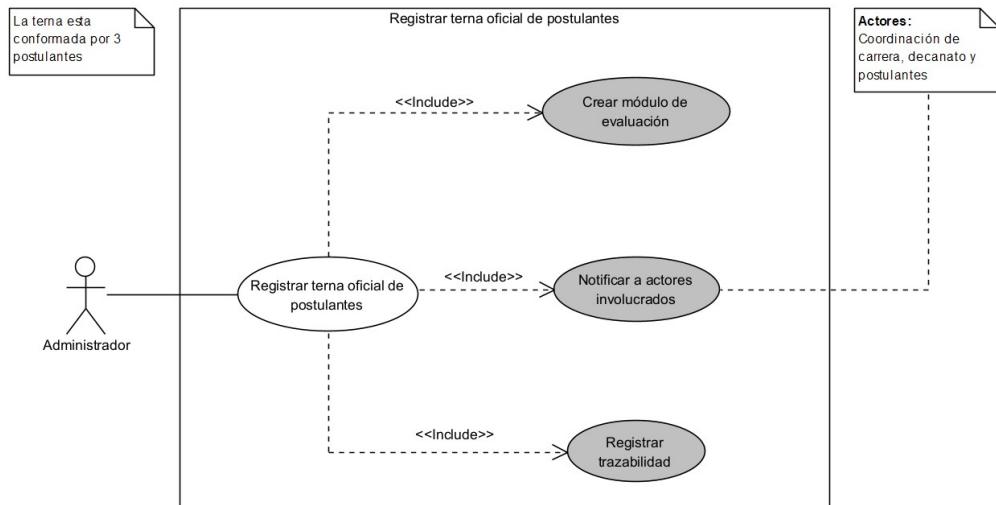


Figura 22: Diagrama de casos de uso Registrar terna oficial de postulantes

## Diseño de interfaz relacionada:

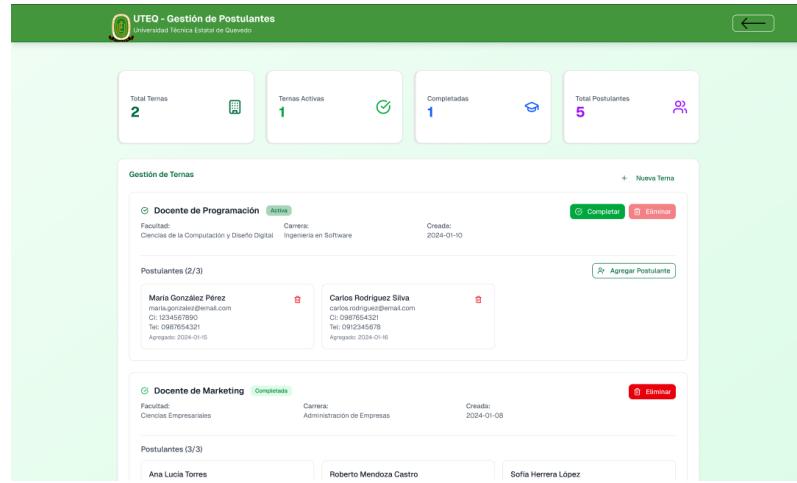


Figura 23: Interfaz de Registro de terna

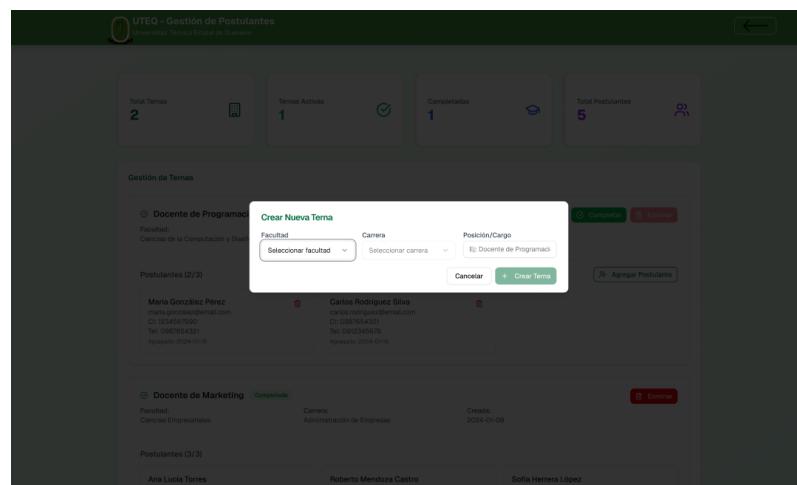


Figura 24: Interfaz de Registro de terna (1)

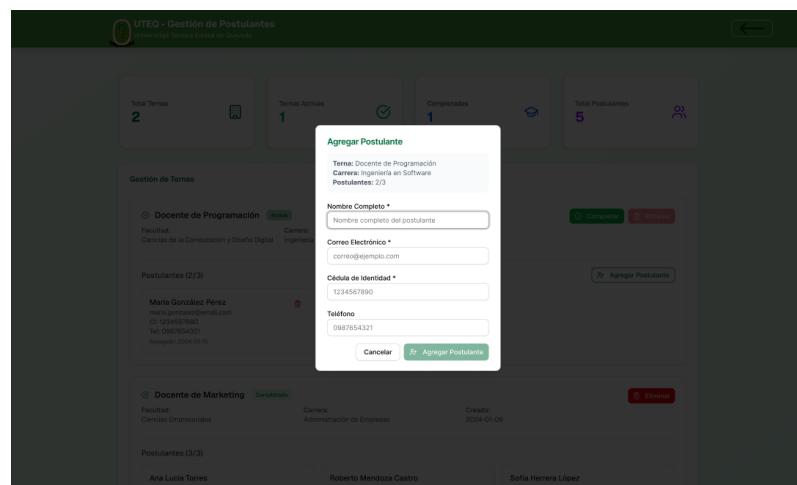


Figura 25: Interfaz de Registro de terna (2)

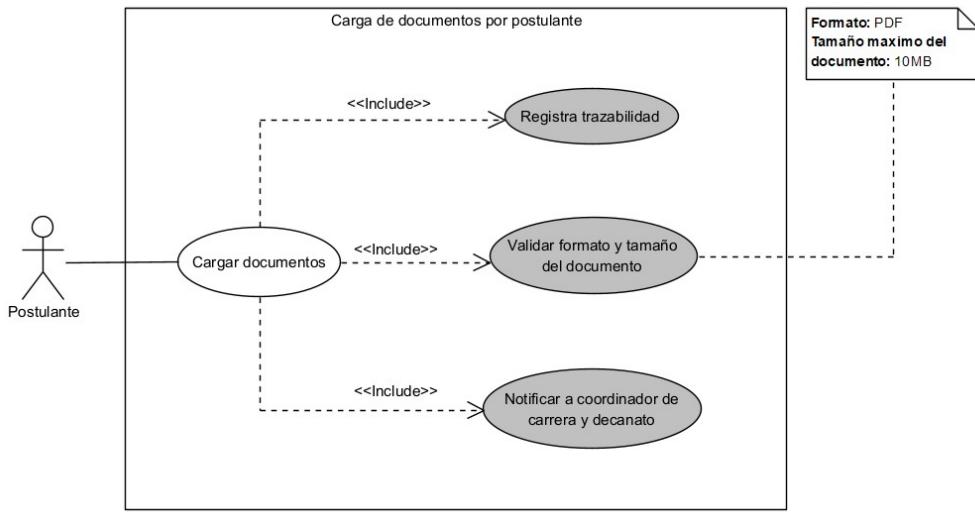


Figura 26: Diagrama de casos de uso Carga de documentos por postulante

### Diseño de interfaz relacionada:

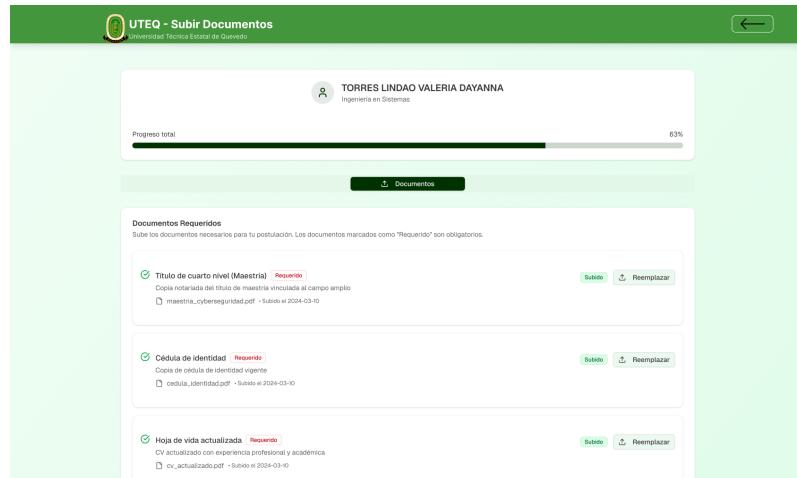


Figura 27: Interfaz de Subir documentos por postulante

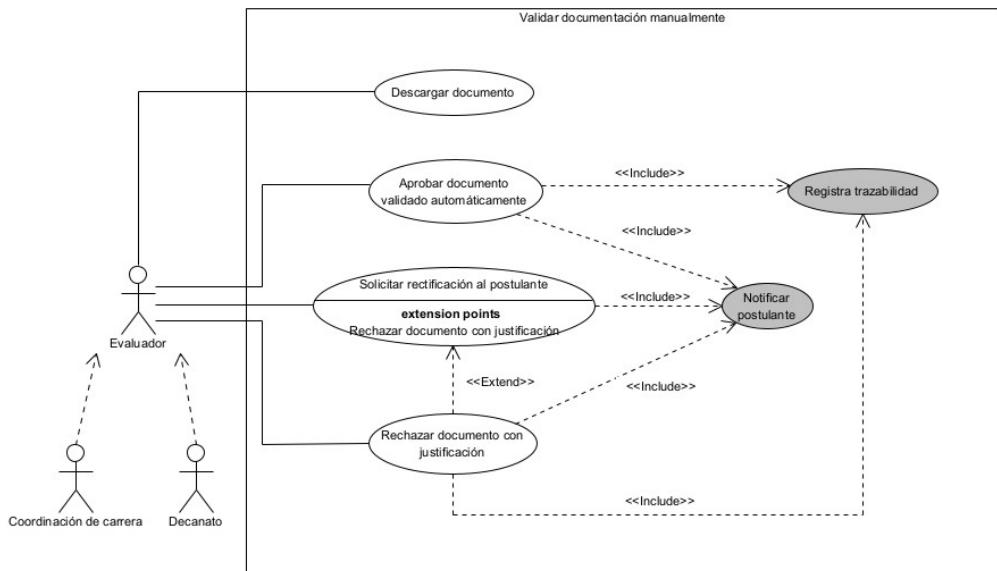


Figura 28: Diagrama de casos de uso Validar documentación manualmente

### Diseño de interfaz relacionada:

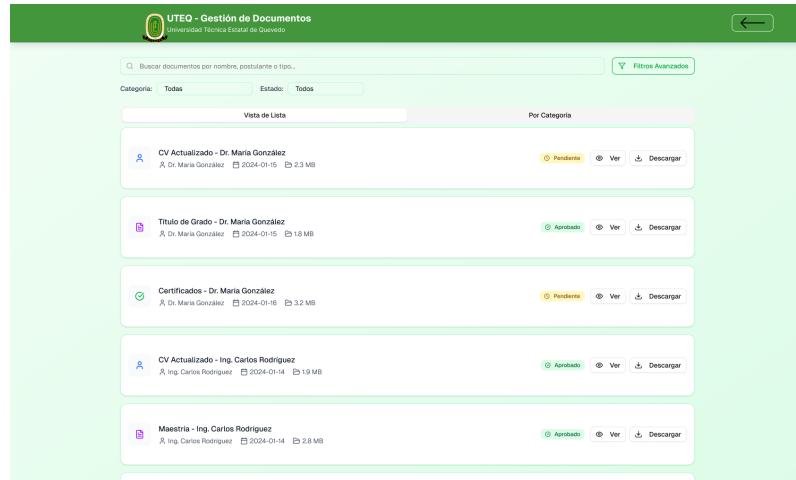


Figura 29: Interfaz de Gestión de documentos

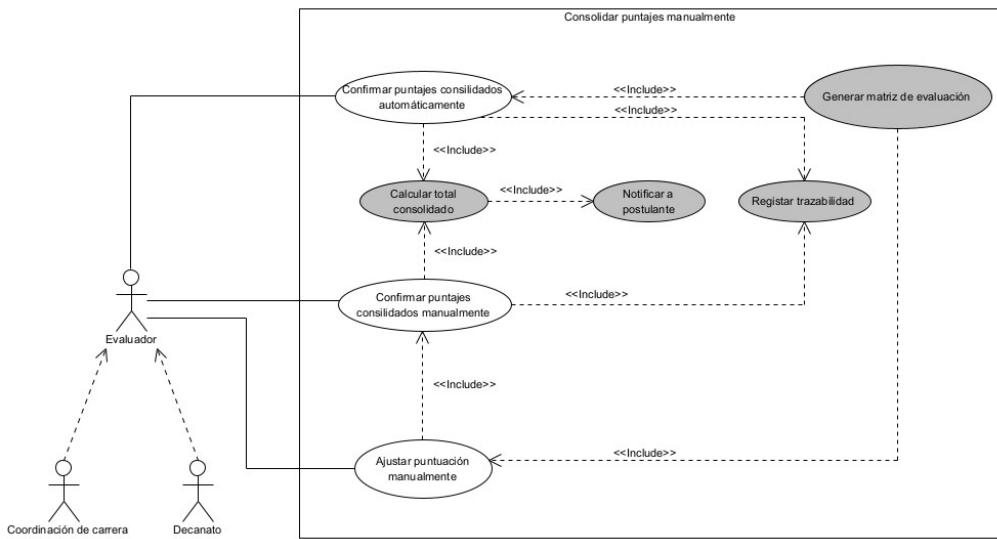


Figura 30: Diagrama de casos de uso Consolidar puntajes manualmente

### Diseño de interfaz relacionada:

UTEQ - Gestión de Evaluación

Ranking de Postulantes

Postulante	Puntaje	Opciones
Dr. María González Docente de Matemáticas	92.5 puntos	<a href="#">Ver Detalle</a>
Ing. Carlos Rodríguez Docente de Ingeniería	87.3 puntos	<a href="#">Ver Detalle</a>
Dr. Luis Herrera Docente de Química	78.9 puntos	<a href="#">Ver Detalle</a>

Figura 31: Interfaz de Gestión de evaluación

UTEQ - Gestión de Evaluación

UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO - CUADRO DE SELECCIÓN DE PROFESOR NO TITULAR

PARÁMETROS	HERREIRA CASTRO MIGUEL JAVIER Ingeniería en Sistemas	RODRIGUEZ VEGA ANDRES FERNANDO Ingeniería en Sistemas	MARTINEZ SILVA CAROLINA ELIZABETH Ingeniería en Sistemas
A) Trabajo de cuarto año Máximo incluido al campo amplio de la asignatura MÁXIMO 20 PUNTOS	20	20	20
B) Experiencia en docencia, investigación, vinculación o gestión Experiencia universitaria o posterior MÁXIMO 10 PUNTOS	7	0	0
C) Publicaciones En el año año del concurso MÁXIMO 8 PUNTOS	20	6	4
- Libros (2 puntos por libro)	0	0	0
- Artículos en revistas indexadas regionales (2 puntos)	12	6	4
- Artículos con factor de impacto (4 puntos)	8	0	0

Figura 32: Interfaz de Matriz de méritos

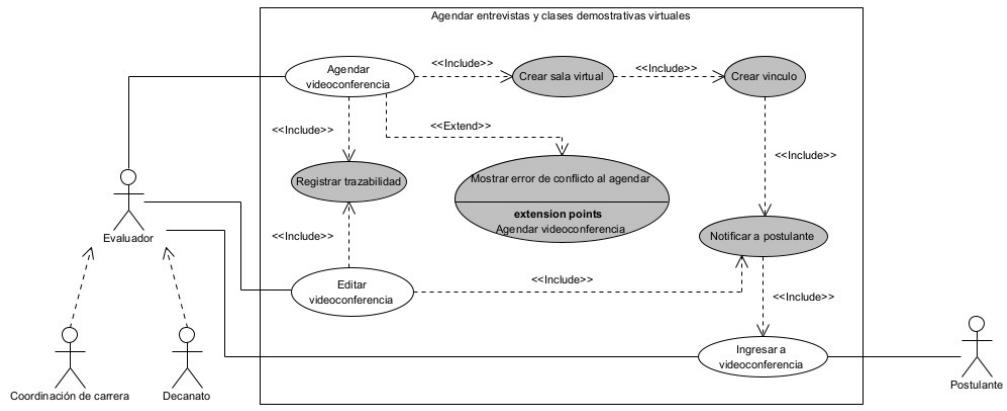


Figura 33: Diagrama de casos de uso Agendar entrevistas y clases demostrativas

## Diseño de interfaz relacionada:

The screenshot shows the UTEQ - Gestión de Postulantes interface. At the top, there's a navigation bar with the university logo and a search bar. Below it, three summary boxes show the total number of applicants (3), approved applicants (1), and interviewed applicants (3). A large central panel titled "Programar Nueva Entrevista" (Schedule New Interview) contains fields for selecting the applicant, setting the date and time, choosing the interviewer, and specifying location and instructions. Below this is a section for "Entrevistas Programadas" (Scheduled Interviews) which lists a single entry: "TORRES LINDAO VALERIA DAYANNA" from "Ingeniería en Sistemas" with a scheduled date of "2024-03-20" at "10:00 AM" and status "COMPLETADA".

Figura 34: Interfaz de Programación de entrevista

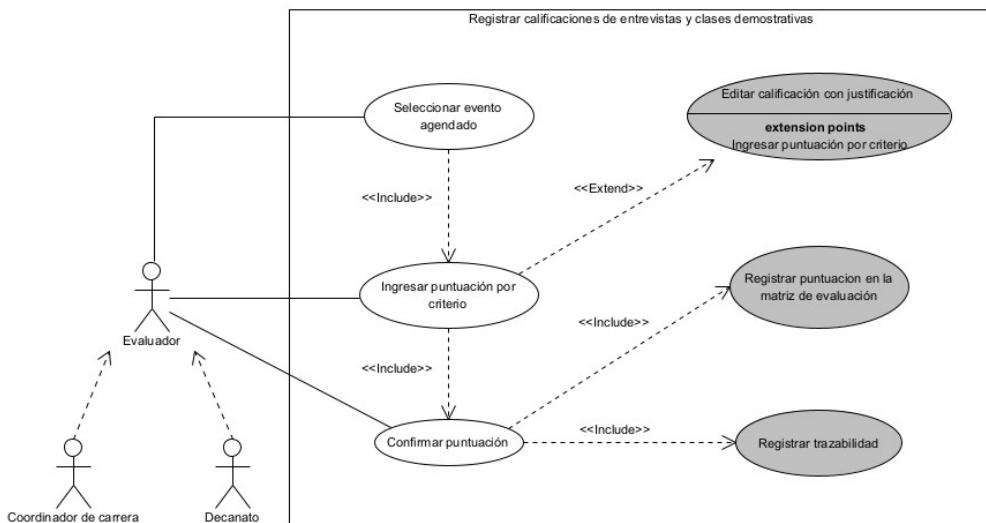


Figura 35: Diagrama de casos de uso Registrar calificaciones de entrevistas y clases demostrativas

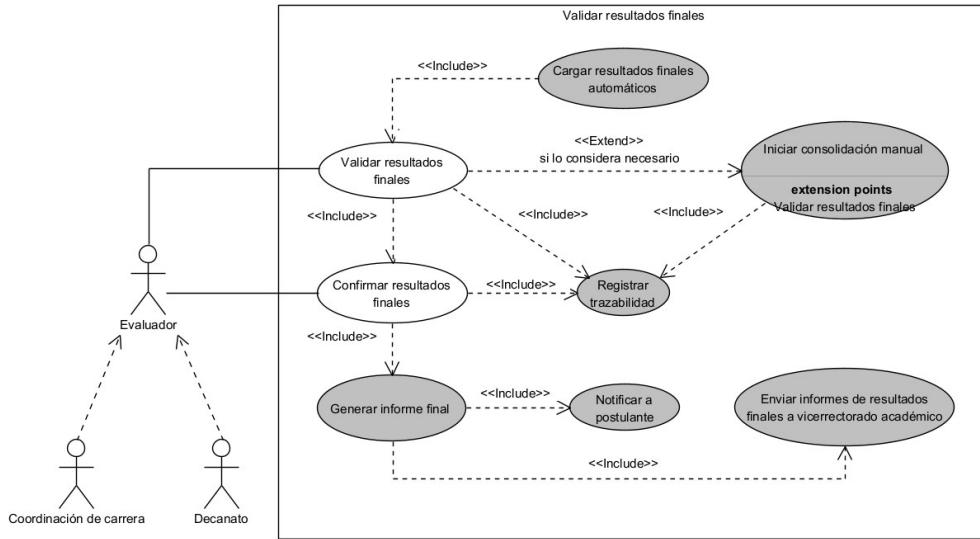


Figura 36: Diagrama de casos de uso Validar resultados finales

## Diagrama de clases

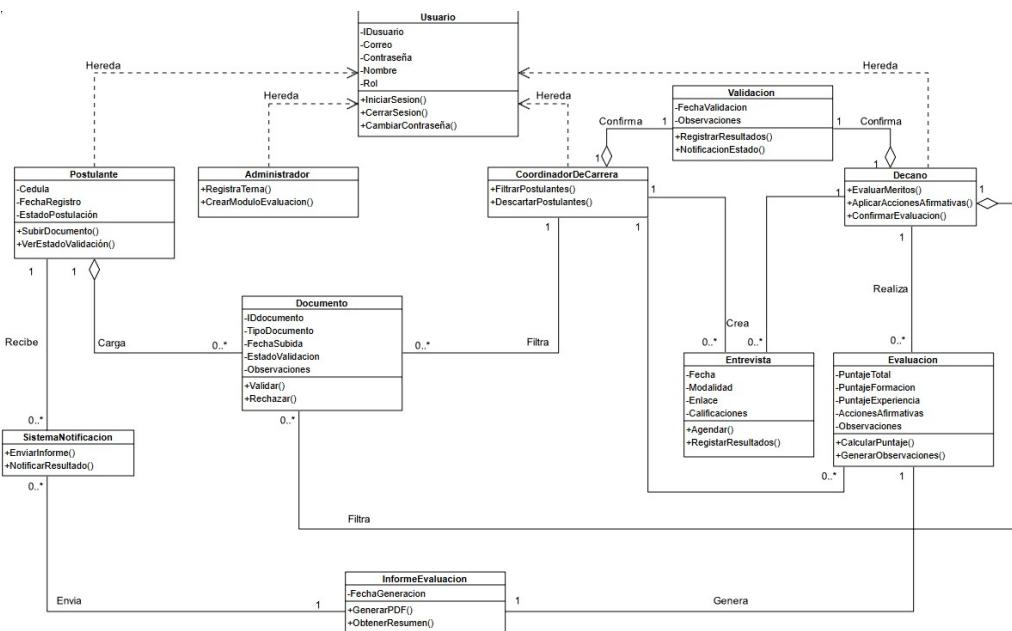


Figura 37: Diagrama de clases

## Diagramas de actividades

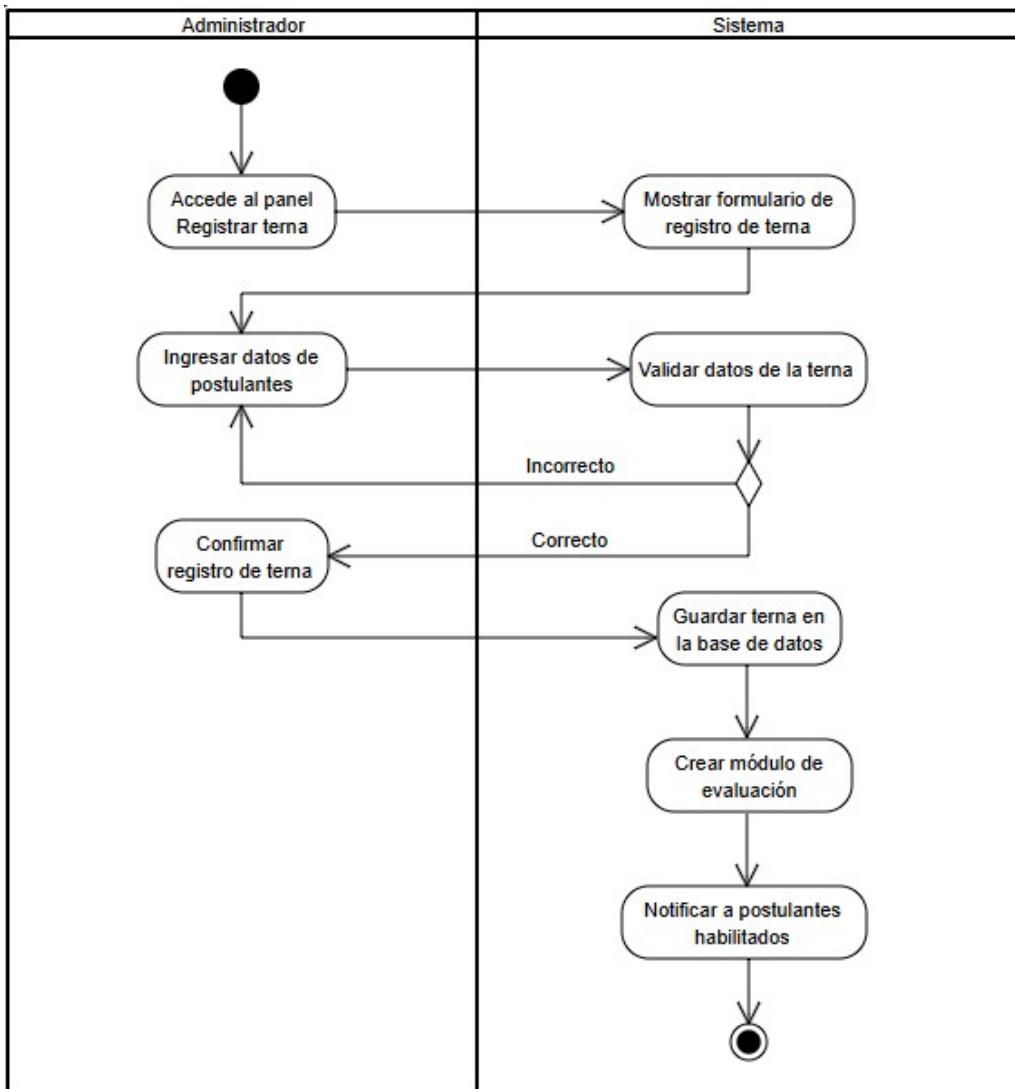


Figura 38: Diagrama de actividades de Registro de terna y creación del módulo de evaluación

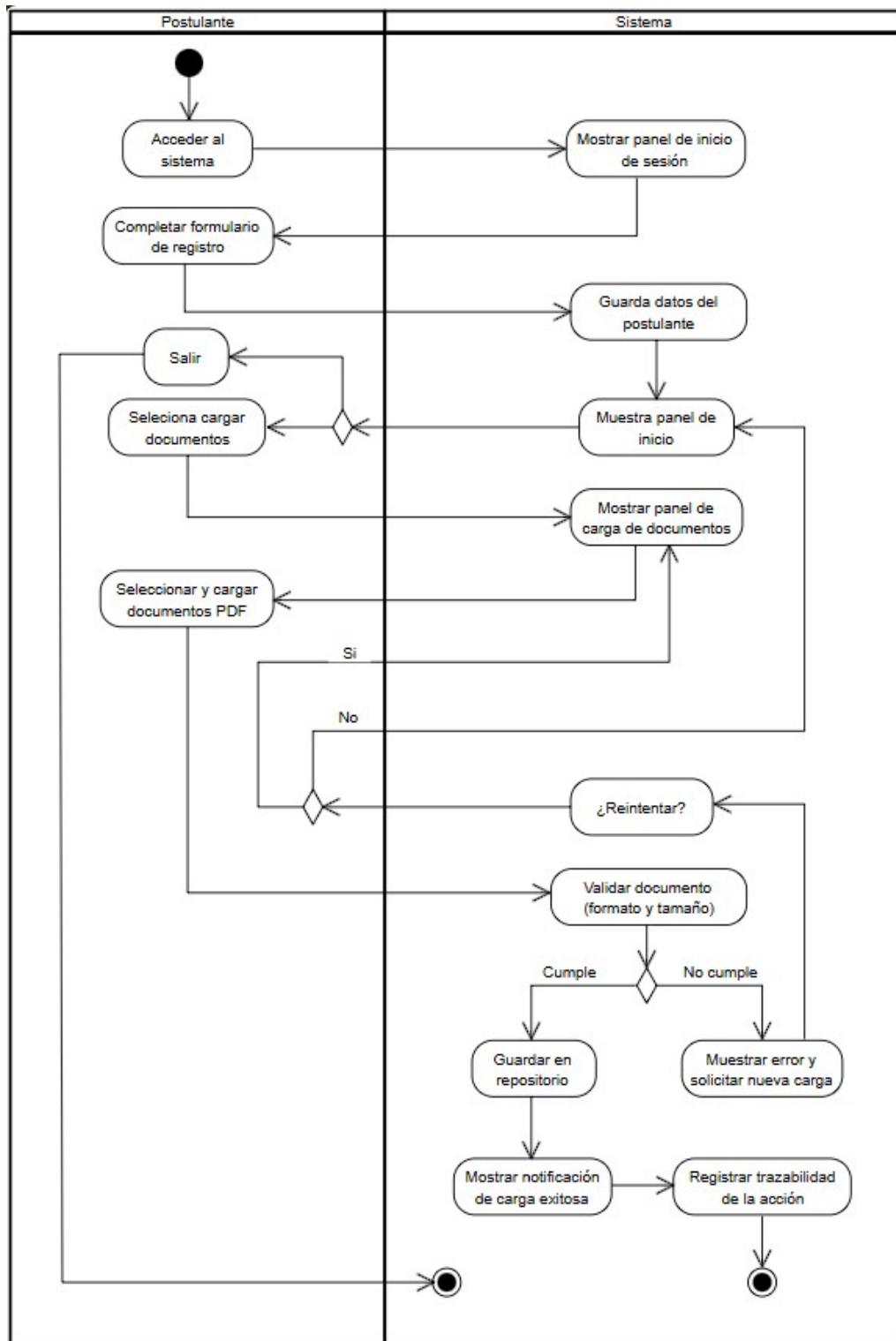


Figura 39: Diagrama de actividades de Registro y carga de documentos por postulante

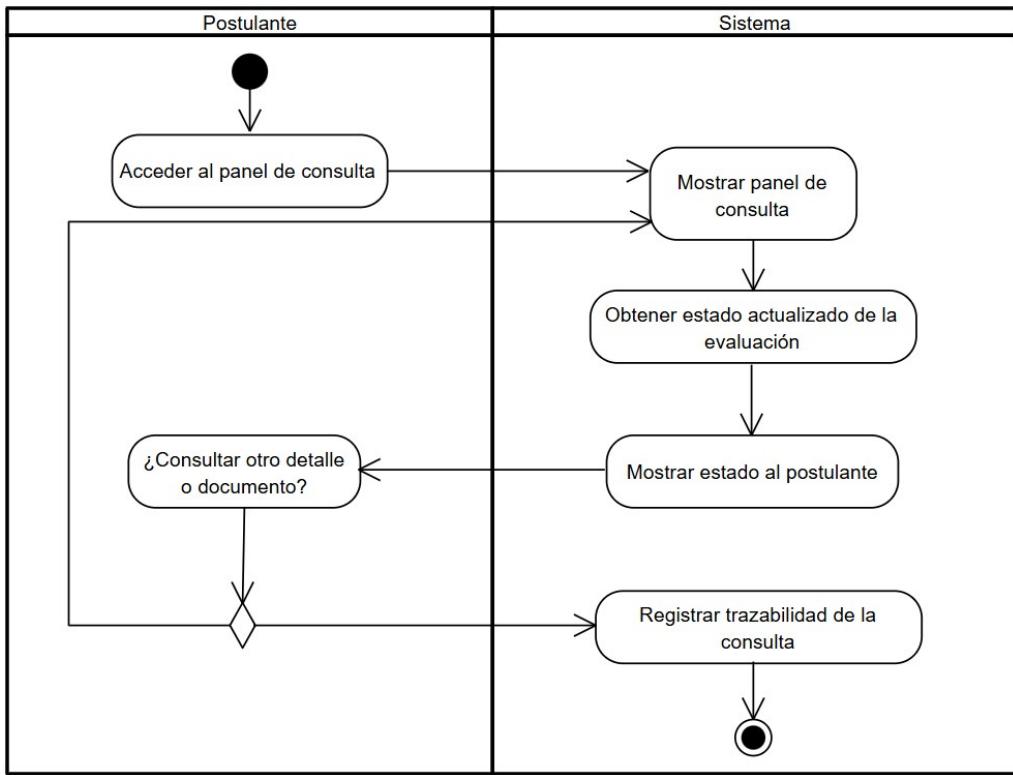


Figura 40: Diagrama de actividades de Consulta de estado del proceso

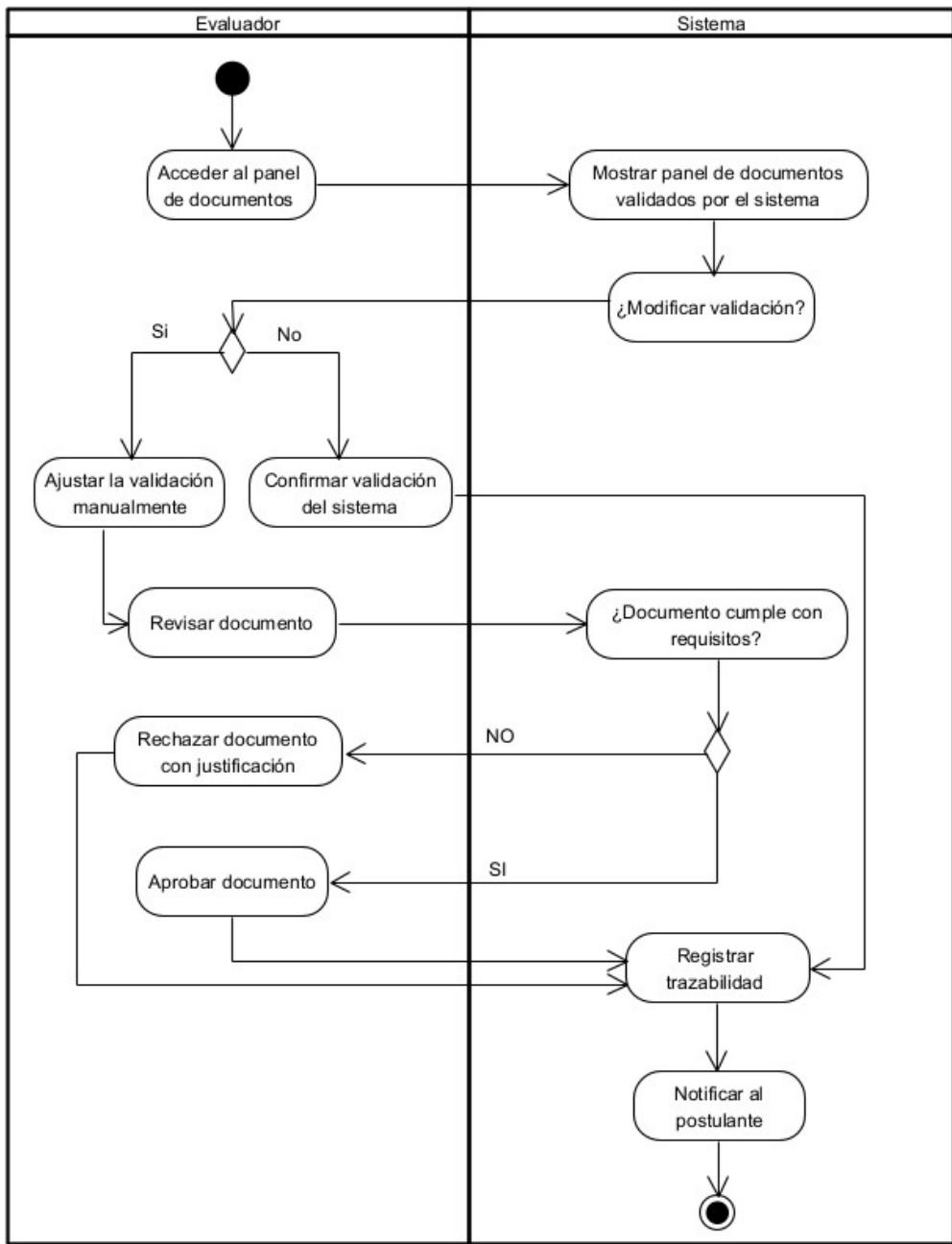


Figura 41: Diagrama de actividades de Revisión y validación de documentos

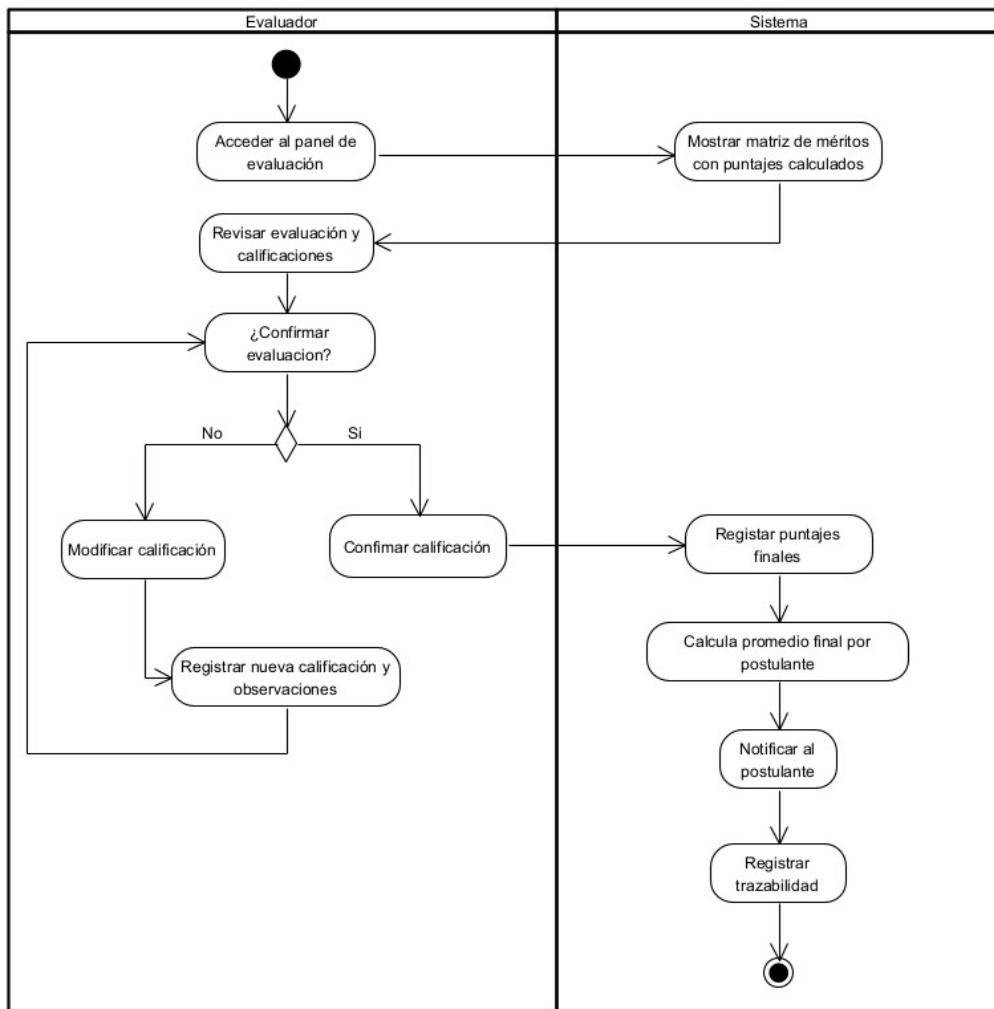


Figura 42: Diagrama de actividades de Evaluación automática de méritos

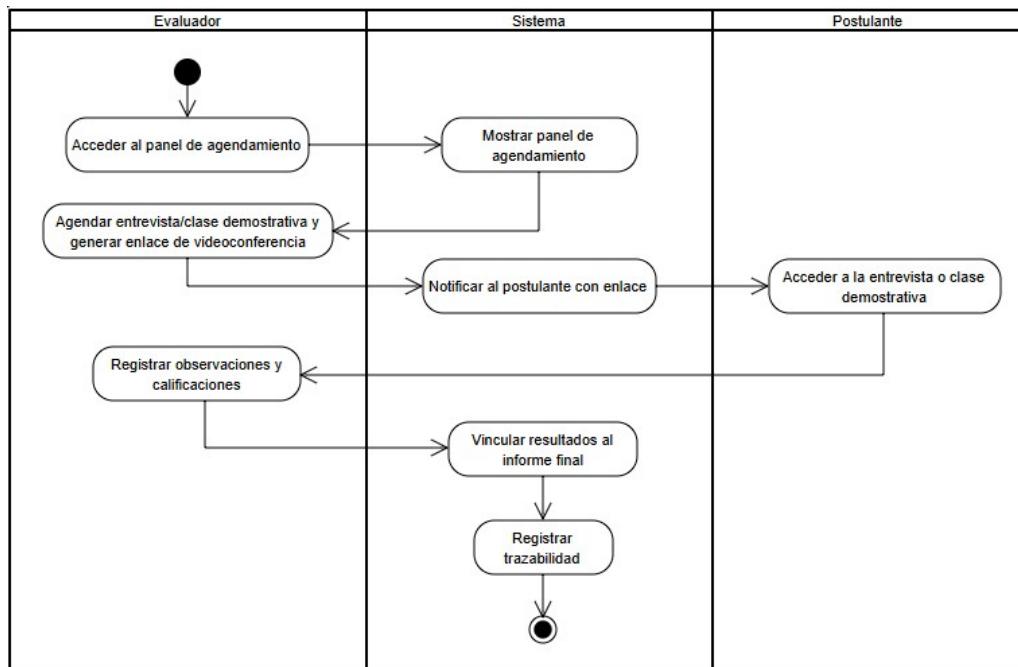


Figura 43: Diagrama de actividades de Gestión de entrevistas y clases demostrativas

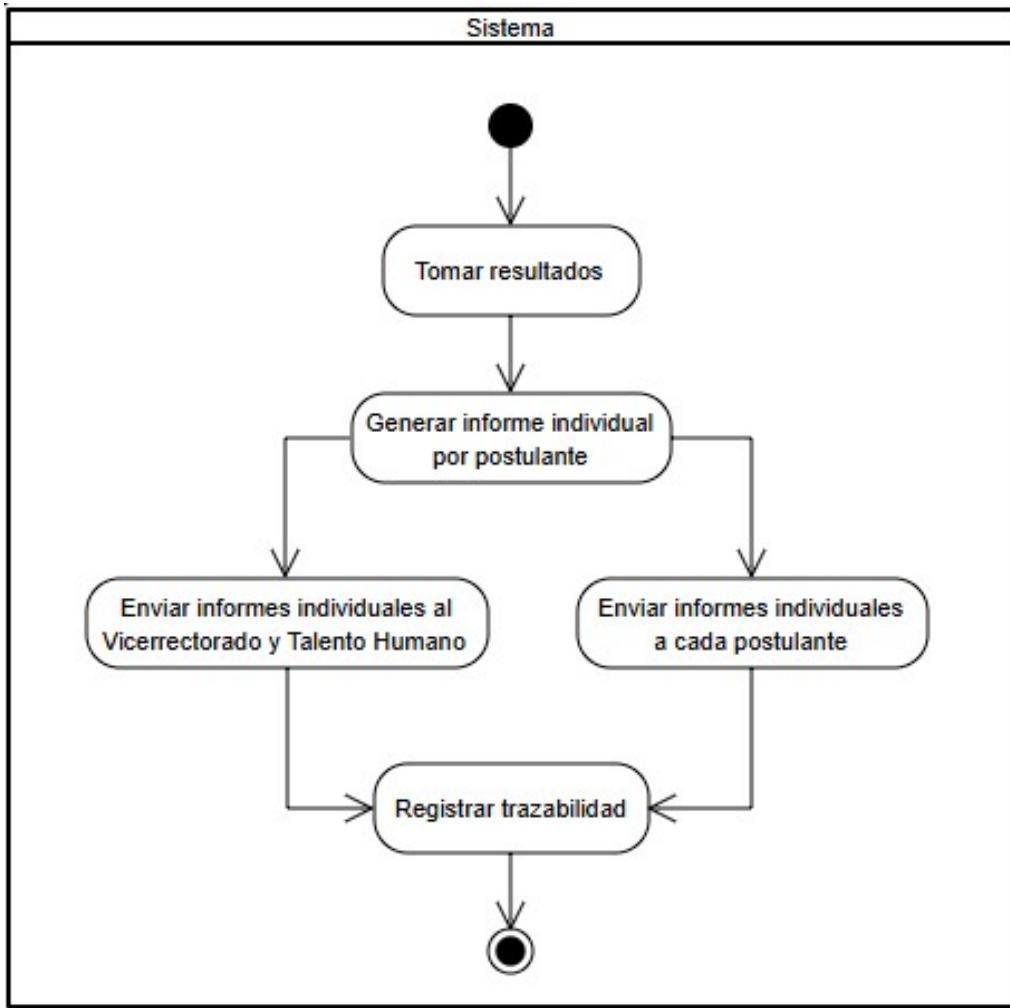


Figura 44: Diagrama de actividades de Generación y envío de informes

### Diagrama de secuencia

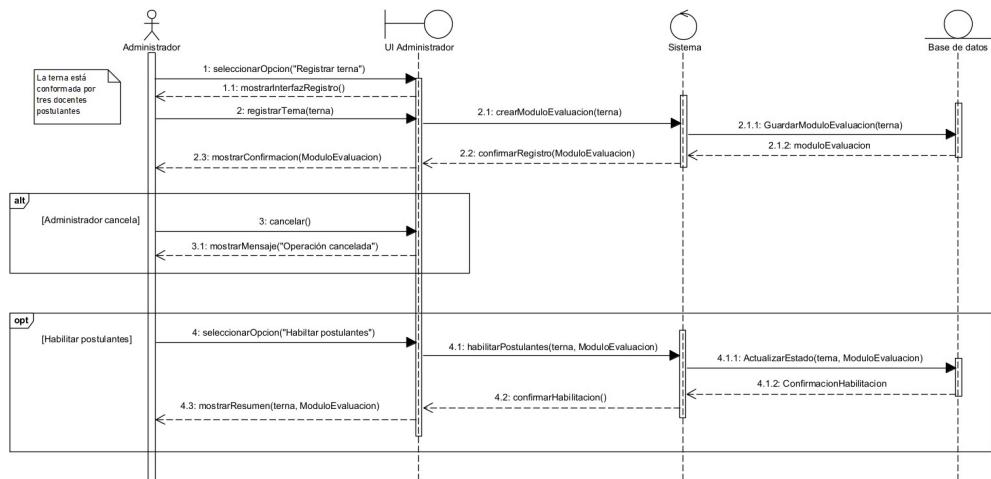


Figura 45: Diagrama de secuencia Registro de terna

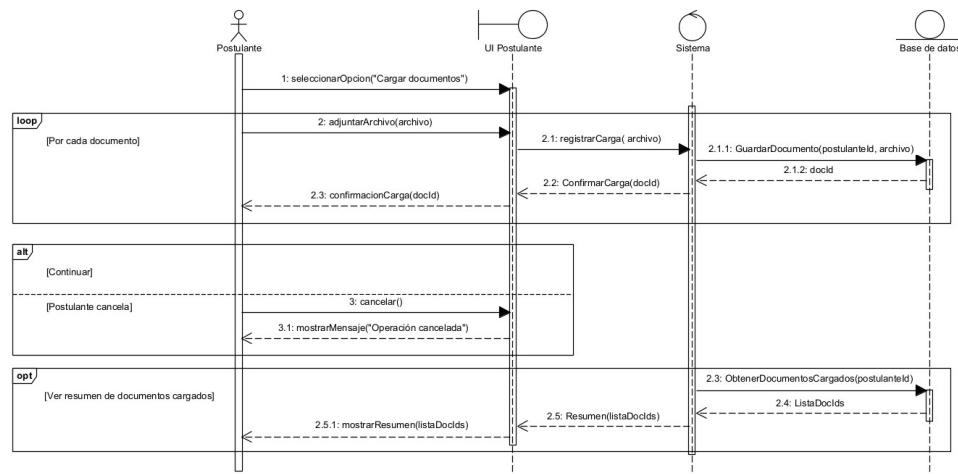


Figura 46: Diagrama de secuencia Carga de documentos por el postulante

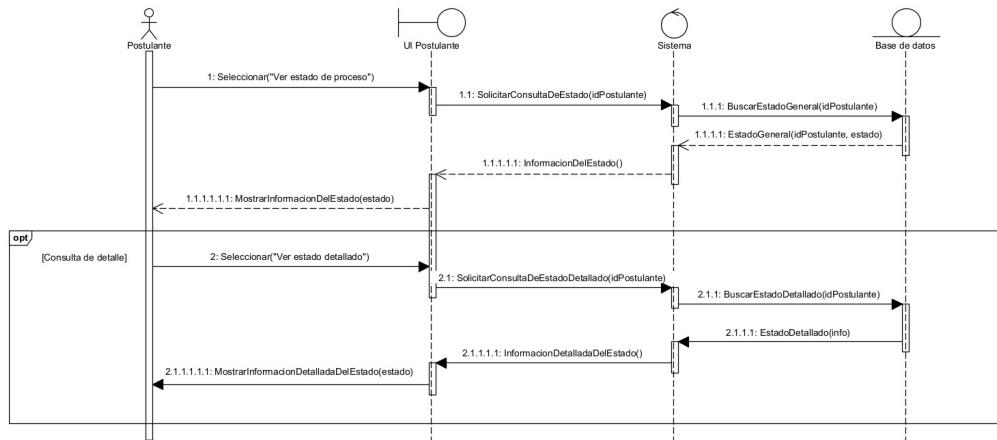


Figura 47: Diagrama de secuencia Consulta del estado del proceso

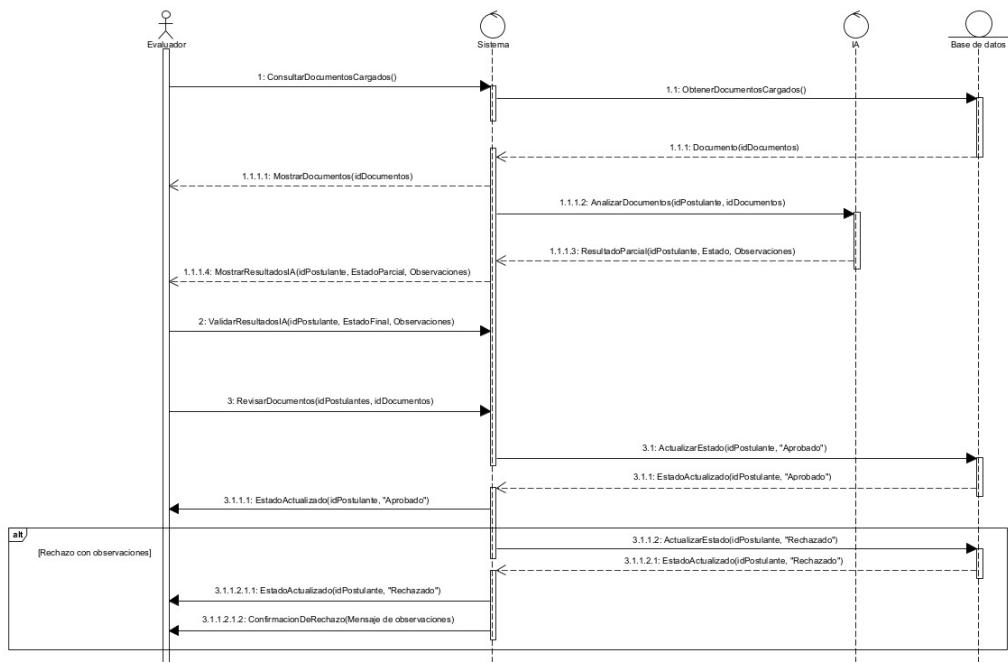


Figura 48: Diagrama de secuencia Revisión y validación de documentos

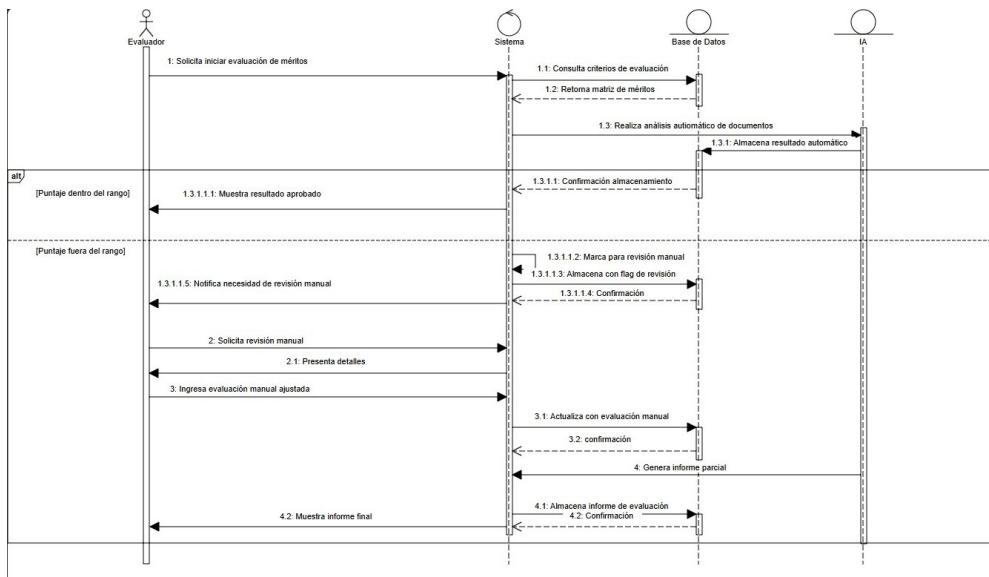


Figura 49: Diagrama de secuencia Evaluación automática de méritos

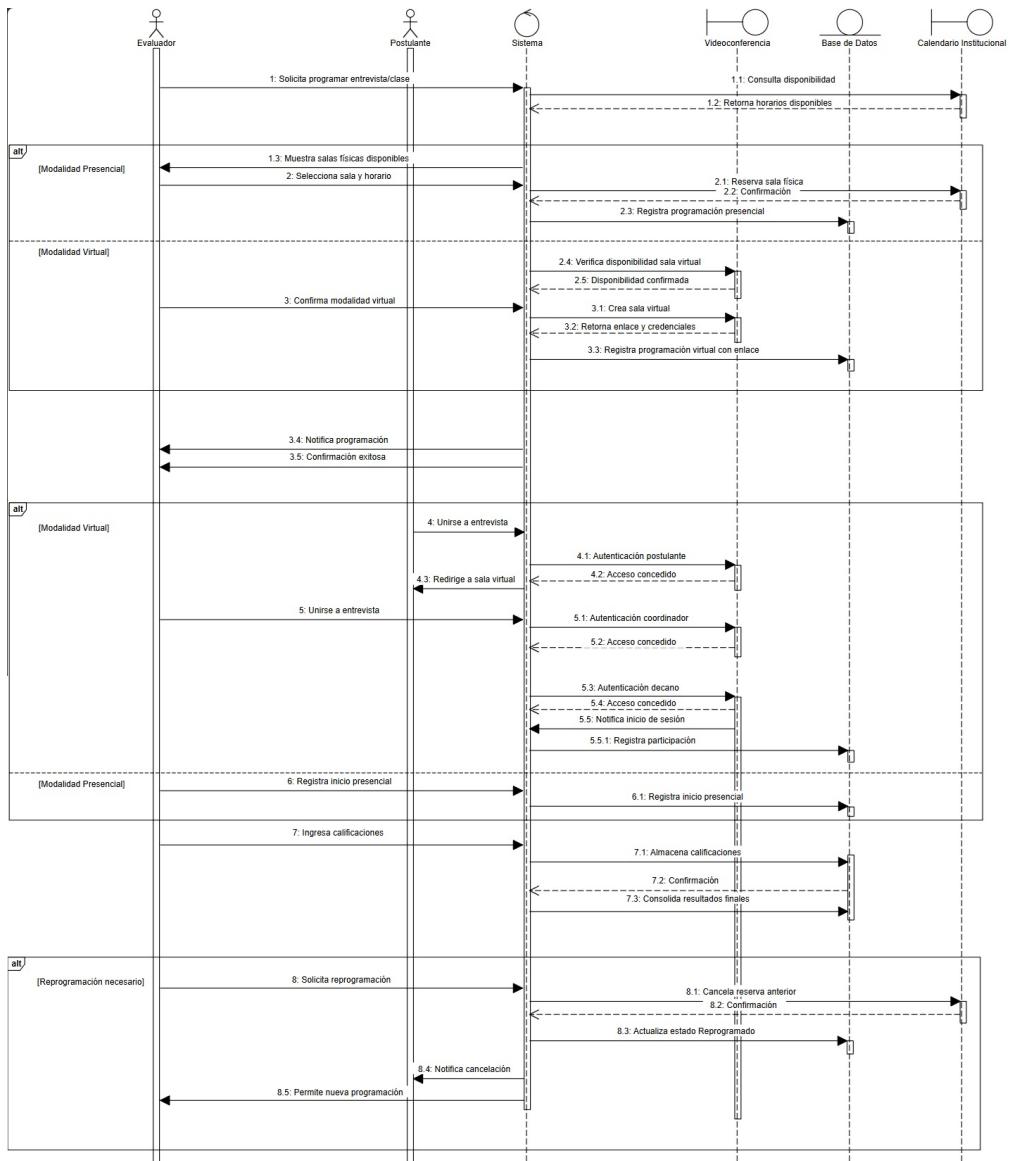


Figura 50: Diagrama de secuencia Gestión de entrevistas y clases demostrativas

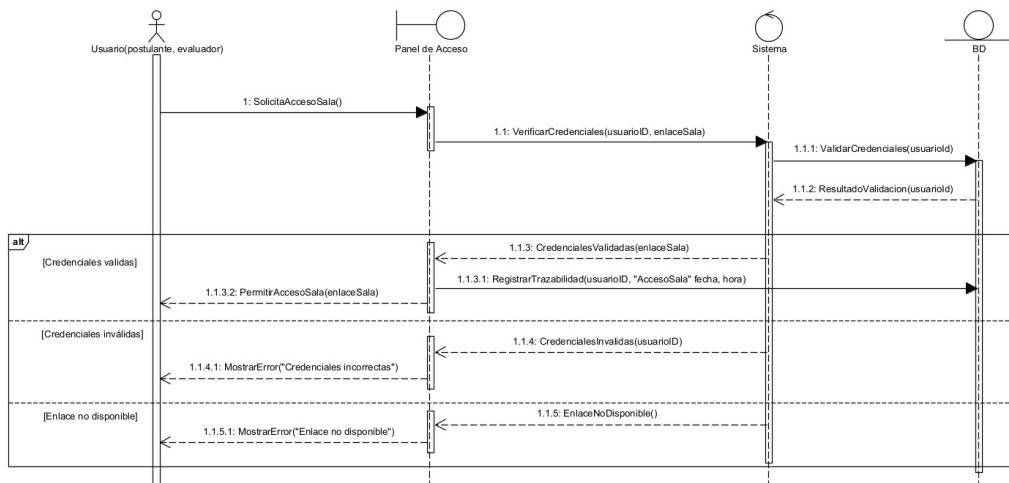


Figura 51: Diagrama de secuencia Acceso a sala de videoconferencia

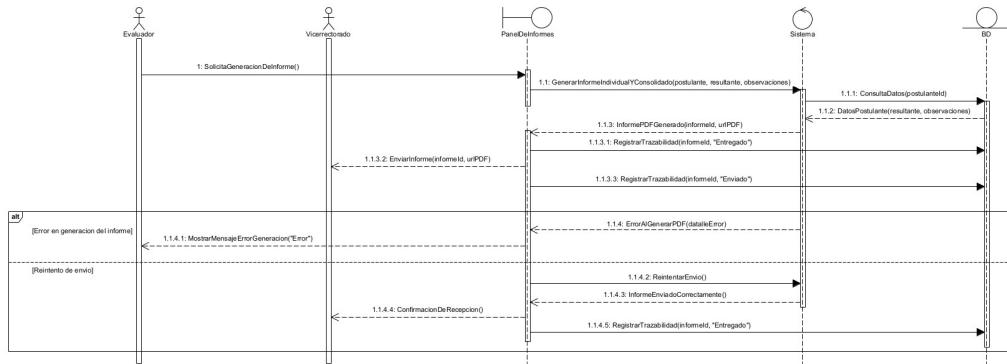


Figura 52: Diagrama de secuencia Generación y envío de informes

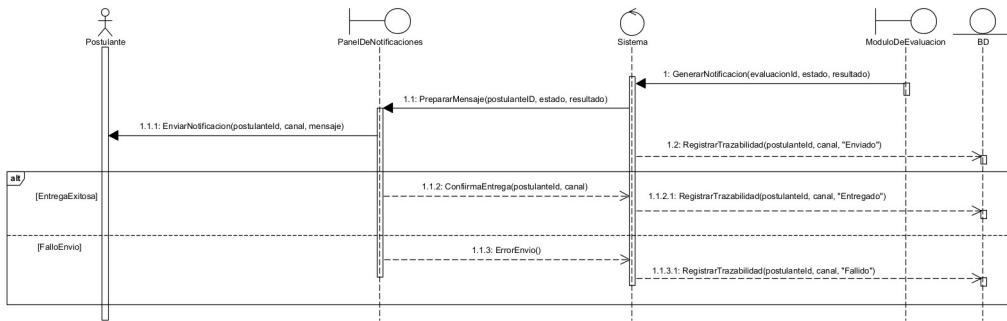


Figura 53: Diagrama de secuencia Notificación de estado y resultados al postulante

## Diagrama de despliegue

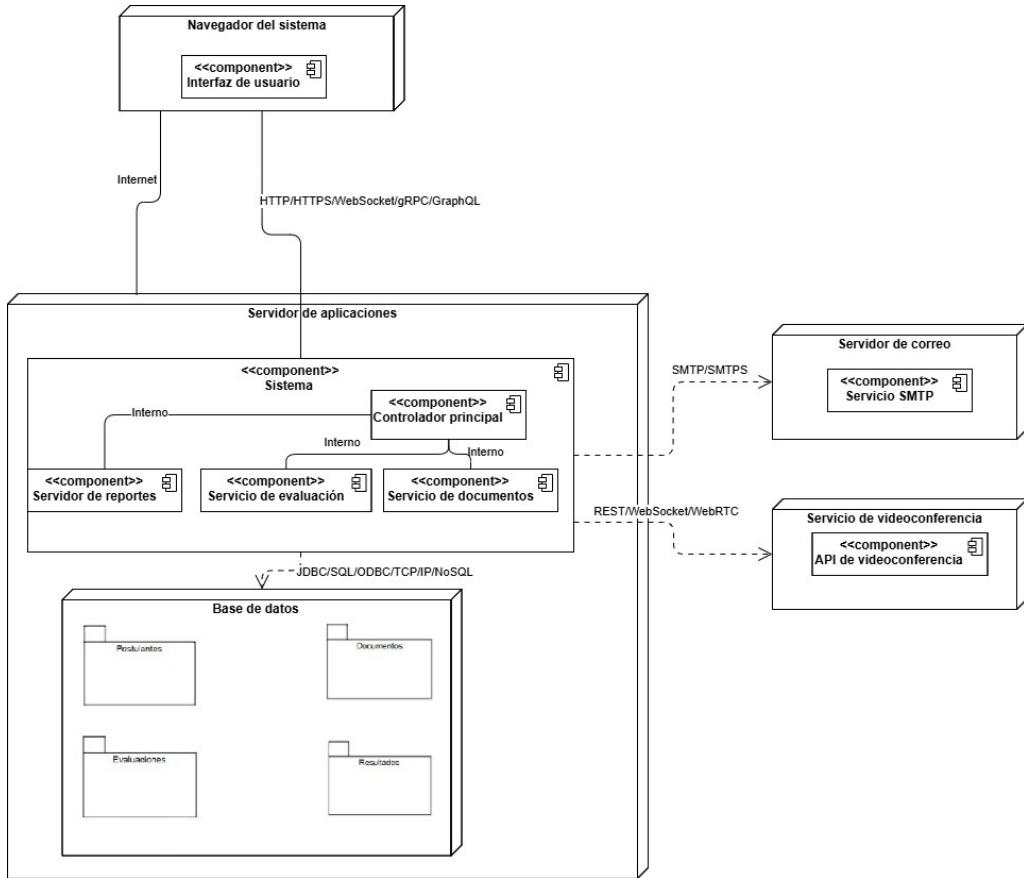


Figura 54: Diagrama de despliegue de la solución

## Diagrama de componentes

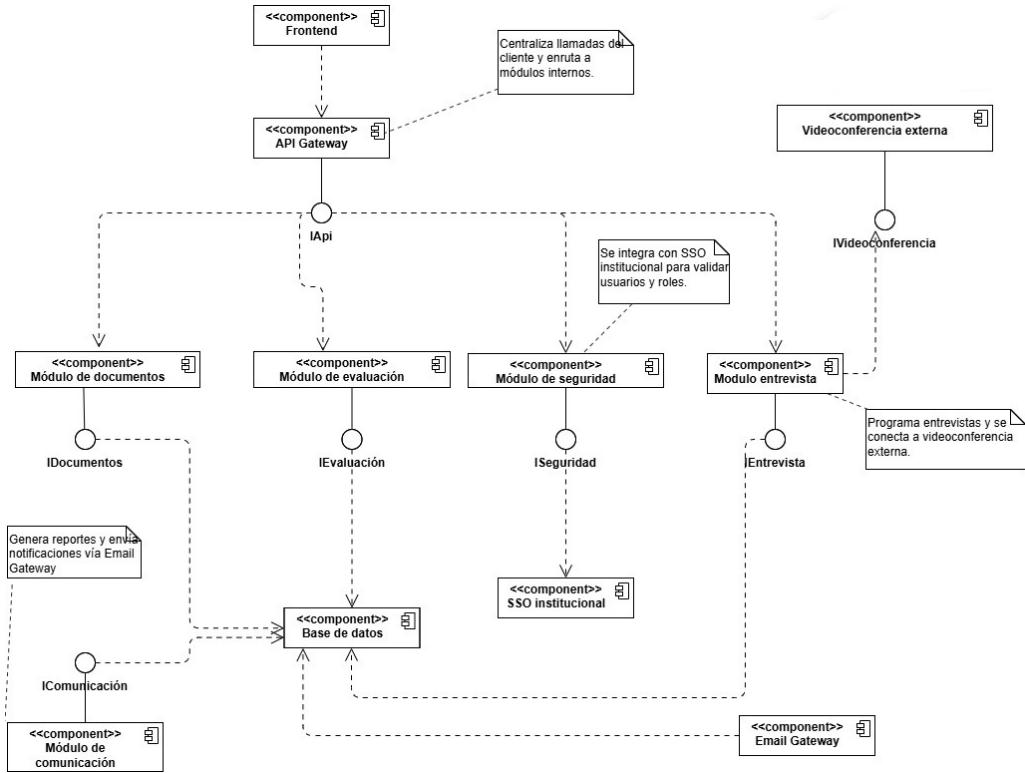


Figura 55: Diagrama de componentes de la solución