



# Especificación De Requerimientos De Software (SRS)

Plataforma Freelance para Jóvenes: Conectando Talento con Oportunidades Laborales

Estudiantes:

Andrés Felipe Vélez Echeverry

Juan Camilo Gómez Contento

Thomas Alejandro Orjuela Bello

Estudiante	Documento	Celular	Correo Javeriano
Andrés Felipe Vélez Echeverry	cc. 1000942132	318-514-3658	andresveleze@javeriana.edu.co
Juan Camilo Gómez Contento	cc. 1019982319	321-4567-243	jcamilo.gomez@javeriana.edu.co
Thomas Alejandro Orjuela Bello	cc. 1014658502	313-8098727	talejandrorjuela@javeriana.edu.co

Director:

Pablo Andrés Márquez Fernández, Ing..

Director	Documento	Celular	Teléfono Fijo	Correo Javeriano
Pablo Andrés Márquez Fernández	cc. 80.756.159	318-712-3845	NA	marquezp-a@javeriana.edu.co

Pontificia Universidad Javeriana

Facultad de Ingeniería

Ingeniería de Sistemas

# Índice

<b>1</b>	<b>Introducción</b>	<b>5</b>
<b>2</b>	<b>Propósito</b>	<b>5</b>
<b>3</b>	<b>Alcance</b>	<b>5</b>
<b>4</b>	<b>Definiciones, acrónimos y definiciones</b>	<b>6</b>
<b>5</b>	<b>Interfaces con el sistema</b>	<b>6</b>
<b>6</b>	<b>Interfaces con el hardware</b>	<b>7</b>
<b>7</b>	<b>Interfaces con el software</b>	<b>8</b>
<b>8</b>	<b>Interfaces de comunicación</b>	<b>9</b>
<b>9</b>	<b>Funciones del producto</b>	<b>9</b>
<b>10</b>	<b>Características del Usuario</b>	<b>11</b>
10.1	Estudiante - Freelancer . . . . .	12
10.2	Contratista . . . . .	12
<b>11</b>	<b>Restricciones</b>	<b>13</b>
<b>12</b>	<b>Modelo del dominio</b>	<b>13</b>
<b>13</b>	<b>Suposiciones y dependencias</b>	<b>15</b>
<b>14</b>	<b>Requerimientos Específicos</b>	<b>15</b>
<b>15</b>	<b>Interfaces con el usuario</b>	<b>18</b>
<b>16</b>	<b>Características del producto de software</b>	<b>19</b>
<b>17</b>	<b>Requerimientos de desempeño</b>	<b>19</b>
17.1	Concurrencia . . . . .	20
17.2	Tiempo de Respuesta . . . . .	20
17.3	Consumo de CPU . . . . .	20
17.4	Latencia Emparejamiento . . . . .	21

17.5 Manejo de transacciones . . . . .	21
<b>18 Restricciones de diseño</b>	<b>22</b>
18.1 Lenguajes de Programación y Frameworks . . . . .	22
18.2 Herramientas CASE (Análisis y Diseño) . . . . .	23
18.3 Restricciones de Hardware y Software . . . . .	23
18.4 Diseño de la Arquitectura Final (Microservicios) . . . . .	23
18.5 Modelo de Bases de Datos . . . . .	24
18.6 Requerimiento de Seguridad . . . . .	24
18.7 Sustitución de Hardware . . . . .	24
18.8 Sanity Check . . . . .	24
18.9 Reconfiguración . . . . .	24
18.10Protección de Datos . . . . .	25
18.11 Control de Acceso . . . . .	25
18.12Gestión de Errores y Excepciones . . . . .	26
<b>19 Atributos de calidad de software</b>	<b>26</b>
19.1 Adaptabilidad . . . . .	26
19.2 Disponibilidad . . . . .	26
19.3 Mantenibilidad . . . . .	27
19.4 Portabilidad . . . . .	27
19.5 Compatibilidad con navegadores y sistemas operativos . . . . .	28
19.6 Confiabilidad . . . . .	28
19.7 Capacidad de Prueba . . . . .	28
19.8 Usabilidad . . . . .	29
<b>20 Requerimientos de la base de datos</b>	<b>29</b>
20.1 Integridad . . . . .	29

20.2 Escalabilidad . . . . .	29
20.3 Seguridad y Acceso . . . . .	30
20.4 Backup y Recuperación . . . . .	30
<b>21 Proceso de ingeniería de requerimientos</b>	<b>31</b>
<b>22 Proceso de verificación</b>	<b>33</b>

## Índice de figuras

## Índice de cuadros

1	Glosario de Términos . . . . .	6
2	Stack Tecnológico y Especificaciones . . . . .	8
3	Historias de Usuario . . . . .	9
4	Perfil de Usuario - Estudiante Freelance . . . . .	12
5	Perfil de Usuario - Contratista . . . . .	12
6	Elemento de Dominio - Estudiante (E01) . . . . .	13
7	Elemento de Dominio - Contratista (E02) . . . . .	14
8	Elemento de Dominio - Proyecto (E03) . . . . .	14
9	Requisitos Funcionales y No Funcionales . . . . .	15
10	Requerimiento No Funcional - Concurrencia (RNF-001) . . . . .	20
11	Requerimiento No Funcional - Tiempo de Respuesta (RNF-002) . . . . .	20
12	Requerimiento No Funcional - Consumo de Recursos (RNF-003) . . . . .	21
13	Requerimiento No Funcional - Tiempo de Matchmaking (RNF-004) . . . . .	21
14	Requerimiento No Funcional - Manejo de Transacciones (RNF-005) . . . . .	22
15	Requerimiento No Funcional - Sustitución de Hardware (RNF-006) . . . . .	24
16	Requerimiento No Funcional - Sanity Check (RNF-007) . . . . .	25
17	Requerimiento No Funcional - Reconfiguración (RNF-010) . . . . .	25
18	Requerimiento No Funcional - Protección de Datos (RNF-011) . . . . .	25
19	Requerimiento No Funcional - Control de Acceso (RNF-012) . . . . .	26
20	Requerimiento No Funcional - Gestión de Errores (RNF-013) . . . . .	26
21	Requerimiento No Funcional - Adaptabilidad (RNF-015) . . . . .	27
22	Requerimiento No Funcional - Disponibilidad (RNF-016) . . . . .	27
23	Requerimiento No Funcional - Mantenibilidad (RNF-018) . . . . .	27
24	Requerimiento No Funcional - Portabilidad (RNF-019) . . . . .	28

25	Requerimiento No Funcional - Compatibilidad (RNF-020) . . . . .	28
26	Requerimiento No Funcional - Confiabilidad (RNF-021) . . . . .	28
27	Requerimiento No Funcional - Capacidad de Prueba (RNF-022) . . . . .	29
28	Requerimiento No Funcional - Usabilidad (RNF-023) . . . . .	29
29	Requerimiento No Funcional - Integridad de Base de Datos (OR-001) . . . . .	30
30	Requerimiento No Funcional - Escalabilidad de Base de Datos (OR-002) . . . . .	30
31	Requerimiento No Funcional - Seguridad y Acceso de Base de Datos (OR-003) . . . . .	30
32	Requerimiento No Funcional - Backup y Recuperación de Base de Datos (OR-004) . . . . .	31

## **1. Introducción**

## **2. Propósito**

El propósito de este proyecto es diseñar e implementar una plataforma web freelance que transforme la manera en que los estudiantes universitarios acceden a oportunidades laborales tempranas, para esto es necesario entender las necesidades de dos grupos principales de usuarios: estudiantes universitarios que realicen proyectos con el fin de enriquecer su perfil y prepararse para un ambiente laboral y contratistas que requieran individuos con capacitación técnica para realizar proyectos o tareas particulares. En este documento se detallan las principales funcionalidades con las que contará la aplicación, como contribuyen al sistema general y cómo estas resuelven las necesidades de los usuarios.

## **3. Alcance**

El producto de software descrito en este documento es una plataforma web freelance para estudiantes universitarios, cuyo propósito es facilitar el acceso a oportunidades laborales tempranas mediante un sistema de emparejamiento inteligente con contratistas. Esta aplicación incluirá funcionalidades como registro y verificación de identidad, creación de perfiles personalizables con habilidades y portafolios, publicación y búsqueda de proyectos, emparejamiento automatizado basado en inteligencia artificial, y un sistema de evaluación mutua al finalizar los proyectos.

La plataforma será de uso exclusivamente web (responsive) y estará orientada a estudiantes sin experiencia previa, permitiéndoles construir un portafolio profesional, mejorar su empleabilidad y obtener ingresos a través de colaboraciones prácticas y flexibles. Su desarrollo busca contribuir a la reducción del desempleo juvenil en contextos como el latinoamericano, donde las barreras de entrada al mercado laboral son especialmente altas.

Dentro del alcance de este proyecto no se incluye el desarrollo de modelos de inteligencia artificial, solo probar, analizar y comparar modelos ya existentes con el fin de escoger el que mejor se adapte a las necesidades de este producto. Tampoco se incluye la especificación de un modelo de negocio asociado al proyecto, solo se describe como un producto de software y como proyecto académico.

## 4. Definiciones, acrónimos y definiciones

Cuadro 1: Glosario de Términos

<b>Término / Acrónimo</b>	<b>Descripción</b>
Freelancing	Modalidad de empleo donde se ofrecen servicios sin relación contractual permanente.
Plataforma web	Sistema accesible por internet donde interactúan estudiantes y contratistas.
Contratista	Persona o entidad que publica proyectos para que los estudiantes puedan postularse.
Emparejamiento	Proceso que vincula automáticamente a estudiantes con proyectos según afinidad de perfiles.
IA	Tecnología que simula el pensamiento humano, usada para emparejamiento en la plataforma.
LLM	Modelo de lenguaje natural utilizado para comprender perfiles.
MVP	Versión inicial funcional de la plataforma con las funcionalidades mínimas necesarias.
UI	Interfaz gráfica mediante la cual los usuarios interactúan con la aplicación.
Scrum / Kanban / Scrumban	Marcos de trabajo usados para el desarrollo iterativo del software y gestión de tareas.
Vectores	Representaciones numéricas multidimensionales utilizadas para codificar información semántica.
Embeddings	Técnica de representación que convierte datos (texto, imágenes) en vectores numéricos para procesamiento.
Microservicios	Arquitectura de software donde una aplicación se compone de servicios pequeños e independientes.
Microfrontends	Patrón arquitectónico que extiende los microservicios al frontend, dividiendo la UI en features independientes.

## 5. Interfaces con el sistema

Aunque la plataforma freelance se desarrolla como un producto nuevo, incluirá una interfaz externa crítica con un servicio de inteligencia artificial de terceros que realizará el emparejamiento entre hojas

de vida de estudiantes y proyectos. Esta comunicación se realizará mediante llamadas a una API externa (por ejemplo, un modelo de lenguaje como GPT o un motor de recomendación desplegado en la nube), a través de protocolos seguros (HTTPS) y autenticación basada en API keys o tokens OAuth. Esta integración permitirá analizar descripciones de proyectos y perfiles de usuarios para generar recomendaciones semánticas precisas.

## **6. Interfaces con el hardware**

La plataforma web freelance será desplegada en un entorno privado, accedida por los usuarios a través de navegadores web en sus dispositivos personales (PC, laptops o smartphones). Aunque el sistema no interactúa directamente con hardware especializado, sí establece comunicación entre componentes distribuidos en distintos entornos.



## 7. Interfaces con el software

Cuadro 2: Stack Tecnológico y Especificaciones

Componente	Versión	Descripción	Fuente
PostgreSQL	15.3	Sistema gestor de bases de datos relacional. Robusto, escalable y compatible con arquitecturas modernas.	postgresql.org
pgAdmin	7.x (estable)	Herramienta de administración y monitoreo para la base de datos PostgreSQL. Facilita el monitoreo visual y la gestión.	pgadmin.org
JDBC	4.3	Conector para que el backend (Java/Spring Boot) acceda a la base de datos. Estándar en aplicaciones Java, garantiza compatibilidad.	jdbc.postgresql.org
Microsoft Windows	11 (64 bits)	Sistema operativo utilizado en entornos locales de desarrollo. Entorno común con alta compatibilidad de herramientas.	microsoft.com
Angular	16+	Framework para desarrollo del frontend. Permite crear aplicaciones web modernas y escalables.	angular.io
Spring Boot	3.x	Framework para desarrollo del backend en Java. Simplifica la creación de aplicaciones empresariales.	spring.io
Module Federation	Latest	Tecnología para microfrontends que permite composición de aplicaciones frontend independientes.	webpack.js.org
Docker	Latest	Plataforma de contenerización para empaquetar y distribuir aplicaciones de forma consistente.	docker.com
Kubernetes (K8s)	1.28+	Sistema de orquestación de contenedores para despliegue escalable y gestión de microservicios.	kubernetes.io
GitHub	-	Plataforma de hosting para control de versiones y colaboración en desarrollo de software.	github.com
GitHub Actions	-	Sistema de CI/CD integrado en GitHub para automatización de builds, tests y despliegues.	docs.github.com

## 8. Interfaces de comunicación

La plataforma web freelance se implementará como una aplicación distribuida que opera sobre entornos en red. Todas las comunicaciones entre sus componentes se realizarán utilizando el protocolo TCP/IP, el cual está definido en el RFC 793. Este protocolo ha sido seleccionado por ser confiable, orientado a conexión y ampliamente adoptado en aplicaciones cliente-servidor modernas, lo cual garantiza la entrega ordenada y sin pérdidas de los paquetes de datos.

La interacción entre el frontend (Angular), el backend (Spring Boot) y la base de datos (PostgreSQL) se llevará a cabo a través de una red local (LAN) utilizando HTTP/HTTPS sobre TCP/IP.

. La aplicación está diseñada para ser operada en dos diferentes modos: freelancer, contratista, los dos siendo modos utilizados por los usuarios finales de la aplicación, se diferencian en las funcionalidades a las que se tiene acceso y características como la información que conforma cada tipo de perfil.

Se desea tener una disponibilidad activa 24/7, permitiendo el acceso continuo de estudiantes y contratistas desde cualquier ubicación con conexión a internet. No obstante, se establecerán ventanas programadas de mantenimiento fuera del horario pico de uso (por ejemplo, entre las 00:00 y 02:00 AM hora local), durante las cuales el sistema podrá estar temporalmente inactivo para tareas de actualización, despliegue de nuevas versiones o mantenimiento correctivo.

## 9. Funciones del producto

Para una visión general de las funcionalidades del sistema se usa el formato de historias de usuario (HU) que se recopilan en la siguiente tabla.

Cuadro 3: Historias de Usuario

ID	Nombre	Rol	Como quiero	Para	Prioridad
HU001	Inicio sesión	Estudiante/Contratista	Iniciar sesión con credenciales (correo y contraseña)	Acceder a mi cuenta y funcionalidades	Media
HU002	Registro estudiante	Estudiante freelance	Registrarme con datos personales, académicos y de contacto	Crear mi perfil y postularme a proyectos	Media

*Continúa en la siguiente página*

Cuadro 3: Historias de Usuario (continuación)

ID	Nombre	Rol	Como quiero	Para	Prioridad
HU003	Registro contratista	Contratista	Registrarme con información básica y validación	Publicar proyectos y contratar estudiantes	Media
HU004	Emparejamiento	Contratista	Recibir sugerencias automáticas de perfiles de estudiantes	Acelerar selección y encontrar candidatos relevantes	Alta
HU005	Perfil de estudiante	Estudiante freelance	Crear/editar perfil con habilidades, estudios, experiencia y subir hoja de vida	Presentarme profesionalmente ante contratistas	Alta
HU006	Búsqueda de proyectos	Estudiante freelance	Buscar proyectos filtrando por categoría, duración, pago	Encontrar oportunidades que se ajusten a mis capacidades	Media
HU007	Búsqueda de perfiles	Contratista	Buscar estudiantes filtrando por habilidades, estudios	Identificar rápidamente candidatos relevantes	Media
HU008	Gestión de proyectos	Contratista	Crear, editar y eliminar proyectos desde mi panel	Mantener actualizada mi oferta de trabajo	Media
HU009	Comunicación con contratista	Estudiante freelance	Enviar mensajes privados al contratista	Aclarar dudas y coordinar tareas durante el proyecto	Media
HU010	Comunicación con estudiante	Contratista	Enviar mensajes directos a estudiantes postulados	Coordinar avances y resolver dudas en tiempo real	Media
HU011	Gestión de postulantes	Contratista	Ver, comparar, aceptar o rechazar postulaciones	Seleccionar al candidato más adecuado	Media
HU012	Postulación	Estudiante freelance	Postularme con mi perfil y mensaje de presentación	Ofrecer servicios y aumentar chances de contratación	Media

*Continúa en la siguiente página*

Cuadro 3: Historias de Usuario (continuación)

ID	Nombre	Rol	Como quiero	Para	Prioridad
HU013	Calificación de estudiante	Contratista	Calificar el trabajo del estudiante al completar proyecto	Ayudar a otros y reconocer desempeño	Baja
HU014	Calificación de contratista	Estudiante freelance	Calificar al contratista al finalizar proyecto	Compartir experiencia y promover contratistas confiables	Baja

De las funciones se destacan como principales:

- **HU004 Emparejamiento:** Esta funcionalidad permite a los contratistas realizar búsquedas intuitivas y rápidas. Implica un proceso técnico de selección del modelo a implementar lo que tomará atención y prioridad en fases de desarrollo.
- **HU005 Perfil de estudiante:** La aplicación tiene un enfoque en dar oportunidades a los estudiantes para generar experiencia laboral, por lo que es prioritario dar a estos usuarios la posibilidad de formar un perfil profesional en el que además se pueda visualizar un progreso y su aporte en previos trabajos.
- **HU008 Gestión de proyectos:** El principal elemento de interacción entre las partes involucradas en la aplicación son los proyectos que postulan los contratistas por lo que facilitar su manejo de alta importancia, mantener un buen flujo a la hora de administrar proyectos permitirá a los usuarios aprovechar todas las funcionalidades que se ofrecen en la plataforma.

## 10. Características del Usuario

La plataforma está dirigida a dos clases principales de usuarios: estudiantes freelancers y contratistas. Ambos grupos presentan características, necesidades y niveles de interacción distintos, por lo que el sistema ha sido diseñado para responder adecuadamente a sus contextos de uso, niveles de experiencia técnica y privilegios dentro de la aplicación.

## 10.1. Estudiante - Freelancer

Cuadro 4: Perfil de Usuario - Estudiante Freelance

Característica	Descripción
Rol	Usuario que se registra para ofrecer servicios en modalidad freelance.
Frecuencia de uso	Alta. Acceden con regularidad para buscar proyectos, aplicar y gestionar tareas.
Nivel de experiencia	Variable. Generalmente usuarios con habilidades técnicas, pero con falta de experiencia laboral, pero familiarizados con plataformas web y móviles.
Nivel de estudios	Estudiantes de pregrado, típicamente entre 18 y 26 años, de diversas áreas académicas.
Privilegios en la plataforma	Crear y editar su perfil, buscar proyectos, postularse, gestionar entregas y recibir calificaciones.
Requisitos funcionales clave	Registro, carga de CV/portafolio, visualización de proyectos, chat.

## 10.2. Contratista

Cuadro 5: Perfil de Usuario - Contratista

Característica	Descripción
Rol	Usuario que publica proyectos y contrata freelancers.
Frecuencia de uso	Media. Acceso cuando necesitan talento joven para proyectos específicos.
Nivel de experiencia	Bajo. Sin conocimientos técnicos avanzados, pero con capacidad para interfaces simples.
Nivel educativo/profesional	Profesionales independientes o de pequeñas empresas. Cargos administrativos/selección.
Privilegios en la plataforma	Publicar proyectos, revisar postulaciones, seleccionar candidatos, calificar, dar feedback.
Requisitos funcionales clave	Registro, creación de proyectos, filtros búsqueda, revisión perfiles, evaluación, emparejamiento.

## 11. Restricciones

- **Idioma del sistema:** La aplicación estará disponible inicialmente en español, dado su enfoque en el contexto latinoamericano.
- **Plataforma:** El desarrollo estará limitado a una plataforma web responsive, sin incluir aplicaciones móviles nativas para iOS o Android en esta fase inicial.
- **Plazo de desarrollo:** El proyecto debe completarse en un plazo máximo de 6 meses, divididos en fases claras de análisis, desarrollo y pruebas.
- **Procesamiento de pagos:** La plataforma no incluirá funcionalidades de procesamiento de pagos o transacciones financieras en esta versión.
- **Lenguajes de programación:** El sistema usará Java 17 para el backend con el framework Spring Boot, y TypeScript con Angular para el frontend.
- **Aspectos relacionados al software:** Lenguajes de programación, licencias, APIs permitidas.

## 12. Modelo del dominio

Cuadro 6: Elemento de Dominio - Estudiante (E01)

<b>ID</b>	E01
<b>Elemento</b>	Estudiante
<b>Descripción</b>	Representa al usuario que se registra en la plataforma para ofrecer servicios como freelancer.
<b>Objetivo</b>	Gestionar y almacenar información relevante sobre el perfil académico del estudiante.

Atributos		
Nombre	Descripción	Tipo
id	Identificador único	UUID
nombre	Nombre y apellido del estudiante	String
correo	Correo electrónico para inicio de sesión	String
institución	Institución educativa del estudiante	String
calificación	Calificación promedio recibida	Float

Cuadro 7: Elemento de Dominio - Contratista (E02)

<b>ID</b>	E02
<b>Elemento</b>	Contratista
<b>Descripción</b>	Usuario que publica proyectos dentro de la plataforma y contrata estudiantes para ejecutarlos.
<b>Objetivo</b>	Representar al usuario que requiere talento joven para desarrollar tareas específicas.

Atributos		
Nombre	Descripción	Tipo
id	Identificador único	UUID
nombre	Nombre del contratista o organización	String
correo	Correo de contacto e inicio de sesión	String
calificación	Calificación promedio recibida	Float

Cuadro 8: Elemento de Dominio - Proyecto (E03)

<b>ID</b>	E03
<b>Elemento</b>	Proyecto
<b>Descripción</b>	Representa una propuesta de trabajo publicada por un contratista para ser ejecutada por un estudiante.
<b>Objetivo</b>	Centralizar la información sobre una tarea concreta, su descripción y estado.

Atributos		
Nombre	Descripción	Tipo
id	Identificador único	UUID
titulo	Nombre del proyecto	String
descripción	Detalle del proyecto	String
estado	Estado actual (abierto, en ejecución, finalizado)	Enum
contratista	Referencia al contratista asociado	FK (contratista_id)
estudiante	Referencia al estudiante asociado	FK (estudiante_id)

## 13. Suposiciones y dependencias

- Los principales usuarios estarán dispuestos a adoptar una nueva plataforma digital para gestionar oportunidades laborales flexibles.
- La plataforma será accedida principalmente desde navegadores web en dispositivos móviles y computadoras personales con conexión a internet estable.
- Existirá suficiente interés por parte de contratistas (empresas y profesionales independientes) en publicar proyectos adecuados para estudiantes.
- Los modelos de IA para emparejamiento podrán utilizar efectivamente los datos de perfiles y proyectos disponibles.

## 14. Requerimientos Específicos

Cuadro 9: Requisitos Funcionales y No Funcionales

ID	HU Relacio- nada	Descripción	Razón	Criterio de Medición
RE001	HU001	El sistema debe permitir a un estudiante crear una cuenta ingresando su nombre completo, correo y contraseña.	Permite habilitar el acceso del usuario a la plataforma.	Se verifica mediante prueba funcional: creación de cuenta exitosa con datos válidos.
RE002	HU002	El sistema debe permitir a un contratista crear una cuenta ingresando nombre de empresa, correo y contraseña.	Habilita a los contratistas para publicar proyectos en la plataforma.	Se verifica si la cuenta queda registrada correctamente en la base de datos.

*Continúa en la siguiente página*



Cuadro 9: Requisitos Funcionales y No Funcionales (continuación)

ID	HU Relacionada	Descripción	Razón	Criterio de Medición
RE003	HU005	El sistema debe enviar datos del perfil del estudiante y descripción del proyecto a una API externa de IA para obtener sugerencias de emparejamiento.	Mejora la precisión en la selección de candidatos mediante IA.	Se verifica que el backend realice la petición a la API y reciba sugerencias válidas.
RE004	HU007	El sistema debe permitir la búsqueda de proyectos aplicando filtros por categoría, duración o tipo de proyecto.	Facilita encontrar proyectos adecuados al perfil e intereses del estudiante.	Se verifica que los filtros actúen correctamente en la interfaz de usuario.
RE005	HU013	El estudiante debe poder postularse a un proyecto enviando su perfil y mensaje de presentación.	Permite que los estudiantes accedan a oportunidades laborales.	Se valida que el botón de postulación funcione y que la información se guarde correctamente.
RE006	HU009	El contratista debe poder crear, editar y eliminar proyectos desde su panel.	Permite administrar la oferta de trabajo disponible.	Se verifica mediante prueba CRUD sobre proyectos.
RE007	HU010	La plataforma debe permitir mensajes privados entre contratista y estudiante dentro del sistema.	Mejora la coordinación durante la ejecución del proyecto.	Se valida mediante prueba de envío y recepción de mensajes entre usuarios.
RE008	HU014	El contratista debe poder calificar al estudiante al finalizar el proyecto.	Fomenta la responsabilidad y permite a futuros contratistas evaluar desempeño previo.	Se verifica que la calificación sea guardada y asociada correctamente al estudiante.

*Continúa en la siguiente página*

Cuadro 9: Requisitos Funcionales y No Funcionales (continuación)

ID	HU Relacio- nada	Descripción	Razón	Criterio de Medición
RE009	—	El sistema debe bloquear temporalmente el acceso tras 5 intentos fallidos de inicio de sesión.	Previene intentos de acceso no autorizados mediante fuerza bruta.	Se prueba que tras 5 intentos el sistema impida acceso por un periodo (ej. 5 minutos).
RE010	—	Todos los formularios deben validar los campos obligatorios antes de enviarlos.	Garantiza la integridad de los datos ingresados por el usuario.	Se verifica que el botón de envío esté deshabilitado hasta completar campos requeridos.
RE011	—	El sistema debe realizar respaldos automáticos de la base de datos diariamente a las 3:00 AM.	Asegura la recuperación ante fallos o pérdida de información.	Se verifica la existencia de archivos de respaldo diarios con sello de fecha/hora.
RE012	—	El sistema debe cifrar las contraseñas de los usuarios usando hashing seguro (ej. bcrypt).	Protege credenciales frente a accesos no autorizados o brechas de seguridad.	Se inspecciona en la base de datos que las contraseñas no estén almacenadas en texto plano.
RE013	—	El sistema debe estar disponible el 99 % del tiempo mensual, excluyendo mantenimientos.	Requisito de confiabilidad mínimo para una plataforma activa 24/7.	Se mide uptime con herramientas de monitoreo como UptimeRobot o Pingdom.
RE014	—	El sistema debe permitir registrar logs de eventos importantes como login, postulación y calificación.	Permite trazabilidad y auditoría de acciones clave dentro de la plataforma.	Se verifica la existencia de registros generados automáticamente ante eventos definidos.

*Continúa en la siguiente página*

Cuadro 9: Requisitos Funcionales y No Funcionales (continuación)

ID	HU Relacio- nada	Descripción	Razón	Criterio de Medición
RE015	—	El sistema debe responder a solicitudes del usuario final en menos de 3 segundos (90 % de las veces).	Mejora la experiencia del usuario, evita frustración por lentitud.	Se mide tiempo de respuesta con pruebas de carga (ej. JMeter, Postman).
RE016	—	El sistema debe ser accesible desde navegadores modernos en versiones actualizadas.	Aumenta compatibilidad y facilita el acceso multiplataforma.	Se comprueba con pruebas funcionales en Chrome, Firefox, Edge y Safari.
RE017	—	La plataforma debe estar disponible desde dispositivos móviles sin instalar aplicaciones.	Facilita el acceso flexible al sistema desde cualquier lugar.	Se verifica que la interfaz sea responsive y funcional en pantallas móviles.
RE018	—	El sistema debe permitir cambiar la contraseña desde el perfil de usuario autenticado.	Proporciona al usuario control sobre la seguridad de su cuenta.	Se prueba el flujo de actualización de contraseña y su efecto inmediato.

## 15. Interfaces con el usuario

La plataforma utiliza una interfaz gráfica de usuario como principal medio de comunicación, el llenado de formularios y la interacción en general se realizará con periféricos como el teclado y ratón en computadoras personales y directamente con la pantalla en el caso de dispositivos móviles. Al ser una página web también es requerida la conexión a internet lo que implica el uso de una tarjeta de red de parte del usuario.

## 16. Características del producto de software

**Gestión de perfiles:** Esta funcionalidad permite a los usuarios registrar, ver y editar sus perfiles, ya sea para usuarios freelance o contratistas. Está relacionada con las historias de usuario HU001, HU002, HU003 y HU005, por lo que incluye los registros de perfiles, inicio de sesiones y en general la base fundamental para el flujo de datos por toda la aplicación.

**Búsqueda de freelancer:** Este módulo incluye los requerimientos orientados a la búsqueda de perfiles de estudiantes, por lo que se incluyen las búsquedas manuales con filtros y el emparejamiento utilizando un modelo de inteligencia artificial. Este flujo es de gran importancia para los usuarios contratistas pues facilita encontrar un candidato adecuado que de otra forma tomaría tiempo y recursos para encontrar.

**Manejo de oferta:** Las ofertas de trabajo representan la conexión de intereses de las partes involucradas en la aplicación. Varios requerimientos constituyen el buen funcionamiento y flujo de las tareas, desde su creación y manejo por parte del contratista, su búsqueda y filtrado por parte del estudiante hasta su culminación y retroalimentación.

## 17. Requerimientos de desempeño

A continuación, se definen los requerimientos no funcionales (RNF) relacionados con el rendimiento del sistema, divididos en:

Cada requerimiento incluye:

- **ID:** Identificador único.
- **Nombre:** Descripción clara.
- **Característica:** Detalle técnico.
- **Razón:** Justificación del requerimiento.
- **Criterio de medición:** Método para validar el cumplimiento.
- **Prioridad:** Escala del 1 (baja) al 10 (crítica).

### 17.1. Concurrency

El sistema debe gestionar al menos 5000 usuarios concurrentes, y escalar hasta 80000 usuarios concurrentes sin degradación en el rendimiento.

Cuadro 10: Requerimiento No Funcional - Concurrency (RNF-001)

<b>ID</b>	RNF-001
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Concurrency
<b>Característica</b>	El sistema debe gestionar al menos 5000 usuarios concurrentes, y escalar hasta 80000 usuarios concurrentes sin degradación en el rendimiento.
<b>Razón</b>	Hay que asegurar que el sistema puede soportar un alto volumen de usuarios simultáneos sin comprometer el rendimiento.
<b>Criterio de Medición</b>	Pruebas de carga que simulen hasta 80000 usuarios concurrentes, monitoreando la degradación del rendimiento.
<b>Prioridad</b>	8

### 17.2. Time to Response

Todas las páginas de la aplicación deben cargar en menos de 5 segundos en condiciones de carga normal.

Cuadro 11: Requerimiento No Funcional - Time to Response (RNF-002)

<b>ID</b>	RNF-002
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Time to response
<b>Característica</b>	Todas las páginas de la aplicación deben cargar en menos de 5 segundos en condiciones de carga normal.
<b>Razón</b>	Mejorar la experiencia del usuario al reducir los tiempos de espera.
<b>Criterio de Medición</b>	Pruebas de velocidad en condiciones de carga normal con menos de 5 segundos de respuesta en el 95 % de los casos.
<b>Prioridad</b>	5

### 17.3. CPU Consumption

En carga máxima, el uso de CPU no debe superar el 80

Cuadro 12: Requerimiento No Funcional - Consumo de Recursos (RNF-003)

<b>ID</b>	RNF-003
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Consumo de recursos
<b>Característica</b>	<ul style="list-style-type: none"> <li>■ Consumo de CPU en servidores no debe superar el 80 % en condiciones de carga máxima.</li> <li>■ Uso de memoria en servidores no debe superar el 75 % de la capacidad total en condiciones de carga máxima.</li> </ul>
<b>Razón</b>	Optimizar el uso de recursos del sistema para evitar el sobreuso que pueda llevar a fallos.
<b>Criterio de Medición</b>	Monitoreo de los recursos del sistema durante pruebas de carga máxima.
<b>Prioridad</b>	7

#### 17.4. Latencia Emparejamiento

El sistema debe devolver sugerencias de matchmaking en ¡5 segundos.

Cuadro 13: Requerimiento No Funcional - Tiempo de Matchmaking (RNF-004)

<b>ID</b>	RNF-004
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Tiempo de respuesta de matchmaking
<b>Característica</b>	El sistema debe devolver sugerencias de matchmaking en ¡5 segundos.
<b>Razón</b>	Optimizar experiencia de usuario y tiempos de respuesta.
<b>Criterio de Medición</b>	Medición del tiempo de respuesta de la API de IA con Postman o k6.
<b>Prioridad</b>	7

#### 17.5. Manejo de transacciones

El sistema debe poder procesar hasta 1000 transacciones por minuto.

Cuadro 14: Requerimiento No Funcional - Manejo de Transacciones (RNF-005)

<b>ID</b>	RNF-005
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Manejo de transacciones
<b>Característica</b>	El sistema debe poder procesar hasta 1000 transacciones por minuto.
<b>Razón</b>	Garantizar que el sistema pueda manejar grandes volúmenes de transacciones en periodos de alta demanda.
<b>Criterio de Medición</b>	Pruebas de estrés con simulación de hasta 1000 transacciones de reserva por minuto.
<b>Prioridad</b>	6

## 18. Restricciones de diseño

Estas limitaciones consideran factores como el lenguaje de programación, las herramientas de análisis y diseño, los requerimientos de hardware y software, y la arquitectura del sistema. A continuación, se presentan los principales tópicos y restricciones relevantes para el diseño de la aplicación:

### 18.1. Lenguajes de Programación y Frameworks

El proyecto está restringido al uso de tecnologías específicas para el desarrollo del frontend y backend, seleccionadas por su robustez, escalabilidad y alineación con los objetivos del sistema:

#### ■ Frontend:

- **Angular 15:** Framework basado en TypeScript que proporciona una arquitectura modular y reactiva. Su uso impone restricciones en la estructura del código, como la adopción de componentes, servicios y módulos.
- **TypeScript:** Lenguaje tipado que reduce errores en tiempo de compilación y mejora la mantenibilidad del código. Esto limita la flexibilidad en la implementación de funcionalidades dinámicas, pero aumenta la calidad del software.

#### ■ Backend:

- **Spring Boot 3:** Framework basado en Java 17, que sigue el paradigma de desarrollo empresarial. Restringe la arquitectura a patrones como MVC (Modelo-Vista-Controlador) y requiere la implementación de APIs RESTful.

## 18.2. Herramientas CASE (Análisis y Diseño)

Para el análisis y diseño del sistema, se utilizan herramientas específicas que imponen ciertas limitaciones:

- **Modelado de Arquitectura:**

- **Diagramas C4 y UML:** Herramientas como Lucidchart o Visual Paradigm se emplean para diseñar la arquitectura del sistema.
- **IEEE 830-1998:** Estándar para la documentación de requisitos, que exige un formato específico y la inclusión de matrices de trazabilidad.

## 18.3. Restricciones de Hardware y Software

El paradigma de microservicios y los lenguajes seleccionados imponen restricciones técnicas en los entornos de desarrollo, pruebas y producción:

- **Hardware:**

- **Entorno de Desarrollo:** Requiere máquinas con al menos 4GB de RAM y procesadores multinúcleo.
- **Producción:** La infraestructura debe soportar escalabilidad horizontal, lo que limita las opciones a servicios compatibles con contenedores.

- **Software:**

- **DevOps:** Docker para contenerización y GitHub CI/CD para integración continua.
- **Bases de Datos:** PostgreSQL (sistema relacional) y MongoDB para almacenamiento de conversaciones.

## 18.4. Diseño de la Arquitectura Final (Microservicios)

La adopción de una arquitectura de microservicios introduce restricciones significativas:

- **Desacoplamiento:** Cada servicio debe ser independiente, obligando al uso de APIs REST o mensajería (RabbitMQ).
- **Gestión de Dependencias:** Los microservicios deben evitar compartir librerías o bases de datos.



## 18.5. Modelo de Bases de Datos

El diseño de la base de datos sigue restricciones técnicas y normativas:

- **Esquema Relacional:** PostgreSQL limita el modelado a tablas, relaciones y claves foráneas.
- **Normalización:** Hasta la tercera forma normal (3FN) para evitar redundancias.
- **Seguridad:** Encriptación con algoritmos como bcrypt y roles de acceso definidos.

## 18.6. Requerimiento de Seguridad

## 18.7. Sustitución de Hardware

El sistema permite la actualización y sustitución de hardware de manera transparente.

Cuadro 15: Requerimiento No Funcional - Sustitución de Hardware (RNF-006)

<b>ID</b>	RNF-006
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Sustitución de hardware
<b>Característica</b>	El sistema permite la actualización y sustitución de hardware de manera transparente.
<b>Razón</b>	Facilitar el mantenimiento y la escalabilidad del sistema sin interrumpir las operaciones.
<b>Criterio de Medición</b>	Monitoreo de los recursos del sistema durante pruebas de carga máxima.
<b>Prioridad</b>	5

## 18.8. Sanity Check

El sistema debe implementar verificaciones automáticas de la validez de las entradas y salidas de los componentes críticos del sistema.

## 18.9. Reconfiguración

El sistema debe tener la capacidad de reasignar recursos disponibles y funcionales para mantener la operación, incluso en presencia de fallos.

Cuadro 16: Requerimiento No Funcional - Sanity Check (RNF-007)

<b>ID</b>	RNF-007
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Sanity Check
<b>Característica</b>	El sistema debe implementar verificaciones automáticas de la validez de las entradas y salidas de los componentes críticos del sistema.
<b>Razón</b>	Prevenir fallos en el sistema debido a entradas o salidas no válidas.
<b>Criterio de Medición</b>	Monitoreo de los componentes críticos y generación de alertas en caso de entradas o salidas no válidas.
<b>Prioridad</b>	8

Cuadro 17: Requerimiento No Funcional - Reconfiguración (RNF-010)

<b>ID</b>	RNF-010
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Reconfiguración
<b>Característica</b>	El sistema debe tener la capacidad de reasignar recursos disponibles y funcionales para mantener la operación, incluso en presencia de fallos.
<b>Razón</b>	Aumentar la resiliencia del sistema frente a fallos de hardware o software.
<b>Criterio de Medición</b>	Pruebas de reconfiguración automática ante fallos simulados, asegurando la continuidad de las operaciones.
<b>Prioridad</b>	7

## 18.10. Protección de Datos

El sistema debe garantizar la confidencialidad de la información de sus usuarios.

Cuadro 18: Requerimiento No Funcional - Protección de Datos (RNF-011)

<b>ID</b>	RNF-011
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Protección de datos
<b>Característica</b>	El sistema debe garantizar la confidencialidad de la información de sus usuarios.
<b>Razón</b>	Proteger la privacidad de los usuarios y cumplir con regulaciones de protección de datos.
<b>Criterio de Medición</b>	Pruebas de seguridad que verifiquen la protección y el cifrado de los datos de los usuarios.
<b>Prioridad</b>	9

## 18.11. Control de Acceso

El sistema debe implementar un control de acceso basado en roles (RBAC) para limitar el acceso a la información y las funciones del sistema según el rol del usuario.

Cuadro 19: Requerimiento No Funcional - Control de Acceso (RNF-012)

<b>ID</b>	RNF-012
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Control de acceso
<b>Característica</b>	El sistema debe implementar un control de acceso basado en roles (RBAC) para limitar el acceso a la información y las funciones del sistema según el rol del usuario.
<b>Razón</b>	Garantizar que solo los usuarios autorizados puedan acceder a determinadas funciones y datos.
<b>Criterio de Medición</b>	Pruebas de control de acceso en distintos niveles de roles, verificando que solo se permita acceso según el rol asignado.
<b>Prioridad</b>	7

## 18.12. Gestión de Errores y Excepciones

El sistema debe manejar de manera segura los errores y excepciones, evitando la exposición de información sensible a los usuarios.

Cuadro 20: Requerimiento No Funcional - Gestión de Errores (RNF-013)

<b>ID</b>	RNF-013
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Gestión de errores y excepciones
<b>Característica</b>	El sistema debe manejar de manera segura los errores y excepciones, evitando la exposición de información sensible a los usuarios.
<b>Razón</b>	Proteger la seguridad del sistema y la privacidad de los usuarios evitando la exposición de información técnica o sensible.
<b>Criterio de Medición</b>	Pruebas de manejo de errores en distintos escenarios, asegurando que no se exponga información sensible en los mensajes de error.
<b>Prioridad</b>	7

## 19. Atributos de calidad de software

### 19.1. Adaptabilidad

Permite la adición o modificación de funcionalidades sin impactar negativamente en el rendimiento.

### 19.2. Disponibilidad

El sistema debe estar disponible el 99.9% del tiempo durante horas de operación definidas (24/7), con un tiempo de inactividad máximo de 8.76 horas al año.

Cuadro 21: Requerimiento No Funcional - Adaptabilidad (RNF-015)

<b>ID</b>	RNF-015
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Adaptabilidad
<b>Característica</b>	El sistema permite la adición o modificación de funcionalidades sin impactar negativamente en el rendimiento.
<b>Razón</b>	Facilitar la evolución del sistema a medida que surjan nuevas necesidades sin afectar la experiencia del usuario o el rendimiento general.
<b>Criterio de Medición</b>	Pruebas de modificación de funcionalidades que verifiquen que no haya impacto significativo en el rendimiento del sistema.
<b>Prioridad</b>	6

Cuadro 22: Requerimiento No Funcional - Disponibilidad (RNF-016)

<b>ID</b>	RNF-016
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Disponibilidad
<b>Característica</b>	El sistema debe estar disponible el 99.9% del tiempo durante horas de operación definidas (24/7), con un tiempo de inactividad máximo de 8.76 horas al año.
<b>Razón</b>	Garantizar que el sistema esté operativo la mayor parte del tiempo, minimizando interrupciones del servicio.
<b>Criterio de Medición</b>	Monitoreo de la disponibilidad del sistema durante un periodo de prueba de un año, verificando que el tiempo de inactividad no exceda las 8.76 horas.
<b>Prioridad</b>	7

### 19.3. Mantenibilidad

El sistema debe diseñarse modularmente para facilitar el mantenimiento y las actualizaciones.

Cuadro 23: Requerimiento No Funcional - Mantenibilidad (RNF-018)

<b>ID</b>	RNF-018
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Mantenibilidad
<b>Característica</b>	El sistema debe diseñarse modularmente para facilitar el mantenimiento y las actualizaciones.
<b>Razón</b>	Facilitar el mantenimiento del sistema y permitir actualizaciones sin afectar la operación general.
<b>Criterio de Medición</b>	Revisión del diseño modular del sistema y pruebas de actualización que no afecten la operación del mismo.
<b>Prioridad</b>	4

### 19.4. Portabilidad

El sistema debe ser accesible desde múltiples dispositivos como dispositivos móviles, tabletas y computadores.

Cuadro 24: Requerimiento No Funcional - Portabilidad (RNF-019)

<b>ID</b>	RNF-019
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Portabilidad
<b>Característica</b>	El sistema debe ser accesible desde múltiples dispositivos como dispositivos móviles, tabletas y computadores.
<b>Razón</b>	Ampliar el alcance y la accesibilidad del sistema, permitiendo a los usuarios acceder desde diferentes dispositivos.
<b>Criterio de Medición</b>	Pruebas de compatibilidad multi-dispositivo.
<b>Prioridad</b>	5

### 19.5. Compatibilidad con navegadores y sistemas operativos

El sistema debe ser compatible con los principales navegadores web y sistemas operativos.

Cuadro 25: Requerimiento No Funcional - Compatibilidad (RNF-020)

<b>ID</b>	RNF-020
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Compatibilidad con navegadores y sistemas operativos
<b>Característica</b>	El sistema debe ser compatible con los principales navegadores web y sistemas operativos.
<b>Razón</b>	Garantizar que los usuarios puedan acceder al sistema sin problemas desde diferentes navegadores y plataformas.
<b>Criterio de Medición</b>	Pruebas de compatibilidad en distintos navegadores y sistemas operativos para asegurar el correcto funcionamiento del sistema.
<b>Prioridad</b>	5

### 19.6. Confiabilidad

El sistema debe garantizar la integridad y consistencia de los datos en todas las transacciones.

Cuadro 26: Requerimiento No Funcional - Confiabilidad (RNF-021)

<b>ID</b>	RNF-021
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Confiabilidad
<b>Característica</b>	El sistema debe garantizar la integridad y consistencia de los datos en todas las transacciones.
<b>Razón</b>	Hay que asegurar que no haya pérdida o corrupción de datos durante las transacciones del sistema.
<b>Criterio de Medición</b>	Pruebas de consistencia y validación de integridad de datos en distintas operaciones del sistema.
<b>Prioridad</b>	8

### 19.7. Capacidad de Prueba

El sistema debe diseñarse para facilitar las pruebas unitarias, de integración y de aceptación.

Cuadro 27: Requerimiento No Funcional - Capacidad de Prueba (RNF-022)

<b>ID</b>	RNF-022
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Capacidad de prueba
<b>Característica</b>	El sistema debe diseñarse para facilitar las pruebas unitarias, de integración y de aceptación.
<b>Razón</b>	Garantizar que el sistema pueda ser probado de manera eficiente en todas las fases de desarrollo.
<b>Criterio de Medición</b>	Pruebas de la facilidad de implementación de pruebas unitarias, de integración y de aceptación en el sistema.
<b>Prioridad</b>	5

## 19.8. Usabilidad

El sistema debe tener una interfaz de usuario intuitiva y fácil de usar, que minimice la necesidad de capacitación para los usuarios y administradores.

Cuadro 28: Requerimiento No Funcional - Usabilidad (RNF-023)

<b>ID</b>	RNF-023
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Usabilidad
<b>Característica</b>	El sistema debe tener una interfaz de usuario intuitiva y fácil de usar, que minimice la necesidad de capacitación para los usuarios y administradores.
<b>Razón</b>	Mejorar la experiencia del usuario y reducir el tiempo necesario para que los usuarios y administradores aprendan a usar el sistema.
<b>Criterio de Medición</b>	Pruebas de usabilidad y encuestas a usuarios y administradores para evaluar la facilidad de uso del sistema.
<b>Prioridad</b>	6

## 20. Requerimientos de la base de datos

### 20.1. Integridad

La base de datos debe garantizar la integridad referencial, asegurando que todas las relaciones entre tablas se mantengan consistentes.

### 20.2. Escalabilidad

La arquitectura de la base de datos debe ser escalable horizontal y verticalmente para manejar un aumento en la cantidad de usuarios y transacciones sin degradar el rendimiento.

Cuadro 29: Requerimiento No Funcional - Integridad de Base de Datos (OR-001)

<b>ID</b>	OR-001
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Integridad de Base de Datos
<b>Característica</b>	La base de datos debe garantizar la integridad referencial, asegurando que todas las relaciones entre tablas se mantengan consistentes.
<b>Razón</b>	Asegurar la consistencia de los datos y evitar errores que puedan comprometer la validez de la información almacenada.
<b>Criterio de Medición</b>	Revisión de las reglas de integridad referencial en el diseño de la base de datos y ejecución de pruebas de consistencia.
<b>Prioridad</b>	9

Cuadro 30: Requerimiento No Funcional - Escalabilidad de Base de Datos (OR-002)

<b>ID</b>	OR-002
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Escalabilidad de Base de Datos
<b>Característica</b>	La arquitectura de la base de datos debe ser escalable horizontal y verticalmente para manejar un aumento en la cantidad de usuarios y transacciones sin degradar el rendimiento.
<b>Razón</b>	Garantizar que el sistema pueda adaptarse a un crecimiento en la carga de trabajo sin afectar el rendimiento general.
<b>Criterio de Medición</b>	Pruebas de escalabilidad en entornos simulados con un crecimiento significativo en la cantidad de usuarios y transacciones.
<b>Prioridad</b>	5

### 20.3. Seguridad y Acceso

La base de datos debe implementar controles de acceso basado en roles (RBAC) para asegurar que los datos sensibles estén protegidos y solo accesibles por personal autorizado.

Cuadro 31: Requerimiento No Funcional - Seguridad y Acceso de Base de Datos (OR-003)

<b>ID</b>	OR-003
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Seguridad y Acceso de Base de Datos
<b>Característica</b>	La base de datos debe implementar controles de acceso basado en roles (RBAC) para asegurar que los datos sensibles estén protegidos y solo accesibles por personal autorizado.
<b>Razón</b>	Proteger la confidencialidad de la información y limitar el acceso a los datos a usuarios autorizados según su rol.
<b>Criterio de Medición</b>	Pruebas de los controles de acceso y auditorías de seguridad para verificar que los datos estén protegidos adecuadamente.
<b>Prioridad</b>	7

### 20.4. Backup y Recuperación

La base de datos debe contar con procedimientos automáticos de backup y recuperación de datos en caso de pérdida de datos.

Cuadro 32: Requerimiento No Funcional - Backup y Recuperación de Base de Datos (OR-004)

<b>ID</b>	OR-004
<b>Tipo</b>	No Funcional
<b>Nombre</b>	Backup y Recuperación de Base de Datos
<b>Característica</b>	La base de datos debe contar con procedimientos automáticos de backup y recuperación de datos en caso de pérdida de datos.
<b>Razón</b>	Asegurar la continuidad de las operaciones y la protección de la información frente a posibles fallos.
<b>Criterio de Medición</b>	Pruebas periódicas de backup y recuperación para asegurar que los procedimientos funcionan correctamente.
<b>Prioridad</b>	6

## 21. Proceso de ingeniería de requerimientos

El proceso de administración de requerimientos se implementará mediante un flujo de trabajo estructurado que garantice el control y seguimiento de todas las solicitudes, tanto nuevas como modificaciones a requerimientos existentes. Este mecanismo asegurará que cada cambio sea evaluado, implementado y validado de manera sistemática, manteniendo la alineación con los objetivos del proyecto y la satisfacción del cliente.

El ciclo comienza cuando los diferentes actores del proyecto ya sean clientes o miembros del equipo de desarrollo, identifican la necesidad de introducir un nuevo requerimiento o modificar uno existente. En esta fase inicial, el ingeniero de requerimientos asume la responsabilidad de documentar formalmente cada petición, registrando toda la información relevante en las historias de usuario correspondientes y en el repositorio central de requerimientos. Este registro debe incluir una descripción detallada de la necesidad, la justificación del cambio, la prioridad inicial asignada y la identificación de todas las partes interesadas que podrían verse afectadas por esta modificación.

Una vez registrada la solicitud, el supervisor técnico realiza un análisis exhaustivo para determinar la viabilidad de su implementación. Este examen considera múltiples factores críticos, incluyendo el impacto potencial en la arquitectura del sistema, la disponibilidad de los recursos necesarios (tanto humanos como tecnológicos y económicos), los tiempos estimados de desarrollo, las posibles afectaciones al cronograma general del proyecto y los riesgos asociados a la implementación. Cuando el análisis revela obstáculos técnicos o económicos significativos, se genera un informe detallado que documenta estas limitaciones, el cual se devuelve al ingeniero de requerimientos para su revisión y eventual reformulación de la solicitud.

Durante la fase de implementación, el equipo de desarrollo mantiene un registro sistemático de todos



los avances, documentando no solo el progreso en la codificación sino también cualquier incidencia encontrada y las desviaciones respecto a lo planeado originalmente. Esta documentación se actualiza constantemente en la herramienta de gestión de proyectos seleccionada, lo que permite mantener una trazabilidad completa de cada requerimiento a lo largo de todo su ciclo de vida.

Antes de considerar completa la implementación de un requerimiento, el líder técnico realiza una verificación exhaustiva para asegurar que el trabajo realizado cumple con todas las especificaciones técnicas originales, satisface los criterios de aceptación definidos y se ajusta a los estándares de calidad establecidos para el proyecto. Este proceso de control incluye una revisión por pares que sirve como filtro adicional antes de que la implementación sea entregada al ingeniero de requerimientos para su evaluación final.

La fase de aprobación formal involucra al cliente directamente, quien valida la implementación mediante sesiones de demostración práctica, pruebas de aceptación específicas y revisión de la documentación asociada. Solo después de obtener esta validación explícita por parte del cliente, el requerimiento se incorpora oficialmente al backlog del producto, donde será considerado para su planificación en los sprints correspondientes según la prioridad establecida.

La implementación exitosa de un requerimiento desencadena una serie de actualizaciones en la documentación del proyecto. Esto incluye la modificación de las historias de usuario afectadas, la actualización de los diagramas de arquitectura cuando sea necesario, la revisión de la documentación técnica relacionada y los ajustes correspondientes en los cronogramas del proyecto. La matriz de trazabilidad, herramienta fundamental para el seguimiento de requerimientos, se actualiza meticulosamente para reflejar el estado actual de cada elemento y su relación con los diferentes componentes del sistema.

El despliegue de los cambios aprobados sigue un protocolo estricto diseñado para minimizar riesgos. Este proceso incluye la ejecución de pruebas de regresión exhaustivas, el monitoreo continuo del desempeño en el entorno productivo y la preparación de un plan de rollback detallado que permita revertir los cambios en caso de detectarse problemas significativos durante o después de la implementación.

La última fase del proceso corresponde a la validación final por parte del cliente, quien realiza pruebas exhaustivas en el entorno productivo para verificar que se cumplen todas las funcionalidades solicitadas, que el rendimiento del sistema se mantiene dentro de los parámetros esperados y que la experiencia de usuario cumple con los estándares de calidad acordados. Solo después de superar satisfactoriamente esta etapa de validación se considera completo el ciclo de gestión para ese requerimiento específico.

A lo largo de todo el ciclo de gestión de requerimientos, se mantienen y actualizan constantemente varios artefactos críticos. Las historias de usuario sirven como documento principal para capturar los requerimientos funcionales y no funcionales en un formato accesible para todos los involucrados. La matriz de trazabilidad establece y mantiene las relaciones entre los requerimientos y los diferentes componentes del sistema, permitiendo un seguimiento preciso del impacto de cada cambio. Complementariamente, el registro de cambios documenta el histórico completo de todas las modificaciones realizadas, proporcionando visibilidad completa sobre la evolución de cada requerimiento a lo largo del tiempo.

Este proceso integral de gestión de requerimientos está diseñado para garantizar una administración eficiente de los cambios, minimizando riesgos y asegurando que cada iteración del producto entregue valor tangible al cliente, manteniendo siempre el equilibrio entre flexibilidad para adaptarse a nuevos requerimientos y el control necesario para garantizar la calidad y consistencia del producto final.

## **22. Proceso de verificación**

Cada requerimiento individual fue verificado mediante inspección formal para asegurar que cumpla con las características de ser atómico, no ambiguo, completo, consistente, trazable y verificable. Se definieron criterios de aceptación específicos para los requerimientos funcionales clave, lo que permite evaluar su cumplimiento durante el desarrollo. Asimismo, se establecieron métodos de prueba como demostraciones, inspecciones o análisis, asociados a cada requerimiento verificable, facilitando su validación técnica en etapas posteriores. Este enfoque asegura que cada requerimiento sea comprensible, implementable y alineado con los objetivos del sistema.

Una vez implementados, los requerimientos se verifican mediante un proceso sistemático que incluye pruebas funcionales, validaciones frente a los criterios de aceptación definidos y revisiones por parte del equipo técnico y el cliente. Cada funcionalidad desarrollada es evaluada para confirmar que cumple exactamente con lo especificado en el requerimiento, sin desviaciones ni omisiones. Además, se ejecutan pruebas de integración y pruebas de regresión para asegurar que la nueva funcionalidad no afecte negativamente a otras partes del sistema. La verificación se completa únicamente cuando el resultado esperado se cumple de manera consistente, y se documenta formalmente como “requerimiento satisfecho” en la matriz de trazabilidad.