

**Plataforma de soporte para la evaluación de proyectos de investigación presentados
ante el Comité de Ética de la Investigación del Hospital Universitario San Ignacio**

Pontificia Universidad Javeriana Bogotá - Grupo 12



ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

Mayo 2025

Versión 2.0

Jose Manuel Rodríguez
Laura Valentina Ovalle
Michael Joel González
María Paula Cardona

HISTORIAL DE CAMBIOS

Versión	Fecha	Sección del documento modificada	Descripción de cambios (corta)	Responsable (S)
01	Noviembre 2024	Todo el documento	Se creó la primer versión del documento	Jose Manuel Rodriguez Laura Valentina Ovalle Michael Joel Gonzalez Maria Paula Cardona
02	03/04/2025	Todo el documento	Se creó la segunda versión del documento, con la actualización de requerimientos.	María Paula Cardona
03	12/05/2025	Descripción global	Se implementó el apartado del documento con la información actualizada del proyecto	Laura Valentina Ovalle
04	13/05/2025	Requerimientos específicos	Se actualizaron los requerimientos de acuerdo a lo desarrollado en el proyecto	Laura Valentina Ovalle

Tabla 1: Historial de cambios

CONTENIDO

HISTORIAL DE CAMBIOS.....	1
CONTENIDO.....	2
LISTA DE TABLAS.....	4
LISTA DE FIGURAS.....	4
1. INTRODUCCIÓN.....	5
Propósito.....	5
Alcance.....	9
Definiciones, Acrónimos, y Abreviaciones.....	10
Referencias.....	11
Apreciación Global.....	12
2. DESCRIPCIÓN GLOBAL.....	13
Perspectiva del producto.....	13
Interfaces con el sistema.....	13
Interfaces con el usuario.....	14
Interfaces con el hardware.....	15
Interfaces con el software.....	15
Interfaces de comunicación.....	17
Restricciones de memoria.....	18
Operaciones.....	19
Requerimientos de adaptación del sitio.....	22
Funciones del producto.....	23
Características del usuario.....	25
Restricciones.....	26
Modelo del dominio.....	27
Suposiciones y dependencias.....	29
Distribución de requerimientos.....	31
3. REQUERIMIENTOS ESPECÍFICOS.....	32
Requerimientos de interfaces externas.....	32
Interfaces con el usuario.....	32
Interfaces con el hardware.....	37
Interfaces con el software.....	38
Interfaces de comunicaciones.....	39
Características del producto de software.....	42
Requerimientos de desempeño.....	51
Restricciones de diseño.....	52
Atributos del sistema de software (No funcionales).....	54
Confiability.....	54
Disponibilidad.....	54
Seguridad.....	55
Mantenibilidad.....	55
Portabilidad.....	55
Requerimientos de la base de datos.....	55

4. PROCESO INGENIERÍA DE REQUERIMIENTOS.....	57
5. PROCESO DE VERIFICACIÓN.....	58
6. ANEXOS.....	60

LISTA DE TABLAS

- Tabla 1: Historial de cambios
- Tabla 2. Plantilla para el formato de documentos de consentimiento informado
- Tabla 3. Formato de evaluación ética para documentos de consentimiento informado
- Tabla 4. Tabla con definiciones, acrónimos y abreviaciones
- Tabla 5. Restricciones de memoria
- Tabla 6. Tabla de características de usuario
- Tabla 7. Elementos del modelo de dominio
- Tabla 8. Asignación de funcionalidades de acuerdo al usuario
- Tabla 9. Requerimientos funcionales del módulo de autenticación y registro
- Tabla 10. Requerimientos funcionales del módulo de gestión de archivos para el evaluador
- Tabla 11. Requerimientos funcionales del módulo de interacción con el modelo de IA
- Tabla 12. Requerimientos funcionales del módulo de gestión de correos
- Tabla 13. Requerimientos funcionales del módulo de análisis y estadísticas
- Tabla 14. Requerimientos funcionales del módulo de configuración del modelo de IA
- Tabla 15. Requerimientos funcionales del módulo de gestión del consentimiento para el investigador
- Tabla 16. Tabla de atributos de calidad

LISTA DE FIGURAS

- Figura 1. Diagrama de casos de uso del sistema (ver anexo 1)
- Figura 2. Modelo de datos del sistema (ver anexo 2)

1. INTRODUCCIÓN

Propósito

El creciente número de proyectos de investigación presentados al Comité de Ética del Hospital Universitario San Ignacio (HUSI), de la Pontificia Universidad Javeriana, ha generado una carga de trabajo cada vez más alta para sus miembros, en especial en la gestión y revisión de dichos proyectos.

Este aumento en la demanda ha derivado en significativas demoras en la aprobación de estudios no solo en el HUSI, sino también en muchos otros centros de investigación en Colombia. Dicha situación afecta gravemente el avance de los proyectos y, en consecuencia, dificulta el cumplimiento de los cronogramas establecidos por los investigadores, reduciendo la capacidad de los equipos de investigación para progresar de manera eficiente y oportuna.

El Comité de Ética del HUSI recibe más de 200 solicitudes de revisión al año, y cada una requiere un análisis ético detallado y riguroso. Actualmente, este proceso se realiza manualmente, lo que lo convierte en una tarea altamente laboriosa y propensa a errores humanos.

La mayoría de estos proyectos presentan problemas de *falta de completitud*, lo que significa que los documentos recibidos no contienen toda la información necesaria. Como consecuencia, estos casos requieren una doble revisión: primero, para identificar y corregir las deficiencias, y luego, una vez completados, para evaluar el cumplimiento de las normas éticas.

La ausencia de un sistema ágil para asistir en la verificación de los documentos prolonga el tiempo de evaluación de cada proyecto, afectando tanto la eficiencia como la calidad del proceso.

En el contexto del presente proyecto, nos centraremos en una parte crítica del proceso: la verificación de los documentos de consentimiento informado, tanto en términos de su completitud como de su adecuación a los estándares éticos.

Habiendo dicho esto, es importante mencionar que un consentimiento informado es un documento legal que los participantes de un estudio deben firmar, indicando que han sido debidamente informados sobre los objetivos, procedimientos, riesgos y beneficios del estudio, así como sobre sus derechos, incluida la posibilidad de retirarse en cualquier momento. Este documento es fundamental para garantizar que los estudios clínicos y de investigación se realicen de manera ética, protegiendo los derechos y el bienestar de los participantes.

Aunque los consentimientos informados cuentan con una estructura definida, basada en normativas específicas y plantillas diseñadas para facilitar su evaluación, el proceso continúa siendo exigente y consume mucho tiempo a nivel administrativo.

Por ello, a los investigadores se les proporciona la siguiente plantilla estándar como guía para la elaboración adecuada de estos documentos de consentimiento informado:

INTRODUCCIÓN

En verde encontrará instrucciones que deben ser eliminadas.

En rojo encontrará los campos que debe reemplazar con la información correspondiente con su proyecto.

Código del sujeto: _____

Este documento de consentimiento es para uso en un proyecto de investigación que involucrará a sujetos que quizá tengan o no la capacidad de dar consentimiento para su participación. En este documento de consentimiento, “usted” hace referencia al participante de la investigación. Si usted es un representante legal, recuerde que “usted” hace referencia al participante de investigación.

Usted está siendo invitado a participar en el proyecto de investigación “nombre del proyecto”. Este documento le proporciona la información necesaria para que usted participe voluntaria y libremente. El proyecto es realizado por nombre de la(s) institución(es) participante(s). Antes de dar su consentimiento, usted necesita entender plenamente el propósito de su decisión. Este proceso se denomina consentimiento informado. Una vez que haya leído este documento y resuelto con el investigador las dudas, se le pedirá que firme este formato en señal de aceptación de participar.

INFORMACIÓN GENERAL

1. ¿Por qué se debe realizar este estudio? Describa brevemente el problema y la pertinencia del estudio.
2. ¿Cuál es el objetivo de este estudio? Enuncie el objetivo general.
3. ¿En qué consiste el estudio? Describa de manera clara y sencilla los procedimientos, intervenciones y su propósito, incluyendo los experimentales (aleatorización, cegamiento, uso de placebo, etc.).
4. ¿Cuáles son las molestias o los riesgos esperados? Describir molestias y riesgos. En caso de que no existan molestias o riesgos igualmente se debe informar.
5. ¿Cuáles son los beneficios que puedo obtener por participar? Enunciar beneficios. En caso de que no existan beneficios igualmente se debe informar.
6. ¿Existe confidencialidad en el manejo de mis datos? Este proyecto se acoge a la ley 1581 de 2012 (Hábeas Data) que aplica para el tratamiento de datos personales.
Indique brevemente cómo se manejarán los datos:
 - Describa en dónde se almacenarán los datos e información, los mecanismos de custodia y seguridad de los mismos y el tiempo de custodia.
 - Describa quiénes tendrán acceso a la información y bajo qué parámetros de seguridad se accederá a ellos
 - Describa cómo se llevará a cabo la anonimización de los datos tanto para los análisis como para la publicación de los resultados.
 - Describa la posibilidad de conocer los datos personales registrados en la base de datos del estudio, solicitar rectificación de los mismos y de retirar su consentimiento para el tratamiento de los datos en cualquier momento del estudio, excepto a partir de la anonimización.
 - Describa los procesos de transferencia de datos a terceros, en caso de estudios colaborativos, y la garantía de mantener la privacidad, confidencialidad y seguridad en el tratamiento por parte del tercero.
7. ¿Existen procedimientos alternativos que pudieran ser ventajosos para mí?
En caso de que se realicen intervenciones con dispositivos, procedimiento médico-quirúrgico o medicamentos explicar si existen otras intervenciones que puedan realizarse para la patología del paciente.
8. Expresar el compromiso de proporcionar información actualizada obtenida durante el estudio, aunque ésta pudiera afectar la voluntad del sujeto para continuar participando.
En caso de realización de estudios que requieran entrega de resultados de procedimiento o consejería (genética, por ejemplo), explicar el proceso.
9. ¿Existe alguna obligación financiera? Participar en este estudio no tiene ningún costo económico para usted. En caso contrario, enunciar costos generados por participar en el estudio, describir cuáles: transporte, alimentación etc. y la forma cómo serán asumidos con cargo al presupuesto del proyecto.

10. ¿Cuánto tiempo durará mi participación en el estudio? **Indique el tiempo de participación y en caso de ser necesario la frecuencia de las intervenciones.**
11. ¿Qué sucede si no deseo participa o me retiro del estudio? Usted puede decidir no participar o retirarse en cualquier momento del estudio, sin que esto afecte de manera alguna el tratamiento médico que necesita.

En estudios en los que se involucran menores de edad o discapacitados físicos y mentales adultos, que propongan intervenciones o procedimientos que superan el riesgo mínimo (no hacen parte del estándar de manejo), se debe evaluar la capacidad de entendimiento de acuerdo a la Res. 8430 de 1993. (Mantener o retirar el numeral 12 de acuerdo a la naturaleza del estudio)

12. Al menor de edad o la persona que usted representa se le solicitará informar su aceptación para ingresar al estudio a través de un Asentimiento Informado. De acuerdo a la Resolución 8430 de 1993, el proceso de asentimiento debe estar precedido de una valoración de razonamiento entendimiento y lógica realizada por un psicólogo, neurólogo o psiquiatra. A través de este documento se solicita su autorización para realizar la valoración.

En caso de estudios con riesgo mayor al mínimo (mantener o retirar el numeral 13 de acuerdo a la naturaleza del estudio):

13. ¿Qué sucede si esta investigación afecta directamente mi salud? **Se debe explicar claramente al paciente que el estudio está diseñado para evitar cualquier efecto nocivo sobre la salud y que, si se llegara a presentar un efecto adverso, el equipo investigador hará todo lo que sea necesario asegurar la atención en salud y evitar que tenga consecuencias para su salud.**

En caso de Ensayos Clínicos debe expresar la existencia de una póliza de responsabilidad civil, nombre de la aseguradora, que cubrirá los costos de atención en salud en caso de que la intervención del estudio haya causado un evento adverso. Utilizar el siguiente párrafo:

Si bien es poco probable que se produzca una lesión, el/la **incluir nombre del patrocinador** (patrocinador del estudio) realizará el pago de los gastos médicos relacionados con una lesión directamente causada por los medicamentos administrados/los procedimientos realizados en esta investigación al centro hospitalario del estudio, y el centro del estudio le proporcionará el tratamiento médico sin costo alguno para usted o para el sistema de salud. Para garantizar la cobertura de los riesgos o los posibles daños y lesiones y la atención adecuada que pueda surgir del estudio, **nombre del patrocinador del estudio**, ha contratado un seguro con la compañía **nombre de la compañía**, con dirección en **incluir dirección**, incluir ciudad Colombia.

En caso que necesite información o en caso de emergencia póngase en contacto con el/la investigador/a del estudio **Dr./Dra. incluir nombre**, celular **XXXXXX**. A fin de activar la póliza la investigadora contactará a la Directora de la Oficina de Investigaciones del Hospital Universitario San Ignacio, **incluir nombre Director/a Of. Investigaciones, celular o al correo email**.

Al firmar este formulario usted no renuncia a ningún derecho legal, aceptar atención médica o aceptar el pago de lo el/la s gastos médicos.

AUTORIZACIÓN

He comprendido las explicaciones que en un lenguaje claro y sencillo se me han brindado. El investigador me ha permitido expresar todas mis observaciones y ha aclarado todas las dudas y preguntas que he planteado respecto a los fines, métodos, ventajas, inconvenientes y pronóstico de participar en el estudio. Se me ha proporcionado una copia de este documento.

Al firmar este documento doy mi consentimiento voluntario para participar en el estudio "**nombre del estudio**".

Si usted tiene dudas acerca de su participación en este estudio puede comunicarse con el investigador principal: **Dr/a. incluir nombre, celular, teléfono ext xxx**. Presidente del Comité de Ética Institucional: **Incluir nombre**, Calle 42 N° 4-49, oficina 507. Teléfono 5946161 ext. 2470.

NOTA: Usted encontrará dos formatos de firmas. El primero es el genérico, el segundo aplica para sujetos que requieren representación legal. Escoja el que aplique a su estudio y elimine el otro.

(En la parte final de este documento se encuentran los apartados para la firma del participante, de los testigos imparciales, miembro del equipo de investigación que realiza el proceso de consentimiento y en caso de ser menor de edad, de los representantes legales).

Tabla 2. Plantilla para el formato de documentos de consentimiento informado

Los evaluadores, por su parte, una vez recibido este documento, deben llenar el siguiente formato para validar que este cumpla con los requisitos y normas éticas establecidas:

Característica		Si cumple	No cumple	No aplica
1	Contiene versión y fecha del FCI			
2	Contiene título del proyecto e instituciones participantes			
3	Describe la justificación y los objetivos de la investigación. Problema, pertinencia del estudio, objetivo general			
4	Describe los procedimientos del estudio y su propósito, incluyendo la identificación de aquellos que son experimentales (aleatorización, cegamiento, uso de placebo, etc.).			
5	Determina las molestias o los riesgos esperados. Si no existen informar.			
6	Determina los beneficios que puedan obtenerse. Si no existen informar.			
7	Indica la seguridad que no se identificará al sujeto y que se mantendrá la confidencialidad de la información relacionada con su privacidad. Se acoge a la Ley 1581 de 2012 (Habeas Data) e indica cómo se manejarán los datos: <ul style="list-style-type: none"> • Dónde se almacenarán los datos. • Mecanismos de custodia y seguridad de los mismos. • Tiempo de custodia. • Quiénes tendrán acceso a la información y bajo qué parámetros de seguridad se accederá a ellos. • Tratamiento de datos: codificación o anonimización tanto para los análisis como para la publicación de los resultados. • Posibilidad de conocer sus datos personales registrados en la base de datos del estudio, solicitar rectificación de los mismos y de retirar su consentimiento para el tratamiento de los datos en cualquier momento del estudio, excepto a partir de la anonimización. • Describe los procesos de transferencia de datos a terceros, en caso de estudios colaborativos, y la garantía de mantener la privacidad, confidencialidad y seguridad en el tratamiento por parte del tercero. 			
8	Indica los procedimientos alternativos que pudieran ser ventajosos para el sujeto.			
9	Expresa el compromiso de proporcionar información actualizada obtenida durante el estudio, aunque ésta pudiera afectar la voluntad del sujeto para continuar participando. En caso de realización de estudios que requieran entrega de resultados de procedimiento o consejería (genética, por ejemplo), se debe explicar el proceso.			

10	Expresa la garantía de que, si existen gastos adicionales, éstos serán cubiertos por el presupuesto de la investigación o de la institución responsable de la misma. Enunciar costos generados por participar en el estudio, ej.: transporte, alimentación etc. y la forma cómo serán reembolsados (bonos de alimentación o transporte, dinero).			
11	Indica tiempo de permanencia en el estudio: Tiempo de participación, frecuencia de las intervenciones si aplica.			
12	Indica la libertad de retirar su consentimiento en cualquier momento y dejar de participar en el estudio sin que por ello se creen perjuicios para continuar su cuidado y tratamiento.			

Tabla 3. Formato de evaluación ética para documentos de consentimiento informado

Como se evidencia, la verificación de estos consentimientos implica que los evaluadores completen un formato en el que cada criterio debe ser clasificado como 'cumple', 'no cumple' o 'no aplica', de acuerdo con la normativa correspondiente. Este proceso exige una revisión minuciosa y un análisis riguroso, ya que es necesario identificar y validar todos los elementos normativos relevantes según el tipo de estudio evaluado.

Debido a esto, se resalta la necesidad de agilizar el proceso actual de verificación manual, que no solo es lento, sino que también incrementa la probabilidad de errores e inconsistencias. Esto ya que la alta carga de trabajo y la revisión manual han creado un cuello de botella que retrasa la aprobación y ejecución de los proyectos de investigación, situación que limita el desarrollo de investigaciones médicas de calidad en el tiempo necesario, y afecta la capacidad de respuesta del comité frente a las solicitudes.

De esta manera, se propone como propósito del proyecto implementar soluciones tecnológicas que permitan asistir gran parte del proceso de revisión ética, específicamente en lo relacionado con la verificación de los consentimientos informados, sin comprometer la rigurosidad y calidad de las evaluaciones. Esto no sólo reduciría los tiempos de evaluación, sino que también contribuiría a minimizar errores, mejorar la consistencia en la revisión, y liberar tiempo para que los miembros del comité puedan enfocarse en tareas de mayor complejidad.

Alcance

El proyecto se enfocará en el desarrollo de una plataforma de soporte para la evaluación ética, dirigida al Comité de Ética del Hospital Universitario San Ignacio. Esta plataforma estará orientada a facilitar la creación y revisión de documentos de consentimiento informado en proyectos de investigación que involucren seres humanos.

Para ello, se integrará un modelo de lenguaje a gran escala (LLM), ajustado mediante técnicas de prompt engineering, con el fin de analizar los documentos cargados y verificar su conformidad con las normativas éticas internacionales. La plataforma contará con una interfaz web intuitiva que permitirá a los evaluadores cargar archivos, visualizar los resultados del análisis automatizado y recibir retroalimentación detallada sobre el grado de cumplimiento normativo. Esto permitirá detectar con mayor facilidad omisiones o inconsistencias, y reducirá significativamente el tiempo invertido en el análisis manual.

Es importante precisar que la plataforma estará diseñada exclusivamente para identificar si un criterio normativo “cumple” o “no cumple” con los estándares definidos. La clasificación de un criterio como “no aplica” quedará bajo la responsabilidad del evaluador humano, quien tomará esta decisión con base en el contexto específico de cada estudio.

Adicionalmente, se incluirá una interfaz dirigida a los investigadores, la cual les permitirá generar documentos de consentimiento de manera más eficiente. En lugar de completarlos manualmente, podrán ingresar la información requerida a través de un formulario en línea y descargar el documento finalizado, disminuyendo así errores por falta de completitud y agilizando la revisión por parte del comité.

El propósito general de la plataforma es mejorar la eficiencia y precisión en el proceso de evaluación ética, asegurando que los consentimientos informados cumplan con los estándares requeridos. Cabe destacar que la solución propuesta no abordará la evaluación integral de todos los aspectos éticos de los proyectos de investigación, ni contemplará la integración con otros sistemas del hospital, soporte multilingüe o la sustitución del juicio de los evaluadores humanos.

Definiciones, Acrónimos, y Abreviaciones

Esta tabla presenta un glosario con algunos de los términos y herramientas tecnológicas más relevantes para el desarrollo del proyecto. Incluye acrónimos, metodologías, lenguajes de programación y plataformas utilizadas a lo largo del proceso. Cada término cuenta con una breve descripción que facilita su comprensión y proporciona contexto sobre su aplicación y utilidad en proyectos tecnológicos, abarcando tanto el ámbito de la inteligencia artificial y el procesamiento de lenguaje natural como las metodologías ágiles y los entornos de desarrollo backend.

Término	Definición
IA	Inteligencia Artificial
LLM	Large Language Model, modelo de lenguaje a gran escala para procesamiento de lenguaje natural
NLP	Natural Language Processing, procesamiento de lenguaje natural
HUSI	Hospital Universitario San Ignacio
Mockup	Modelo visual o esquema del diseño de una interfaz
Sprint	Periodo de tiempo en la metodología ágil donde se completan tareas específicas
TDD	Test-Driven Development, desarrollo guiado por pruebas
API	Application Programming Interface, interfaz de programación de aplicaciones
Git	Sistema de control de versiones distribuido para gestionar cambios en el código fuente
GitHub	Plataforma de colaboración y alojamiento de repositorios Git
Node.js	Entorno de ejecución de JavaScript que permite crear aplicaciones escalables en backend
TypeScript	Superconjunto de JavaScript que añade tipado estático y otras características avanzadas

MongoDB	Base de datos NoSQL ideal para datos no estructurados o semiestructurados
Scrum	Metodología ágil de gestión de proyectos enfocada en entregas incrementales mediante sprints
JAVA	Lenguaje de programación de propósito general, orientado a objetos, utilizado para desarrollo en backend
SpringBoot	Framework de desarrollo para Java que simplifica la creación de aplicaciones web y servicios backend
Angular	Framework de desarrollo de aplicaciones web basado en JavaScript desarrollado por Google, enfocado en el desarrollo de aplicaciones de una sola página (SPA).
React	Biblioteca de JavaScript desarrollada por Facebook para construir interfaces de usuario, especialmente útiles para crear aplicaciones de una sola página.
Express.js	Framework de servidor web para Node.js que facilita la creación de aplicaciones web y APIs RESTful, proporcionando herramientas y funcionalidades para el backend.
Prompt Engineering	Técnica en Inteligencia Artificial para diseñar instrucciones (prompts) que optimicen las respuestas de modelos de lenguaje como los LLMs.
JavaScript	Lenguaje de programación de propósito general que se utiliza principalmente en desarrollo web para mejorar la interactividad y la dinámica de las páginas web.
TypeScript	Superconjunto de JavaScript que añade tipado estático y características avanzadas, lo que facilita el mantenimiento y la escalabilidad de aplicaciones grandes.
Jest	Framework de pruebas en JavaScript, popular para probar aplicaciones React y JavaScript en general, ofreciendo funcionalidades de prueba unitarias y de integración.
Frontend	Parte de una aplicación que interactúa directamente con el usuario.
HTTPS	Protocolo de Transferencia de Hipertexto
TCP/IP	Protocolo de comunicación estándar para redes
RESTful	Arquitectura que sigue el modelo REST (Representational State Transfer)
Docker	Plataforma que permite crear, desplegar y ejecutar aplicaciones en contenedores
Backend	Parte de una aplicación que se ejecuta en el servidor y se encarga del procesamiento de datos y la lógica de negocio, conectando con bases de datos y gestionando las solicitudes de los clientes.

Tabla 4. Tabla con definiciones, acrónimos y abreviaciones

Referencias

- Node.js. (s.f.). Node.js. Recuperado de <https://nodejs.org/>
- TypeScript. (s.f.). TypeScript: Typed JavaScript at Any Scale. Recuperado de <https://www.typescriptlang.org/>
- Python Software Foundation. (s.f.). Python. Recuperado de <https://www.python.org/>
- Angular. (s.f.). One framework. Mobile & desktop. Recuperado de <https://angular.io/>
- React. (s.f.). A JavaScript library for building user interfaces. Recuperado de <https://reactjs.org/>
- Express.js. (s.f.). Express - Node.js web application framework. Recuperado de <https://expressjs.com/>
- Jest. (s.f.). Delightful JavaScript Testing. Recuperado de <https://jestjs.io/>

- MongoDB. (s.f.). MongoDB: The application data platform. Recuperado de <https://www.mongodb.com/>
- Git. (s.f.). Git. Recuperado de <https://git-scm.com/>
- GitHub. (s.f.). GitHub: Where the world builds software. Recuperado de <https://github.com/>
- Visual Studio Code. (s.f.). Code editing. Redefined. Recuperado de <https://code.visualstudio.com/>
- Visual Paradigm. (s.f.). Visual Paradigm - UML, Agile, PMBOK, TOGAF, BPMN and More. Recuperado de <https://www.visual-paradigm.com/>
- Microsoft Teams. (s.f.). Microsoft Teams. Recuperado de <https://www.microsoft.com/es-co/microsoft-teams/group-chat-software>
- Microsoft Outlook. (s.f.). Outlook – free personal email and calendar from Microsoft. Recuperado de <https://outlook.live.com/>
- Discord. (s.f.). Discord | Your Place to Talk and Hang Out. Recuperado de <https://discord.com/>
- WhatsApp. (s.f.). WhatsApp. Recuperado de <https://www.whatsapp.com/>
- Google Docs. (s.f.). Google Docs: Free Online Documents for Personal Use. Recuperado de <https://www.google.com/docs/about/>
- ClickUp. (s.f.). One app to replace them all. Recuperado de <https://clickup.com/>
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781.
- Le, Q., & Mikolov, T. (2014). Distributed Representations of Sentences and Documents. In Proceedings of the 31st International Conference on Machine Learning (pp. 1188–1196).
- Friedl, J. E. F. (2006). Mastering Regular Expressions (3rd ed.). O'Reilly Media.
- Jurafsky, D., & Martin, J. H. (2009). Speech and Language Processing (2nd ed.). Prentice Hall.
- Google Cloud. (s.f.). Document AI. Recuperado de <https://cloud.google.com/document-ai>
- Amazon Web Services. (s.f.). Amazon Textract. Recuperado de <https://aws.amazon.com/textract/>
- IBM. (s.f.). Watson Discovery. Recuperado de <https://www.ibm.com/cloud/watson-discovery>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
- Chollet, F. (2018). Deep Learning with Python. Manning Publications.

Apreciación Global

Este documento de Especificación de Requisitos del Software (SRS) para la plataforma destinada al *Comité de Ética del Hospital Universitario San Ignacio* está organizado para facilitar su comprensión y asegurar la trazabilidad de los requisitos a lo largo del desarrollo del proyecto. La estructura del documento se detalla a continuación:

- Introducción: Define el propósito del documento, el alcance del proyecto y los términos clave utilizados, proporcionando contexto sobre el análisis ético de los consentimientos informados mediante el uso de modelos de lenguaje a gran escala (LLM), ajustados por técnicas de prompt engineering.
- Descripción Global: Presenta una visión general del producto, abarcando su perspectiva funcional, las interfaces con sistemas, usuarios, hardware y software, así como las restricciones operativas y de memoria. Además, describe el entorno tecnológico en el que se desarrollará la plataforma, incluyendo la integración con herramientas de virtualización, como

Docker para la gestión de contenedores, y el uso de recursos computacionales locales proporcionados por la institución para soportar los procesos de mayor carga.

- **Requerimientos Específicos:** Esta sección describe en detalle las funcionalidades clave de la plataforma, tales como la carga de documentos de consentimiento informado, el análisis automatizado mediante un modelo de lenguaje a gran escala (LLM) ajustado con técnicas de prompt engineering, y la generación de retroalimentación específica para cada criterio ético evaluado. También se abordan las características de la interfaz web que permiten a los evaluadores revisar los resultados y gestionar los casos de forma intuitiva. Adicionalmente, se establecen los requerimientos de desempeño esperados, incluyendo tiempos de respuesta y tolerancia a fallos, así como las restricciones de diseño que condicionan la selección tecnológica y la arquitectura del sistema
- **Modelo del Dominio:** Proporciona un análisis inicial del sistema mediante diagramas UML, describiendo los objetos del negocio y sus relaciones, lo cual permite una representación clara del flujo de datos y las entidades clave dentro de la plataforma.
- **Suposiciones y Dependencias:** Esta sección enumera las dependencias del proyecto, como la disponibilidad de recursos computacionales locales proporcionados por la institución para ejecutar los modelos de lenguaje, así como el uso de componentes de software reutilizables. Estas suposiciones son fundamentales para el desarrollo exitoso de la plataforma y deben ser evaluadas y actualizadas continuamente a lo largo del ciclo del proyecto.
- **Proceso de Verificación y Validación:** Describe los métodos utilizados para asegurar la calidad del software mediante pruebas automatizadas y revisiones manuales. El documento incluye procedimientos para validar los requerimientos funcionales y no funcionales, siguiendo estándares establecidos en el SPMP.

2. DESCRIPCIÓN GLOBAL

Perspectiva del producto

Interfaces con el sistema

Este proyecto contempla el desarrollo de una plataforma web completamente nueva, orientada a agilizar tanto la creación como la revisión de documentos de consentimiento informado en proyectos de investigación. Para ello, se integrará un modelo de inteligencia artificial generativa ya existente, al cual se le formularán instrucciones específicas mediante técnicas de prompt engineering, con el objetivo de obtener análisis claros y precisos del cumplimiento normativo de los documentos.

Dado que se trata de un sistema autónomo, las interfaces necesarias para su funcionamiento se limitan a tres componentes fundamentales: la comunicación con la API del modelo de IA generativa, la interacción con una base de datos no relacional para el almacenamiento estructurado de la información procesada, y la conexión con un sistema de almacenamiento de objetos (MinIO) para la gestión de los documentos cargados y generados por los usuarios.

- **Interfaz con la API de IA generativa:**

La plataforma se comunica con el modelo de lenguaje a través de llamadas RESTful, mediante las cuales se envían los documentos de consentimiento informado junto con prompts cuidadosamente diseñados. Estos prompts orientan la respuesta del modelo para evaluar los criterios éticos de forma estructurada.

En el entorno de desarrollo entregado, esta comunicación se realiza sin cifrado (HTTP), dado que el sistema será desplegado localmente en contenedores Docker sin exposición pública. Sin embargo, en un entorno de producción real —por ejemplo, si se implementa en los servidores del Hospital Universitario San Ignacio— será responsabilidad del equipo encargado del despliegue asegurar la implementación de HTTPS u otros mecanismos equivalentes para proteger la confidencialidad e integridad de los datos transmitidos.

Adicionalmente, la plataforma ya cuenta con un sistema de autenticación robusto basado en JSON Web Tokens (JWT) y control de acceso mediante validación de roles a través de middlewares, lo cual garantiza que únicamente los usuarios autorizados puedan interactuar con los distintos endpoints de la aplicación.

- Interfaz con la Base de Datos No Relacional (MongoDB):

La plataforma utiliza MongoDB como base de datos NoSQL, destinada a almacenar de forma eficiente la información estructurada relacionada con usuarios, casos, evaluaciones y configuraciones de prompts. La interacción con la base de datos se gestiona mediante un ORM o librerías especializadas, lo que garantiza un acceso seguro y un rendimiento óptimo.

- Interfaz con el sistema de almacenamiento de objetos (MinIO):

Los documentos cargados por los investigadores y los generados por el sistema (como reportes de evaluación) se almacenan en MinIO, una solución de almacenamiento de objetos compatible con el protocolo S3. Esta herramienta permite gestionar los archivos de manera segura, organizada y escalable, preservando su disponibilidad y persistencia a lo largo del tiempo.

Dado que la plataforma maneja información sensible, se han implementado mecanismos de seguridad apropiados para el entorno local, incluyendo autenticación, control de sesiones, encriptación de contraseñas y gestión de permisos por rol. No obstante, en un entorno de producción se deberán incorporar medidas adicionales, como el cifrado de las comunicaciones mediante HTTPS y la gestión adecuada de certificados, de acuerdo con las políticas de seguridad de la institución correspondiente.

Finalmente, se entregará al Hospital Universitario San Ignacio una documentación técnica completa de las interfaces implementadas, incluyendo métodos de acceso, estructuras de datos, formatos de solicitud y respuesta, y parámetros clave. Esta documentación está diseñada para facilitar futuras integraciones o adaptaciones según los requisitos del entorno donde la plataforma sea desplegada.

Interfaces con el usuario

La plataforma implementa una única interfaz de usuario basada en una aplicación web, que actúa como el punto central de interacción para todos los perfiles. Esta interfaz ha sido diseñada para ofrecer una experiencia clara, fluida y accesible, permitiendo a los usuarios acceder a las

funcionalidades del sistema de forma intuitiva, independientemente de su nivel de experiencia tecnológica.

Aunque todos los usuarios acceden desde la misma plataforma web, las funcionalidades disponibles varían en función del rol asignado:

- **Evaluadores:** Tienen acceso a herramientas avanzadas que les permiten cargar documentos de consentimiento informado, ejecutar evaluaciones automáticas utilizando el modelo de inteligencia artificial, consultar los resultados detallados de dichas evaluaciones, editarlos en caso de considerarlo necesario, descargar informes generados por la plataforma y enviarlos por correo electrónico a los investigadores. Además, cuentan con un panel estadístico que presenta métricas clave relacionadas con el proceso de evaluación, como tiempos promedio de análisis, criterios más frecuentes de no cumplimiento y volumen de documentos revisados.
- **Investigadores:** Pueden ingresar información directamente a través de un formulario guiado, el cual genera automáticamente un documento de consentimiento informado basado en la plantilla establecida por el Comité de Ética. También tienen acceso a un historial de casos previamente creados, donde pueden consultar sus documentos enviados a evaluación.

La interfaz ha sido optimizada para garantizar usabilidad, organización visual y accesibilidad. Por ejemplo, el proceso de registro de usuario ha sido diseñado para ser rápido, seguro y guiado paso a paso, promoviendo una experiencia satisfactoria desde el primer ingreso a la plataforma.

Interfaces con el hardware

Dado que el proyecto consiste en una plataforma web, no se requiere integración con hardware especializado ni con periféricos externos. Toda la comunicación entre los componentes del sistema se gestiona de manera virtual, utilizando la infraestructura de red existente y el protocolo TCP/IP. Durante el desarrollo, la aplicación se ejecuta en dispositivos de cómputo personales y opera a través del puerto estándar HTTP 80, lo cual permite un acceso sencillo sin configuraciones adicionales en entornos locales.

Inicialmente se contempló la posibilidad de utilizar recursos computacionales locales del Centro de Investigación de la Pontificia Universidad Javeriana (CINE) para el procesamiento del modelo de lenguaje a gran escala (LLM). Sin embargo, esta alternativa no fue necesaria, ya que el análisis de los documentos se realiza mediante el consumo de una API externa que expone un modelo de inteligencia artificial generativa ya entrenado. La interacción con este modelo se basa en el envío de documentos y prompts específicos, sin requerir procesamiento intensivo en los equipos locales.

En resumen, la plataforma no requiere interfaces de hardware especiales y ha sido diseñada para funcionar eficientemente en entornos locales utilizando contenedores Docker. Las tareas de análisis se delegan a una API externa, lo que reduce significativamente la carga computacional necesaria en el entorno del usuario o del servidor donde se despliegue.

Interfaces con el software

Esta sección describe las interfaces de software requeridas para que el sistema interactúe de forma adecuada con las tecnologías y herramientas utilizadas durante su desarrollo, despliegue y operación. Incluye tanto sistemas operativos, gestores de bases de datos, herramientas de virtualización, motores de ejecución, como también frameworks específicos del frontend y backend.

A continuación, se detallan las interfaces previstas:

- **Sistemas Operativos:**
La plataforma es compatible con entornos basados en Linux y Windows, aunque su ejecución está completamente contenida en un entorno Dockerizado, por lo que el sistema operativo subyacente no afecta directamente su funcionamiento. La elección del sistema operativo para el servidor dependerá del entorno de producción definido por la institución que realice el despliegue.
- **Base de Datos (MongoDB):**
Se utiliza MongoDB como sistema gestor de base de datos NoSQL, adecuado para manejar información estructurada y semiestructurada relacionada con usuarios, casos, evaluaciones y configuraciones. MongoDB se ejecuta dentro de un contenedor Docker como parte del entorno unificado definido en Docker Compose.
- **Contenedores Docker:**
Todo el sistema está empaquetado y orquestado mediante Docker Compose, lo que permite una arquitectura modular, portable y reproducible. Se definen contenedores individuales para el frontend, backend, base de datos (MongoDB) y sistema de almacenamiento de archivos (MinIO), garantizando coherencia entre los entornos de desarrollo, prueba y producción.
- **Modelos de Lenguaje (LLM) a través de APIs:**
La plataforma no entrena ni ejecuta localmente modelos de lenguaje. En su lugar, se integra con APIs comerciales de proveedores como Gemini (Google) y Groq, lo que permite acceder a una variedad de modelos pre-entrenados, incluyendo Gemini Pro, LLaMA, DeepSeek, entre otros. El análisis ético de los documentos se realiza mediante prompt engineering, técnica que permite guiar el comportamiento del modelo a través de instrucciones cuidadosamente diseñadas, sin requerir técnicas de procesamiento de lenguaje natural (NLP) tradicionales.
- **Frameworks y herramientas de desarrollo:**
En el backend, se utiliza Express.js sobre Node.js, estructurado bajo los principios de Clean Architecture, lo que promueve una separación clara de responsabilidades y una base de código mantenible.
En el frontend, el sistema está construido con React y TypeScript, siguiendo el enfoque de Atomic Design para lograr una interfaz modular y reutilizable. Se emplea Vite como herramienta de desarrollo por su velocidad y simplicidad.
Además, se integran librerías como TanStack (para gestión de datos y tablas) y shadcn/ui (para componentes de interfaz accesibles y personalizables).
- **Motor de ejecución (Bun):**
Todo el proyecto es gestionado y ejecutado con Bun, una herramienta moderna que reemplaza a Node.js en el entorno de ejecución. Bun proporciona funcionalidades completas como gestor de paquetes, servidor HTTP, ejecutor de scripts y empaquetador, ofreciendo mayor rendimiento y simplicidad en comparación con soluciones tradicionales basadas en múltiples herramientas.

Interfaces de comunicación

De acuerdo con el diseño de la plataforma, el sistema web desarrollado se apoya en diversas interfaces de comunicación para garantizar el correcto intercambio de datos entre sus componentes internos y servicios externos.

- **Protocolo de comunicación (TCP/IP):**
La plataforma utiliza el protocolo TCP/IP como base para la transmisión de datos entre los distintos módulos: frontend, backend, base de datos y almacenamiento. Este protocolo es ampliamente utilizado en entornos web y garantiza confiabilidad y compatibilidad entre componentes, aspectos esenciales para asegurar una operación estable del sistema.
- **Comunicación con APIs externas:**
La interacción con los modelos de lenguaje generativo se realiza a través de APIs externas (como las ofrecidas por Gemini y Groq), utilizando solicitudes HTTP.
En el entorno de desarrollo, no se ha implementado cifrado HTTPS, ya que el despliegue se realiza localmente mediante contenedores Docker y sin exposición a redes públicas.
Sin embargo, en un entorno de producción real, como su implementación en servidores del Hospital Universitario San Ignacio, será imprescindible configurar un canal seguro (HTTPS) para proteger la integridad y confidencialidad de los datos transmitidos y cumplir con normativas institucionales y estándares de seguridad.
- **Estructura y control de mensajes:**
La comunicación entre la plataforma y las APIs externas se realiza mediante mensajes estructurados en formato JSON. Este formato permite representar datos de manera clara, jerárquica y fácilmente manipulable, lo que simplifica tanto el envío de información como el procesamiento de las respuestas por parte del sistema.
- **Gestión de autenticación y acceso:**
La plataforma cuenta con un sistema de autenticación robusto basado en JSON Web Tokens (JWT), mediante el cual se verifica la identidad del usuario y se controla el acceso a funcionalidades según su rol asignado (investigador o evaluador).
Además del uso de access tokens con expiración limitada, se ha implementado un mecanismo de refresh tokens, que permite extender la sesión de usuario de forma segura sin requerir un nuevo inicio de sesión, mejorando la experiencia de usuario sin comprometer la seguridad.
Si bien no se han incorporado mecanismos de protección contra tráfico malicioso o no autorizado a nivel de red (como firewalls, WAFs o inspección de capa 7), sí se han definido políticas de seguridad dentro de la lógica de la aplicación, tales como:
 - Validación de tokens en cada solicitud.
 - Protección de rutas sensibles mediante middlewares de autorización por rol.
 - Reglas de expiración y revocación de tokens para evitar accesos persistentes indebidos.

No obstante, se recomienda complementar esta lógica de aplicación con medidas adicionales en el entorno de producción, como filtrado de tráfico, segmentación de red y auditoría de accesos, de acuerdo con las políticas de seguridad institucionales.

Restricciones de memoria

El uso de la plataforma presenta diferentes consideraciones de memoria dependiendo del entorno en que se ejecute (cliente o servidor), así como de los componentes que la conforman.

- Memoria en el entorno del usuario (cliente)

En el caso de los usuarios finales, la plataforma es accesible a través de un navegador web moderno. Por tanto, las restricciones de memoria están determinadas por los límites establecidos por cada navegador y el sistema operativo del dispositivo. Estos límites, en sistemas operativos de 64 bits, se distribuyen típicamente de la siguiente manera:

- Google Chrome, Microsoft Edge, Opera: entre 1.4 GB y 2 GB por pestaña.
- Mozilla Firefox: entre 1.5 GB y 2 GB por pestaña.
- Safari: alrededor de 1.5 GB por pestaña.

Dado que la plataforma está optimizada para funcionar con eficiencia en el cliente, no se espera que la interfaz consuma más de 150 MB en condiciones normales, lo cual se encuentra ampliamente por debajo de estos límites.

- Memoria en el entorno del servidor

La plataforma será desplegada mediante Docker Compose, utilizando una arquitectura modular basada en contenedores, lo que facilita su mantenimiento y escalabilidad. Cada uno de los servicios principales opera de forma independiente en su propio contenedor:

- MinIO: sistema de almacenamiento de objetos para gestionar los documentos subidos y generados.
- MongoDB: base de datos NoSQL utilizada para almacenar la información estructurada de la aplicación.
- Backend: servidor que implementa la lógica de negocio utilizando Node.js y Express.js.
- Frontend: aplicación de interfaz de usuario desarrollada en React, servida estáticamente desde su contenedor correspondiente.

Con base en pruebas y estimaciones realizadas durante el desarrollo, el consumo aproximado de memoria RAM por cada servicio es el siguiente:

Servicio	Consumo estimado de RAM
MongoDB	400 – 600 MB
MinIO	300 – 500 MB
Backend	150 – 300 MB

Frontend	50 – 150 MB
Docker Desktop (solo en entornos locales)	500 MB – 2 GB

Tabla 5. Restricciones de memoria

Por lo tanto, el consumo total de memoria puede oscilar entre 1.5 GB y 3.5 GB, dependiendo de la carga operativa, el número de solicitudes concurrentes y el entorno específico en el que se despliegue.

Se recomienda que el servidor destinado a alojar la plataforma cuente con al menos 4 GB de memoria RAM disponibles exclusivamente para la solución, a fin de asegurar una operación fluida y dejar un margen adicional para el sistema operativo y otros procesos del entorno.

- **Requerimientos especiales de hardware**

No se establecen requerimientos especiales de hardware para el cliente o desarrolladores. La plataforma puede ejecutarse sin necesidad de tarjetas gráficas dedicadas, almacenamiento adicional o capacidades específicas de procesamiento, dado que todo el procesamiento intensivo, como la evaluación de documentos, se realiza a través de servicios externos de IA, y no localmente. La capacidad de memoria disponible fuera de los contenedores dependerá directamente de las especificaciones de cada dispositivo personal utilizado para ejecutar o probar la solución.

Finalmente, la aplicación está configurada para operar por defecto en el puerto HTTP estándar (80), lo que facilita el manejo eficiente de múltiples solicitudes sin necesidad de configuraciones avanzadas de red en el entorno de ejecución local.

Operaciones

Modos de operación según el rol del usuario:

La plataforma define distintos modos de operación en función del rol asignado a cada usuario, lo que determina su acceso a las funcionalidades del sistema. Esta diferenciación permite garantizar un flujo de trabajo organizado, seguro y adaptado a los procesos de evaluación ética.

Actualmente, los roles contemplados son los siguientes:

- **Evaluador:**
Es el rol con mayores permisos dentro del sistema. Se encarga de revisar los documentos de consentimiento informado cargados por los investigadores, haciendo uso de las herramientas de evaluación automática proporcionadas por la plataforma.

Las funciones principales de un evaluador incluyen:

- Cargar documentos de consentimiento informado.
- Ejecutar evaluaciones automáticas mediante el modelo de lenguaje integrado.

- Visualizar y analizar los resultados generados por el sistema.
- Descargar reportes con observaciones estructuradas.
- Enviar retroalimentación por correo electrónico al investigador.
- Acceder a estadísticas sobre los documentos evaluados.

Adicionalmente, al momento de inicializar la aplicación, se crea automáticamente un usuario raíz con rol de evaluador, mediante un archivo *seed*. Este usuario tiene como finalidad facilitar el acceso inicial al sistema en entornos de desarrollo o despliegue, y puede ser utilizado como punto de partida para la creación de otros usuarios evaluadores.

- Investigador:

Este rol está orientado a facilitar la generación de documentos que cumplan con los estándares éticos requeridos. Las funcionalidades disponibles para los investigadores son:

- Completar formularios con la información necesaria para la elaboración del consentimiento informado.
- Generar automáticamente un documento basado en la plantilla institucional.
- Consultar el historial de casos previamente creados.
- Visualizar observaciones o recomendaciones emitidas por los evaluadores.

Aunque todos los usuarios acceden a través de la misma aplicación web, la plataforma aplica un sistema de control de acceso basado en roles, lo que permite mostrar únicamente las funcionalidades correspondientes a cada tipo de usuario. Esto garantiza una experiencia personalizada, segura y alineada con las responsabilidades de cada perfil.

Periodos de actividad e inactividad:

Actualmente, la planificación de periodos de mantenimiento programado no forma parte del alcance funcional de este proyecto. Sin embargo, se reconoce que este es un aspecto relevante para garantizar la estabilidad y continuidad operativa de la plataforma en entornos reales.

Establecer ventanas de mantenimiento definidas permitirá, en futuras etapas de despliegue y operación, realizar tareas como actualizaciones, respaldos y ajustes técnicos sin afectar la disponibilidad del servicio ni la experiencia de los usuarios. Para lograrlo, se recomienda programar estas intervenciones en horarios de baja actividad, y complementar esta práctica con herramientas que permitan notificar automáticamente a los usuarios con suficiente antelación sobre las interrupciones planificadas.

Aunque esta funcionalidad no se implementará en la versión actual, dejar establecida esta recomendación sienta las bases para que, en etapas posteriores, la plataforma pueda evolucionar hacia un esquema de operación más estable, predecible y orientado a buenas prácticas de gestión de sistemas.

Procesos de recuperación:

En la versión actual del sistema, no se ha implementado ningún mecanismo automatizado de respaldo ni recuperación de datos. La plataforma opera con una instancia de MongoDB autoalojada, ejecutada en un contenedor Docker como parte del entorno definido por Docker Compose, sin configuraciones de replicación, respaldo automático o recuperación ante fallos.

No obstante, a continuación se presentan recomendaciones para futuras implementaciones, que permitirían fortalecer la integridad y disponibilidad de los datos, especialmente en entornos de producción.

- Despliegue local (actual, con mejoras futuras)

Aunque MongoDB se encuentra actualmente desplegado localmente, se recomienda que, en entornos reales, se adopten las siguientes estrategias:

- Respallos periódicos utilizando herramientas como:
 - mongodump: para generar copias de seguridad de las bases de datos o colecciones.
 - mongorestore: para restaurar datos a partir de un respaldo previamente generado.
- Automatización de respaldos mediante tareas programadas (por ejemplo, cron jobs) en el servidor donde se aloja la solución.
- Uso de volúmenes Docker persistentes, para asegurar que los datos no se pierdan al reiniciar o recrear los contenedores.
- Configuración de replica sets locales para habilitar redundancia de datos y tolerancia a fallos.

Estas estrategias no forman parte del alcance actual del proyecto, pero se recomienda su implementación por parte del equipo técnico que asuma el despliegue final.

- MongoDB Atlas

Aunque la solución no está actualmente integrada con servicios en la nube, se ha diseñado con la posibilidad de permitir, en el futuro, una migración a MongoDB Atlas, una plataforma de base de datos como servicio (DBaaS). Esta opción puede ser considerada por el Hospital Universitario San Ignacio u otra entidad responsable del despliegue, si se requieren mayores niveles de disponibilidad y robustez.

MongoDB Atlas ofrece ventajas como:

- Clusters con alta disponibilidad y réplicas automáticas.
- Respallos automáticos programables.
- Recuperación a punto en el tiempo.
- Escalabilidad gestionada y monitoreo integrado.

La transición a esta alternativa requeriría únicamente ajustes mínimos en la configuración de conexión de la base de datos, ya que la plataforma es compatible tanto con despliegues locales como en la nube.

- *Funciones de soporte y procesamiento de datos:*

La plataforma no realiza procesamiento intensivo de datos a nivel local. En su lugar, delega las tareas de análisis de documentos a modelos de lenguaje de gran escala (LLMs) mediante el consumo de APIs comerciales como las ofrecidas por Gemini (Google) y Groq.

El procesamiento es guiado por medio de técnicas de prompt engineering, lo que permite orientar el comportamiento del modelo para realizar evaluaciones éticas automatizadas sobre los documentos de consentimiento informado.

Dentro de la plataforma, el backend cumple funciones de soporte que facilitan esta integración, tales como:

- Construcción dinámica de prompts, ajustados al contenido del documento y los criterios normativos.
- Envío estructurado de solicitudes a las APIs externas.
- Recepción e interpretación de respuestas devueltas por los modelos en formato JSON.
- Presentación clara de los resultados a los evaluadores a través de la interfaz web.

Esto permite que la plataforma aproveche la capacidad de procesamiento de los modelos externos sin requerir infraestructura local de alto rendimiento, manteniendo así una experiencia fluida y eficiente para los usuarios.

Requerimientos de adaptación del sitio

La plataforma ha sido concebida como una aplicación web accesible desde navegadores modernos, lo que implica una serie de requerimientos de adaptación para asegurar su correcto funcionamiento en distintos entornos, dispositivos y contextos de uso. A continuación, se describen los principales aspectos considerados:

- **Compatibilidad con navegadores modernos:**
La aplicación está diseñada para ser compatible con los navegadores más utilizados, incluyendo Google Chrome, Mozilla Firefox, Microsoft Edge y Safari. Se contemplan pruebas de compatibilidad para garantizar una experiencia de usuario uniforme en todos ellos. No se garantiza compatibilidad con navegadores obsoletos o versiones no soportadas oficialmente.
- **Diseño responsivo:**
La interfaz de usuario ha sido desarrollada con principios de diseño responsivo, lo que permite su adaptación automática a diferentes tamaños y resoluciones de pantalla. Esto asegura una experiencia óptima tanto en computadores de escritorio, como en portátiles y dispositivos con pantallas medianas. No se contempla soporte específico para dispositivos móviles o pantallas táctiles en esta versión.
- **Configuración regional y localización:**
La plataforma ha sido configurada para operar principalmente en idioma español, con soporte para formatos regionales como fechas, horas y numeración acorde al contexto colombiano. Actualmente, no se implementa soporte multilingüe.
- **Seguridad y control de acceso por roles:**
La plataforma implementa un sistema de autenticación basado en JSON Web Tokens (JWT) y control de acceso por roles, actualmente definidos como evaluador e investigador. No existe un rol de administrador como tal; sin embargo, al iniciar el sistema se crea un usuario raíz evaluador, preconfigurado mediante archivos de seed.
Si bien no se han aplicado mecanismos de cifrado de datos en tránsito (como HTTPS) dentro del entorno de desarrollo, se establece como requisito esencial para el despliegue en producción que el sistema sea configurado detrás de un servidor web con certificados SSL válidos, en cumplimiento con las políticas de seguridad del Hospital Universitario San

Ignacio.

- Accesibilidad y buenas prácticas de UI/UX:
La interfaz ha sido diseñada para ser clara, legible y funcional para usuarios con distintos niveles de habilidad tecnológica. Se recomienda, en etapas posteriores, evaluar el cumplimiento de estándares de accesibilidad web (WCAG) si la plataforma se amplía a una población más amplia o requiere certificación institucional.

Funciones del producto

La plataforma ha sido desarrollada para asistir a dos tipos de usuarios principales: investigadores y evaluadores éticos, en el proceso de creación y evaluación de documentos de consentimiento informado. Aunque ambos interactúan con la misma aplicación web, sus flujos de trabajo son completamente independientes y no se cruzan dentro del sistema.

- Funcionalidades para investigadores

Los investigadores utilizan la plataforma para crear nuevos casos, completando un formulario con toda la información requerida por el Comité de Ética del Hospital Universitario San Ignacio. Este formulario permite ingresar datos como la versión del documento, fecha, introducción, objetivos, descripción del estudio, procedimientos, riesgos, beneficios, aspectos de confidencialidad y autorización del participante.

Una vez completado, el sistema genera un documento PDF con el consentimiento informado estructurado según la plantilla institucional. El investigador tiene la opción de descargar este documento y se almacena en su historial de casos personales dentro de la plataforma.

Posteriormente, el investigador debe cargar este documento en la plataforma Fundanet, utilizada por el Hospital Universitario San Ignacio para la gestión de proyectos de investigación y evaluación ética. Fundanet permite automatizar los procesos de registro de proyectos de investigación para la posterior evaluación por parte del comité de ética, facilitando el acceso seguro a documentos en función del perfil del usuario y garantizando el cumplimiento de la legalidad .

- Funcionalidades para evaluadores

Los evaluadores no acceden a los documentos generados por los investigadores directamente desde la plataforma. En su lugar, acceden a los proyectos que les asigna la plataforma institucional Fundanet, desde donde descargan el consentimiento informado previamente cargado por el investigador.

Luego, el evaluador accede a nuestra plataforma para cargar ese documento PDF y realizar una evaluación asistida por inteligencia artificial (IA). El sistema analiza el contenido del documento utilizando modelos de lenguaje de gran escala (LLMs) mediante técnicas de *prompt engineering*, y devuelve:

- Una clasificación por criterio ético (cumple / no cumple).

- Fragmentos relevantes del documento que justifican la decisión.
- Una explicación en lenguaje natural que detalla por qué un criterio no cumple, si aplica.
- La posibilidad de editar la evaluación realizada por el modelo en caso de no estar de acuerdo con ella.

Estos resultados permiten al evaluador contar con un insumo estructurado y ágil para tomar decisiones, sin reemplazar su criterio profesional.

Adicionalmente, desde la misma interfaz, el evaluador puede enviar los resultados por correo electrónico:

- Al investigador, en caso de que el consentimiento requiera correcciones.
- Al Comité de Ética, si el documento ha sido aprobado y está listo para su validación.

- Configuración y seguimiento

La plataforma permite configurar los prompts éticos utilizados por el modelo, lo cual otorga flexibilidad para adaptar el análisis a cambios normativos o criterios nuevos sin necesidad de modificar el código fuente.

Además, los evaluadores cuentan con un panel estadístico que muestra indicadores clave, tales como:

- Total de evaluaciones realizadas.
- Total de documentos aprobados y rechazados.
- Tasa de devolución de documentos.
- Tiempo promedio de evaluación.
- Comparación de métricas con respecto al mes anterior.

Estas funcionalidades hacen de la plataforma una herramienta modular, flexible y enfocada en mejorar la productividad, al tiempo que asegura la trazabilidad y transparencia en el proceso de evaluación ética. Su diseño busca facilitar el trabajo de los evaluadores, reducir el esfuerzo manual y mantener alineadas las decisiones con los lineamientos institucionales del Comité de Ética.

A continuación, se presenta un diagrama de casos de uso para mejorar el entendimiento de estas funcionalidades en la plataforma:

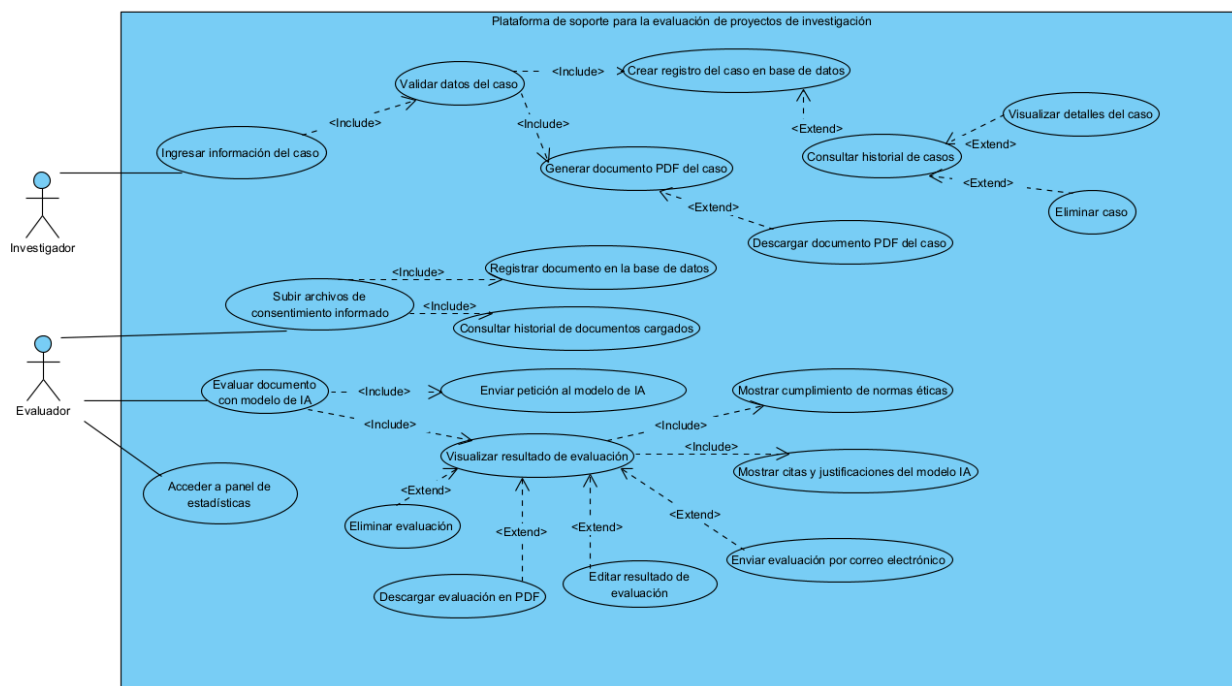


Figura 1. Diagrama de casos de uso del sistema (ver anexo 1)

Características del usuario

Características del Usuario	Descripción
Nivel de Seguridad o de Privilegios	<p>Investigador: Acceso restringido únicamente al formulario para generar documentos de consentimiento informado, consultar su historial de casos y descargar los archivos generados. No tiene visibilidad sobre los casos evaluados ni acceso a los módulos de evaluación.</p> <p>Evaluador: Acceso a herramientas de evaluación automática mediante IA, carga de documentos descargados desde Fundanet, configuración de prompts, visualización de resultados y estadísticas, además del envío de reportes por correo. No puede visualizar información de los investigadores ni sus formularios directamente.</p>
Roles	<p>Investigador: Usuario que crea casos mediante formularios y genera el documento de consentimiento informado para posteriormente cargarlo en Fundanet.</p> <p>Evaluador: Miembro del comité de ética que evalúa documentos descargados desde Fundanet mediante la plataforma, con apoyo de IA, y toma decisiones sobre su cumplimiento ético.</p>
Nivel de Estudios o Experiencia Técnica	<p>Investigador: No se requiere experiencia técnica avanzada. Basta con saber manejar plataformas web y llenar formularios en línea.</p> <p>Evaluador: Requiere familiaridad con herramientas digitales y formación previa en bioética o evaluación ética para interpretar los resultados generados por la IA.</p>

Frecuencia de Uso	<p>Investigador: Uso ocasional, según el número de proyectos que deba presentar para evaluación ante el comité de ética.</p> <p>Evaluador: Uso frecuente, condicionado por la cantidad de documentos asignados para revisión en cada ciclo de evaluación institucional.</p>
-------------------	---

Tabla 6. Tabla de características de usuario

Restricciones

El desarrollo de la plataforma web se encuentra sujeto a diversas restricciones que condicionan decisiones de diseño, implementación y despliegue. A continuación, se detallan las principales:

- Ausencia de tolerancia a fallos:

Actualmente, la plataforma no cuenta con mecanismos de tolerancia a fallos. La arquitectura está compuesta por servicios en contenedores (frontend, backend, base de datos y almacenamiento), pero no se ha configurado redundancia ni reinicio automático ante fallos. Por lo tanto, si alguno de los servicios críticos —como el backend— se detiene, la plataforma deja de estar operativa hasta que el servicio sea restablecido manualmente.

- Consideraciones de seguridad:

Dado que la plataforma maneja documentos de consentimiento informado, se han implementado medidas básicas de seguridad, como control de acceso por roles y autenticación basada en JWT. Sin embargo, no se ha configurado cifrado HTTPS en el entorno de desarrollo, por lo que se establece como requisito para producción que las comunicaciones se aseguren mediante certificados válidos. Tampoco se integran controles de seguridad avanzados como firewalls.

- Limitaciones de hardware y procesamiento:

El sistema ha sido diseñado para operar en dispositivos personales y servidores sin requerimientos de hardware especializado. No se entrena ni ajusta localmente ningún modelo de lenguaje, ya que todo el procesamiento intensivo se realiza a través de APIs externas (como Gemini y Groq). Inicialmente se consideró el uso de recursos del Centro de Investigación (CINE), pero no fue necesario implementarlo.

- Independencia del modelo de IA:

La plataforma no aloja ni modifica directamente ningún modelo de IA. Toda la interacción se realiza mediante prompt engineering sobre modelos accesibles a través de APIs externas, lo cual implica una dependencia directa de la disponibilidad y funcionamiento de dichos servicios.

- Políticas institucionales:

El desarrollo y uso de la plataforma debe alinearse con las políticas internas del Hospital Universitario San Ignacio, particularmente en lo relacionado con el manejo de información sensible, confidencialidad, y buenas prácticas en la evaluación ética de proyectos.

- Idioma de operación:

La plataforma está diseñada para operar en idioma español, acorde al contexto institucional. No se contempla soporte multilingüe en esta versión.

- Protocolo de comunicación:

Las comunicaciones entre componentes internos de la plataforma se realizan mediante el protocolo TCP/IP, utilizando el puerto estándar HTTP 80 para simplificar el acceso en entornos locales. Se recomienda usar HTTPS en producción

- Normativas éticas:

Todas las funcionalidades del sistema están diseñadas para facilitar el cumplimiento de las normas éticas internacionales y locales que regulan la investigación con participación humana.

- Licenciamiento y sostenibilidad:

Se ha priorizado el uso de tecnologías y herramientas de código abierto o con licencias gratuitas, para garantizar la viabilidad económica del proyecto y su continuidad.

- Lenguajes y tecnologías:

El desarrollo se ha realizado utilizando TypeScript, con Node.js y Express.js en el backend, y React con Vite en el frontend. La infraestructura se gestiona mediante contenedores Docker orquestados con Docker Compose.

- Integración con servicios externos:

El procesamiento de lenguaje natural no se realiza localmente. Se integran APIs de terceros para acceder a modelos de lenguaje (LLMs), como los provistos por Gemini y Groq, para realizar evaluaciones éticas automáticas mediante prompt engineering.

- Flexibilidad en la infraestructura:

La plataforma ha sido diseñada con una arquitectura modular y contenedorizada, lo cual permite su despliegue en distintos entornos con recursos variables. Sin embargo, su correcto funcionamiento dependerá de la disponibilidad de recursos mínimos, como al menos 4 GB de memoria RAM.

Modelo del dominio

El modelo del dominio representa la estructura conceptual fundamental de la plataforma, describiendo de manera clara las entidades principales, sus atributos y relaciones. Este modelo actúa como una base de entendimiento compartido entre los miembros del equipo de desarrollo y los usuarios involucrados en el proyecto, facilitando la toma de decisiones de diseño y la validación de requerimientos funcionales.

Definiciones:

El modelo del dominio se compone de los siguientes elementos esenciales:

- Entidades del negocio: Representan los elementos clave del sistema, como los usuarios, los casos generados por los investigadores, las evaluaciones realizadas por los evaluadores, los prompts configurables y las normas éticas que guían el análisis automatizado.
- Relaciones: Describen cómo interactúan estas entidades entre sí. Por ejemplo, un usuario puede tener el rol de investigador y generar documentos mediante formularios, o ser evaluador y cargar archivos para ser analizados por la IA. Las evaluaciones se vinculan con prompts configurados previamente y contienen criterios éticos que se derivan del análisis del documento.
- Atributos: Cada entidad está compuesta por un conjunto de atributos que definen su estructura y propósito. Por ejemplo, una evaluación contiene atributos como fechas, estado, archivo evaluado, resultado, entre otros. Un usuario tiene nombre, correo, tipo de rol, y credenciales de acceso.

Estado Actual:

Este modelo refleja la estructura definitiva del sistema planteado, resultado del análisis de requerimientos y del diseño funcional. A partir de este modelo se construyen las funcionalidades de la plataforma, por lo cual representa una base sólida para su implementación y evolución.

Documentación

La documentación de cada elemento del modelo sigue el formato descrito a continuación:

ID	Elemento del dominio	Descripción	Atributos	Tipos de dato
1.	Usuario	Representa a los usuarios del sistema, ya sean evaluadores o investigadores.	- <i>_id</i> - <i>name</i> - <i>last_name</i> - <i>email</i> - <i>password</i> - <i>type</i> - <i>model</i> - <i>provider</i>	- <i>ObjectId</i> - <i>String</i>
2.	Caso	Documento creado por el investigador que contiene la información del consentimiento informado.	- <i>_id</i> - <i>uid</i> - <i>project_name</i> - <i>date</i> - <i>version</i> - <i>code (opcional)</i> - <i>pdf</i>	- <i>ObjectId</i> - <i>String</i> - <i>Date</i> - <i>Number</i>
3.	Evaluación	Proceso en el que el evaluador analiza un documento utilizando IA.	- <i>_id</i> - <i>uid</i> - <i>id_fundanet</i> - <i>file</i> - <i>status</i> - <i>error_type</i> - <i>approved</i> - <i>student_email</i>	- <i>ObjectId</i> - <i>String</i> - <i>Boolean</i> - <i>ENUM</i> - <i>Date</i> - <i>Number</i>

			- <i>version</i> - <i>start_date</i> - <i>end_date</i>	
4.	Prompt	Configuraciones personalizables que guían al modelo de IA durante la evaluación.	- <i>_id</i> - <i>uid</i> - <i>name</i> - <i>text</i> - <i>code</i> - <i>description</i>	- <i>ObjectId</i> - <i>String</i>
5.	Norma ética	Representa cada uno de los criterios normativos aplicados en una evaluación.	- <i>_id</i> - <i>evaluationId</i> - <i>description</i> - <i>status</i> - <i>justification</i> - <i>quote</i> - <i>codeNumber</i>	- <i>ObjectId</i> - <i>String</i> - <i>ENUM</i>

Tabla 7. Elementos del modelo de dominio

A continuación se presenta el modelo de datos:

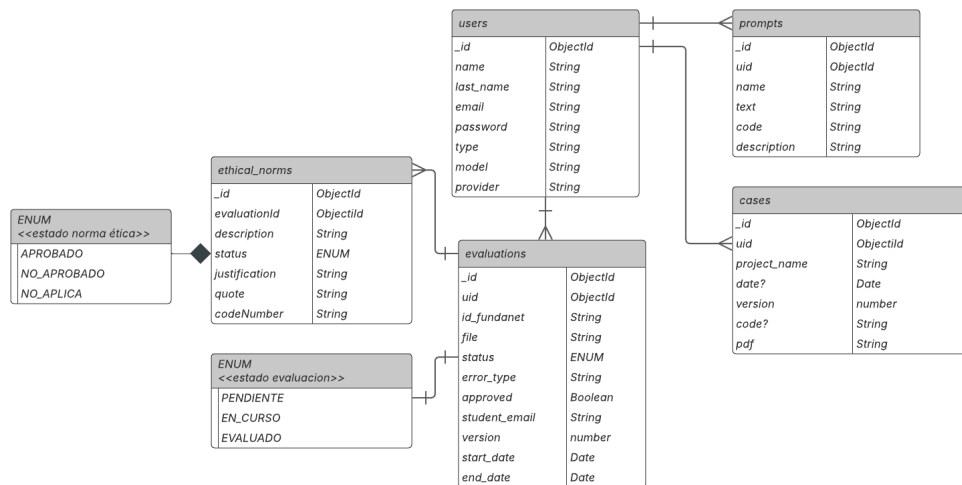


Figura 2. Modelo de datos del sistema (ver anexo 2)

Suposiciones y dependencias

Este apartado presenta una lista de factores asumidos como ciertos durante el desarrollo del proyecto. Estas suposiciones afectan directamente los requerimientos funcionales y no funcionales establecidos, y cualquier cambio en ellas podría impactar negativamente la implementación, funcionalidad o despliegue de la plataforma.

Suposiciones:

- Conectividad a internet estable: Se asume que tanto el equipo de desarrollo como los usuarios finales disponen de una conexión a internet constante y confiable. Esto es esencial para que la

plataforma pueda interactuar con las APIs externas de modelos de lenguaje (como Gemini y Groq), las cuales son fundamentales para la evaluación asistida por inteligencia artificial.

- Disponibilidad de recursos locales suficientes: Se da por sentado que los equipos personales utilizados durante el desarrollo cuentan con los recursos mínimos necesarios (CPU, RAM, espacio en disco) para ejecutar la plataforma en contenedores Docker, sin requerir GPU ni procesamiento intensivo.
- Compatibilidad de herramientas y tecnologías: Se supone que las herramientas seleccionadas (como Node.js, Bun, Docker, MongoDB y React) funcionarán correctamente en los entornos previstos (Linux/macOS/Windows), sin problemas de compatibilidad que afecten el despliegue o el desarrollo.
- Funcionamiento adecuado de servicios externos (APIs de IA): El análisis automatizado se apoya completamente en APIs externas (Gemini, Groq). Se asume que estos servicios estarán disponibles de forma continua y sin interrupciones críticas. En caso de cambios en sus políticas, estructura de respuestas o costos, el proyecto podría requerir adaptaciones.
- Uso aislado del sistema por parte de cada rol: Se asume que los usuarios evaluadores y los investigadores interactúan con la plataforma de forma separada, sin necesidad de sincronización directa o intercambio de datos entre ellos dentro del sistema. El flujo de trabajo depende de que los documentos generados por los investigadores sean cargados a plataformas externas (como Fundanet), y luego recuperados por los evaluadores para su análisis.
- No intervención en los flujos institucionales existentes (Fundanet): Se considera que el hospital mantendrá su flujo habitual mediante Fundanet, y que nuestra plataforma no reemplazará ni modificará esa infraestructura, sino que operará como una herramienta de soporte autónoma.

Dependencias:

- Disponibilidad y estabilidad de la red institucional: La plataforma requiere conexión constante a internet para acceder a las APIs de modelos de lenguaje (Gemini, Groq). Una baja velocidad o intermitencia en la red afectaría directamente los tiempos de respuesta y la experiencia del usuario al realizar evaluaciones automáticas.
- Funcionamiento de servicios externos (APIs de LLMs): El análisis automatizado depende completamente de APIs de terceros. Cambios en los endpoints, formatos de respuesta, políticas de uso, costos o tiempos de respuesta de estos proveedores podrían impactar la operatividad del sistema o requerir ajustes en el código.
- Estabilidad de la infraestructura del servidor local: Dado que la solución se ejecuta sobre contenedores Docker, su correcto funcionamiento depende de que el servidor local (o máquina donde se despliegue) mantenga disponibilidad, recursos adecuados (memoria, CPU)

y estabilidad en los servicios que la componen (MongoDB, MinIO, backend, frontend).

- Cumplimiento de políticas y normativas institucionales: La plataforma debe ajustarse a las normas de seguridad, privacidad y ética definidas por el Hospital Universitario San Ignacio. Cualquier modificación en estas directrices puede requerir cambios en la lógica del sistema, los prompts de evaluación o la estructura de almacenamiento.
- Interoperabilidad con el flujo institucional (Fundanet): Aunque la plataforma funciona de forma autónoma, su adopción depende de la continuidad del flujo institucional que ocurre en Fundanet. Cualquier cambio sustancial en la forma en que los investigadores o evaluadores interactúan con Fundanet podría generar barreras de adopción o requerir una integración más profunda.
- Capacitación del personal evaluador: La plataforma requiere que los evaluadores estén dispuestos y capacitados para interpretar los resultados generados por la IA, gestionar sus propios prompts y operar la herramienta sin acompañamiento técnico permanente. La falta de capacitación o resistencia al cambio limitaría su adopción efectiva.

Distribución de requerimientos

La siguiente tabla resume la distribución de funcionalidades principales según los dos roles definidos en la plataforma: Investigador y Evaluador. Cada funcionalidad ha sido asignada con base en las responsabilidades reales de los usuarios dentro del flujo de trabajo del sistema.

Esta asignación refleja que los investigadores están encargados de crear y cargar documentos de consentimiento informado, mientras que los evaluadores gestionan la evaluación ética de dichos documentos, desde su análisis hasta la emisión y eventual retroalimentación de resultados. Esta separación clara garantiza el cumplimiento de las normas institucionales y facilita la trazabilidad del proceso evaluativo.

La tabla asegura que los requerimientos funcionales estén alineados con los niveles de acceso y operaciones específicas de cada rol, evitando superposición de responsabilidades y mejorando la seguridad del sistema.

Módulo / Funcionalidad	Investigador	Evaluador
Registrarse en el sistema	✓	
Iniciar sesión	✓	✓
Cambiar contraseña	✓	✓
Cargar documento	✓ (generar FCI)	✓ (para evaluar)

Generar documento PDF (Formulario FCI)	✓	
Ver historial de documentos propios	✓	
Descargar documento generado	✓	✓ (resultado evaluación IA)
Cargar consentimiento informado para evaluación		✓
Generar evaluación con IA		✓
Visualizar resultados de la evaluación		✓
Editar resultado de evaluación (estado/justificación)		✓
Enviar resultados por correo electrónico		✓
Ver historial de evaluaciones		✓
Consultar estadísticas		✓
Modificar prompts de evaluación		✓

Tabla 8. Asignación de funcionalidades de acuerdo al usuario

3. REQUERIMIENTOS ESPECÍFICOS

Requerimientos de interfaces externas

Interfaces con el usuario

En este proyecto se ha implementado una única interfaz basada en una aplicación web, que opera como el punto principal de interacción entre el sistema y sus usuarios: investigadores y evaluadores. A través de esta plataforma, los usuarios pueden ingresar datos, consultar información, cargar documentos y ejecutar acciones clave como iniciar una evaluación o enviar resultados.

La interfaz está diseñada bajo principios de usabilidad, accesibilidad y consistencia visual, con base en un enfoque modular inspirado en Atomic Design. Ya se han definido los mockups que establecen la estructura visual, jerarquía de información y distribución de componentes, con el fin de optimizar la experiencia del usuario. Funcionalidades adicionales como accesos rápidos por teclado podrían ser consideradas en futuras versiones.

A continuación, se describen los dispositivos y componentes necesarios para el correcto funcionamiento de la aplicación y su interacción con el usuario:

- Pantalla: Permite la visualización de todas las interfaces de la aplicación web, facilitando la interacción con los distintos módulos del sistema.
- Teclado: Se utilizará para el ingreso de datos y la navegación a través de accesos rápidos, mejorando la eficiencia en el uso del sistema.
- Mouse: Facilita la selección y navegación entre las opciones presentadas en la interfaz gráfica de usuario (GUI).
- Interfaz GUI: La interfaz gráfica de usuario mostrará todas las funcionalidades del sistema de forma visual e intuitiva, adaptada a las necesidades del usuario.
- Tarjeta de Red: Permite la conexión a internet necesaria para acceder a la aplicación web y para el envío y recepción de datos en tiempo real.

# Requerimiento	Tipo de Requerimiento	Casos de Uso Asociados
1	No Funcional - Requerimiento de Interfaz	Ingresar información del caso
Descripción	El sistema debe contar con una interfaz gráfica accesible a través de navegador web, que permita a los usuarios autenticados interactuar con los distintos módulos del sistema de forma intuitiva y eficiente.	Subir archivos de consentimiento o informado
Razón	Facilitar una interacción eficiente y accesible con el sistema para todos los usuarios finales.	Evaluar documento con modelo de IA
Autor	Equipo de Desarrollo	Acceder a panel de estadísticas
Criterio de medición	Verificación de la accesibilidad y funcionalidad de la GUI en la aplicación web.	Visualizar resultado de evaluación
Prioridad	Alta	Enviar evaluación por correo electrónico
Módulo Asociado	Módulo de Interfaz de Usuario	
Versión	1.0	
Fecha	Noviembre 2024	

# Requerimiento	Tipo de Requerimiento	Casos de Uso Asociados
2	No funcional – Requerimiento técnico de conectividad y visualización	Todos los casos de uso del sistema
Descripción	La plataforma debe ser operativa en dispositivos que cuenten con navegador web, acceso a internet y componentes básicos como pantalla, teclado y mouse. La comunicación con el sistema se realizará mediante solicitudes HTTP estándar.	
Razón	Garantizar que los usuarios puedan interactuar correctamente con el sistema a través de los dispositivos adecuados.	
Autor	Equipo de Desarrollo	
Criterio de medición	Pruebas de compatibilidad con dispositivos típicos de escritorio.	
Prioridad	Alta	
Módulo Asociado	Módulo de Interacción de Usuario	
Versión	1.0	
Fecha	Noviembre 2024	

# Requerimiento	Tipo de Requerimiento	Casos de Uso Asociados
3	No funcional – Usabilidad	Ingresar información del caso
Descripción	La interfaz debe ser intuitiva, clara y coherente. Se deben aplicar buenas prácticas como agrupación lógica de secciones, retroalimentación visual y uso de íconos comprensibles.	Subir archivos de consentimiento informado
Razón	Reducir la curva de aprendizaje de los usuarios y aumentar la eficiencia en la navegación y uso del sistema.	Evaluar documento con modelo de IA
Autor	Equipo de Desarrollo	Acceder a panel de estadísticas

Criterio de medición	Pruebas de usuario enfocadas en eficiencia, error y satisfacción.	Visualizar resultado de evaluación
Prioridad	Alta	Enviar evaluación por correo electrónico
Módulo Asociado	Módulo de Experiencia de Usuario	
Versión	1.0	
Fecha	Abril 2025	

# Requerimiento	Tipo de Requerimiento	Casos de Uso Asociados
4	No funcional – Requerimiento de compatibilidad	Todos los casos de uso accesibles desde navegador
Descripción	La aplicación web debe ser compatible con los navegadores modernos más utilizados, como Google Chrome, Microsoft Edge y Firefox, en sus versiones recientes.	
Razón	Asegurar la accesibilidad del sistema desde diversos entornos.	
Autor	Equipo de Desarrollo	-
Criterio de medición	Pruebas de visualización y funcionalidad en cada navegador.	-
Prioridad	Alta	-
Módulo Asociado	Interfaz Web	-
Versión	1.0	-
Fecha	Abril 2025	-

# Requerimiento	Tipo de Requerimiento	Casos de Uso Asociados
5	No funcional – Requerimiento de adaptabilidad	Visualizar detalles del caso
Descripción	La interfaz debe ajustarse correctamente a diferentes resoluciones de pantalla, incluyendo laptops y monitores estándar de escritorio.	Consultar historial de documentos
Razón	Facilitar la visualización y operación del sistema en distintos entornos de trabajo.	Visualizar resultado de evaluación
Autor	Equipo de Desarrollo	Ver panel de estadísticas
Criterio de medición	Evaluación de diseño y comportamiento visual en dispositivos con diferentes resoluciones.	
Prioridad	Media	
Módulo Asociado	Interfaz Web	
Versión	1.0	
Fecha	Abril 2025	

A continuación, se adjuntan los mockups aprobados por el cliente final, los cuales representan una visión general del diseño propuesto para la plataforma web. Estas maquetas ilustran las principales pantallas con las que interactuarán los dos tipos de usuarios del sistema: investigadores y evaluadores.

Los diseños incluyen funcionalidades clave como:

- La creación de nuevos casos de evaluación.
- La carga y visualización de documentos.
- La ejecución de evaluaciones automáticas mediante inteligencia artificial.
- La consulta y gestión de resultados.

Cada interfaz fue concebida priorizando la claridad visual, la facilidad de uso y una experiencia de usuario coherente, con una disposición ordenada de elementos, botones de acción estratégicamente ubicados, tablas segmentadas por estado y una navegación fluida entre secciones.

Estos mockups sirven como referencia visual y funcional del sistema final, y constituyen la base sobre la cual se desarrollará la interfaz definitiva. (Ver anexo 3)

Interfaces con el hardware

# Requerimiento	Tipo de Requerimiento	Casos de Uso Asociados
5	No funcional – Requerimiento de Hardware / Comunicación	Ingresar información del caso
Descripción	La plataforma debe comunicarse utilizando el protocolo TCP/IP a través del puerto estándar HTTP 80, facilitando el acceso vía navegador sin requerir configuraciones de red adicionales.	Subir archivos de consentimiento informado
Razón	Garantizar la eficiencia y accesibilidad de la comunicación entre los componentes de la plataforma en una red web estándar.	Evaluar documento con modelo de IA
Autor	Equipo de Desarrollo	Enviar evaluación por correo electrónico
Criterio de medición	Verificación de conectividad mediante solicitudes HTTP exitosas al puerto 80.	Acceder a panel de estadísticas
Prioridad	Alta	
Módulo Asociado	Módulo de Comunicación	
Versión	1.0	
Fecha	Noviembre 2024	

# Requerimiento	Tipo de Requerimiento	Casos de Uso Asociados
6	No funcional – Requerimiento de infraestructura externa	Evaluar documento con modelo de IA
Descripción	El sistema se apoya en APIs externas (como Groq y Gemini) que alojan modelos de lenguaje de gran escala (LLM), eliminando la necesidad de recursos de cómputo propios como GPUs o servidores especializados.	
Razón	Optimizar la comunicación con el modelo LLM y asegurar capacidad computacional necesaria sin depender de hardware propio.	
Autor	Equipo de Desarrollo	

Criterio de medición	Pruebas exitosas de evaluación automática utilizando modelos vía API.	
Prioridad	Media	
Módulo Asociado	Módulo de Procesamiento de LLM	
Versión	1.0	
Fecha	Noviembre 2024	

# Requerimiento	Tipo de Requerimiento	Casos de Uso Asociados
7	No Funcional - Requerimiento de Hardware	Ingresa información del caso
Descripción	La plataforma será accesible y ejecutable en dispositivos de cómputo personales de los desarrolladores sin requerimientos de hardware adicional.	Subir archivos de consentimiento informado
Razón	Facilitar el desarrollo y pruebas en entornos de trabajo accesibles y prácticos.	Consultar historial de documentos
Autor	Equipo de Desarrollo	Evaluar documento con modelo de IA
Criterio de medición	Ejecución exitosa de la plataforma en dispositivos personales sin incidencias.	Acceder a panel de estadísticas
Prioridad	Baja	Enviar evaluación por correo electrónico
Módulo Asociado	Módulo de Desarrollo	
Versión	1.0	
Fecha	Noviembre 2024	

Interfaces con el software

Para el correcto funcionamiento de la plataforma, se requiere la integración con diversos productos de software. A continuación, se describen las interfaces y requerimientos mínimos, los cuales han sido validados en el entorno de desarrollo actual. Estas especificaciones podrían ajustarse en futuras etapas según las necesidades de despliegue institucional.

- Sistema Operativo: La plataforma se ejecuta completamente dentro de contenedores Docker, por lo que es independiente del sistema operativo del host. Puede ser desplegada en entornos compatibles con Docker, como sistemas basados en Linux o Windows con Docker Desktop instalado.
- Base de Datos NoSQL: Se utiliza MongoDB, una base de datos NoSQL ideal para manejar datos semi estructurados como documentos de consentimiento informado y sus evaluaciones. MongoDB se ejecuta en su propio contenedor dentro del entorno Docker Compose.
- Lenguajes y Frameworks
 - Backend: Desarrollado con Node.js y Express.js, estructurado bajo Clean Architecture, y ejecutado con Bun como entorno de ejecución y gestor de paquetes.
 - Frontend: Construido con React y TypeScript, usando Atomic Design como principio de diseño y Vite para la experiencia de desarrollo. Se integran bibliotecas como shadcn/ui y TanStack para la construcción eficiente de componentes UI.
- Modelos de Lenguaje y Evaluación Automática: La plataforma integra APIs de modelos de lenguaje de gran escala (LLM) como Gemini y Groq, permitiendo realizar evaluaciones automáticas de documentos mediante técnicas de prompt engineering. No se entrena ningún modelo localmente ni se utilizan técnicas tradicionales de NLP.
- Almacenamiento de Archivos: Para el almacenamiento de documentos, se utiliza MinIO, un sistema de almacenamiento de objetos auto alojado compatible con el API de S3. MinIO se ejecuta en un contenedor separado y permite almacenar de forma eficiente los archivos cargados y generados por los usuarios.
- Contenedores y Virtualización: Todos los servicios se gestionan mediante Docker Compose, lo cual permite ejecutar en contenedores independientes el backend, frontend, base de datos MongoDB y servicio de almacenamiento MinIO, garantizando portabilidad, modularidad y facilidad de despliegue.

Interfaces de comunicaciones

Los requerimientos identificados por el equipo de desarrollo para las interfaces de comunicación son los siguientes:

# Requerimiento	Tipo de Requerimiento	Casos de Uso Asociados
8	No funcional – Interoperabilidad	Subir archivos de consentimiento informado
Descripción	Toda la comunicación entre los componentes del sistema (frontend, backend, base de datos y servicios externos) debe realizarse utilizando el protocolo TCP/IP sobre el puerto HTTP 80.	Ver dashboard de estadísticas
Razón	Garantizar una transmisión de datos confiable y compatible en internet.	Enviar evaluación por correo electrónico

Autor	Equipo de Desarrollo	
Criterio de medición	Comunicación estable mediante solicitudes HTTP.	
Prioridad	Crítica	
Módulo Asociado	Módulo de Comunicación	
Versión	1.0	
Fecha	Noviembre 2024	

# Requerimiento	Tipo de Requerimiento	Casos de Uso Asociados
9	No Funcional - Seguridad	Configurar prompts de evaluación
Descripción	Todas las comunicaciones con APIs externas (como Groq o Gemini) deben realizarse mediante el protocolo HTTPS para proteger la integridad y confidencialidad de los datos. Aunque actualmente no se implementa HTTPS para la interfaz local, esta será una medida obligatoria en entornos de producción.	Enviar evaluación por correo electrónico
Razón	Proteger la integridad y confidencialidad de los datos transmitidos.	
Autor	Equipo de Desarrollo	
Criterio de medición	Verificación de uso de HTTPS en todas las llamadas salientes.	
Prioridad	Crítica	
Módulo Asociado	Módulo de Comunicación Segura	
Versión	1.0	
Fecha	Noviembre 2024	

# Requerimiento	Tipo de Requerimiento	Casos de Uso Asociados
-----------------	-----------------------	------------------------

10	No funcional – Interoperabilidad	Configurar prompts de evaluación
Descripción	Los datos intercambiados entre la plataforma y las APIs externas deben estar estructurados en formato JSON estándar.	Evaluar documento con modelo de IA
Razón	Facilitar la manipulación y lectura de datos.	
Autor	Equipo de Desarrollo	
Criterio de medición	Inspección de solicitudes y respuestas en formato JSON.	
Prioridad	Alta	
Módulo Asociado	Módulo de Intercambio de Datos	
Versión	1.0	
Fecha	Noviembre 2024	

# Requerimiento	Tipo de Requerimiento	Casos de Uso Asociados
11	Funcional - Seguridad	Configurar prompts de evaluación
Descripción	La plataforma debe autenticar todas las interacciones con sus servicios internos mediante tokens JWT, asegurando que solo usuarios autorizados puedan acceder a funcionalidades protegidas.	Enviar evaluación por correo electrónico
Razón	Asegurar que solo usuarios autorizados puedan acceder a la API.	
Autor	Equipo de Desarrollo	
Criterio de medición	Verificación de tokens válidos en cada solicitud.	
Prioridad	Crítica	
Módulo Asociado	Módulo de Seguridad	
Versión	1.0	
Fecha	Noviembre 2024	

# Requerimiento	Tipo de Requerimiento	Casos de Uso Asociados
12	No Funcional - Seguridad	Subir archivos de consentimiento informado
Descripción	La plataforma aplica políticas de control de acceso a nivel de aplicación mediante validación de sesión y roles. No se han implementado controles a nivel de red (como firewalls), pero se recomienda su inclusión en entornos de producción.	Enviar evaluación por correo electrónico
Razón	Asegurar la integridad del sistema y restringir el acceso solo a usuarios autorizados.	
Autor	Equipo de Desarrollo	
Criterio de medición	Validación de rutas protegidas por autenticación y roles.	
Prioridad	Alta	
Módulo Asociado	Módulo de Control de Acceso	
Versión	1.0	
Fecha	Noviembre 2024	

Características del producto de software

En esta sección se agrupan todos los requerimientos funcionales del sistema y se los relaciona con los módulos o funcionalidades que utilizan los dos roles principales de la plataforma: Investigador y Evaluador.

Cada módulo cuenta con una tabla estructurada que documenta, uno por uno, los requisitos asociados a su funcionalidad. Esta organización permite una lectura clara, trazabilidad completa y facilita su validación e implementación.

Además, como complemento a esta sección, se adjunta como anexo el documento de requerimientos funcionales detallados en formato Excel (Anexo 4), el cual contiene el listado completo de requerimientos (con su redacción oficial, clasificaciones, criterios de aceptación, prioridad y módulo asociado), facilitando su consulta, mantenimiento y verificación.

Cómo leer las tablas:

- ID: Identificador único del requerimiento (ej. RF00x).
- Elemento del dominio: Acción o entidad del sistema a la que se refiere el requerimiento (ej. “Iniciar sesión”, “Cargar archivo”).

- Descripción: Enunciado con la fórmula "El sistema debe...", que detalla lo que se espera que haga el sistema.
- Atributos: Se incluye solo si el requerimiento modifica o crea datos persistentes; describe campos y estructuras involucradas.
- Objetivo: Resume en una línea el propósito del requerimiento, facilitando la comprensión de su valor funcional o de negocio.

Funcionalidad de Autenticación y registro

Permite crear cuentas de investigador, iniciar sesión, cambiar contraseñas y mantener el usuario evaluador predeterminado.

ID	Elemento del dominio	Descripción (El sistema debe...)	Atributos (si aplica)	Objetivo
RF001	Usuario evaluador inicial	Crear automáticamente un usuario <i>evaluador raíz</i> en la primera ejecución del sistema; no se permite crear otros evaluadores vía interfaz.	<i>id, email, password (hash), role="evaluator"</i>	Garantizar acceso inicial a la evaluación.
RF002	Inicio de sesión (Evaluador)	Permitir al evaluador autenticarse con correo y contraseña.	email, password	Permitir acceso seguro al evaluador.
RF003	Verificar credenciales (Evaluador)	Comprobar credenciales del evaluador en la base de datos, notificando errores claramente.	token (si éxito)	Asegurar autenticación correcta y segura.
RF004	Cambiar contraseña (Evaluador)	Permitir al evaluador autenticado cambiar su contraseña desde su cuenta.	oldPassword, newPassword	Autogestión segura del evaluador.
RF005	Validar contraseña actual y actualizar (Evaluador)	Validar contraseña actual antes de actualizar a una nueva contraseña encriptada de manera segura.	passwordHash, updatedAt	Evitar accesos indebidos y garantizar seguridad.
RF006	Confirmación cambio contraseña (Evaluador)	Mostrar un mensaje claro confirmando que la contraseña se actualizó correctamente.	—	Informar claramente el éxito del cambio.

RF007	Registro investigador	Presentar formulario para registrar investigadores con nombre, correo institucional y contraseña.	name, email, password	Crear cuenta para investigadores nuevos.
RF008	Validar campos obligatorios (Investigador)	Verificar que todos los campos obligatorios del formulario de registro investigador estén completos antes del envío.	—	Garantizar integridad de los datos de registro.
RF009	Validar correo no duplicado	Comprobar que el correo institucional no exista previamente en la base de datos, notificando claramente si está duplicado.	email	Evitar duplicación de usuarios.
RF010	Validar fortaleza de contraseña	Verificar que la contraseña cumpla criterios mínimos de seguridad (≥ 8 caracteres, mayúscula, minúscula, número, carácter especial).	password	Asegurar credenciales robustas.
RF011	Encriptar contraseña (Investigador)	Realizar hashing seguro de la contraseña antes de guardarla en base de datos.	passwordHash	Proteger datos sensibles del usuario.
RF012	Persistir registro investigador	Almacenar en la base de datos la información del investigador tras validar todos los campos y encriptar la contraseña.	userDoc (name, email, passwordHash, createdAt)	Completar exitosamente el registro del investigador.
RF013	Inicio de sesión (Investigador)	Permitir al investigador autenticarse con correo institucional y contraseña.	email, password	Acceso seguro al módulo investigador.
RF014	Verificar credenciales (Investigador)	Validar que el correo y contraseña coincidan con los almacenados, notificando claramente errores.	token (si éxito)	Autenticación segura del investigador.
RF015	Cambiar contraseña (Investigador)	Habilitar opción para que el investigador pueda cambiar su contraseña actual desde la aplicación.	oldPassword, newPassword	Autogestión segura del investigador.

RF016	Validar contraseña actual y actualizar (Investigador)	Comprobar contraseña actual del investigador antes de actualizar a una nueva contraseña segura mediante hashing.	passwordHash, updatedAt	Seguridad e integridad de los datos del investigador.
RF017	Confirmación cambio contraseña (Investigador)	Notificar explícitamente al investigador que la contraseña se ha actualizado correctamente.	—	Informar claramente el éxito del cambio.

Tabla 9. Requerimientos funcionales del módulo de autenticación y registro

Funcionalidad de Gestión de archivos (Evaluador)

Carga de consentimientos PDF, almacenamiento en MinIO, listado, descarga de resultados.

ID	Elemento del dominio	Descripción (El sistema debe...)	Atributos (si aplica)	Objetivo
RF018	Cargar archivos	Permitir al evaluador subir archivos PDF que contengan consentimientos informados para evaluación ética.	file, uid, fechaCarga	Iniciar el proceso de evaluación ética.
RF019	Almacenamiento en MinIO	Garantizar que los archivos cargados se almacenen correctamente en el sistema MinIO para gestión segura y accesible.	bucketId, filePath, filename	Almacenar documentos de manera segura y escalable.
RF020	Listar documentos cargados	Presentar una tabla organizada con los documentos cargados mostrando estado, fecha de carga y acciones disponibles.	fileList (status, date, actions)	Facilitar revisión y control de documentos cargados.
RF027	Descargar PDF resultado evaluación IA	Ofrecer al evaluador la opción de descargar un PDF con los resultados generados por la evaluación automática de IA.	fileUrl, fileName	Permitir uso externo de los resultados generados.

Tabla 10. Requerimientos funcionales del módulo de gestión de archivos para el evaluador

Funcionalidad de Interacción con el modelo de IA

Envío de documentos al LLM, recepción y edición de la retroalimentación.

ID	Elemento del dominio	Descripción (El sistema debe...)	Atributos (si aplica)	Objetivo
RF021	Opción de evaluar documentos	Ofrecer una opción (por ejemplo, un botón "Evaluar") que permita iniciar el proceso de evaluación para cada documento en la lista de forma individual.	...	Darle la opción al usuario de evaluar un archivo
RF022	Enviar documento a IA	Enviar automáticamente el documento cargado al modelo de IA externo para realizar la evaluación ética.	documentId, fileUrl	Evaluar documentos automáticamente con IA.
RF023	Mostrar resultados (cumple/no cumple)	Mostrar claramente los resultados globales de la evaluación generada por la IA, identificando criterios cumplidos o no cumplidos.	resultId, status ("cumple"/"no cumple")	Facilitar toma rápida de decisiones sobre documentos.
RF024	Localizar evidencias del cumplimiento	Indicar dentro del documento las secciones o fragmentos específicos que sustentan la evaluación realizada por la IA.	resultId, evidencias [text, position]	Ayudar al evaluador a verificar fácilmente el análisis de IA.
RF025	Editar retroalimentación	Permitir al evaluador editar la retroalimentación generada por la IA antes de enviarla al investigador o al comité.	feedbackId, editedFeedback	Ofrecer control y precisión al evaluador sobre la retroalimentación.
RF026	Comentar retroalimentación	Permitir al evaluador añadir comentarios adicionales o notas explicativas sobre la retroalimentación generada por la IA.	feedbackId, comments	Complementar el análisis automático con criterio humano.

Tabla 11. Requerimientos funcionales del módulo de interacción con el modelo de IA

Funcionalidad de Gestión de correos

Envío de retroalimentación a investigadores y comité de ética.

ID	Elemento del dominio	Descripción (El sistema debe...)	Atributos (si aplica)	Objetivo
RF028	Enviar correo al investigador	Enviar la retroalimentación detallada resultante de la evaluación al investigador mediante correo electrónico.	destinatarioEmail, evaluationId	Comunicar resultados directamente al investigador.
RF029	Generar automáticamente el contenido (investigador)	Crear automáticamente el contenido del correo electrónico, incorporando la retroalimentación generada por la evaluación IA con un formato claro y legible.	evaluationId, feedback	Mantener consistencia y claridad en la comunicación.
RF030	Confirmar envío investigador	Notificar al evaluador sobre el éxito o fallo del envío del correo al investigador, permitiendo reintentos manuales en caso de error.	statusEnvio, fechaHora	Garantizar trazabilidad y entrega efectiva del correo.
RF031	Enviar correo al comité de ética	Enviar los resultados de la evaluación del proyecto al comité de ética, incluyendo el estado (aprobado o no) y justificaciones detalladas.	destinatarioEmail, evaluationId, estadoEvaluacion	Formalizar comunicación oficial al comité.
RF032	Generar automáticamente contenido (Comité)	Crear automáticamente el contenido del correo para el comité de ética, detallando claramente la decisión y justificación de la evaluación realizada.	evaluationId, decision, justification	Consistencia y claridad en comunicaciones oficiales al comité.

RF033	Reenvío de correo	Permitir al evaluador reenviar manualmente el correo electrónico al investigador en caso de error, notificando claramente el resultado de cada intento.	statusEnvio, fechaHora	Garantizar recepción efectiva y registro de comunicaciones críticas.
-------	-------------------	---	------------------------	--

Tabla 12. Requerimientos funcionales del módulo de gestión de correos

Funcionalidad de Análisis y estadísticas

Gráficos de desempeño, métricas de tiempos y resultados acumulados.

ID	Elemento del dominio	Descripción (El sistema debe...)	Atributos (si aplica)	Objetivo
RF034	Estadísticas generales	Mostrar claramente estadísticas generales como total de evaluaciones, aprobadas, rechazadas y tasa de devolución.	totalEvaluaciones, aprobadas, rechazadas, tasaDevolucion (%)	Proveer una visión global rápida del desempeño.
RF035	Estadísticas comparativas	Mostrar comparativas respecto al periodo anterior (mes pasado) para las métricas generales de evaluación.	periodoActual, periodoAnterior, variacion (%)	Facilitar análisis comparativo del desempeño mensual.
RF036	Tiempo promedio de evaluación	Calcular y presentar el tiempo promedio utilizado para evaluar documentos, expresado claramente en la interfaz.	tiempoPromedioEvaluacion	Medir y optimizar la eficiencia del proceso evaluativo.
RF037	Interfaz visual intuitiva	Proveer gráficos e indicadores visualmente claros e intuitivos para que el evaluador comprenda rápidamente los datos presentados.	tipoGrafico, datos	Mejorar la interpretación visual rápida de estadísticas clave.

RF039	Actualización dinámica de estadísticas	Actualizar automáticamente los datos presentados en la interfaz al momento de generarse cambios en las evaluaciones.	—	Asegurar que la información estadística sea siempre actualizada.
-------	--	--	---	--

Tabla 13. Requerimientos funcionales del módulo de análisis y estadísticas

Funcionalidad de Configuración del modelo de IA

Modificación de prompts y umbrales de precisión.

ID	Elemento del dominio	Descripción (El sistema debe...)	Atributos (si aplica)	Objetivo
RF040	Modificar prompts de evaluación	Permitir al evaluador modificar y ajustar los prompts que utiliza el modelo de IA, sin necesidad de conocimientos técnicos avanzados.	promptId, textoPrompt, version, activo (bool)	Adaptar criterios fácilmente según necesidades del comité.
RF041	Restablecer prompts	Permitir devolver el estado de los prompts a su configuración inicial según lo requiera el evaluador.	promptId, activo (bool)	Gestionar flexiblemente los criterios activos.
RF042	Prompts efectivos	Ser capaz de identificar las leyes que no se están cumpliendo ya sea por falta de información o incumplimiento de leyes éticas.	promptId, textoPrompt, version, fechaCreacion	Mantener registro histórico de cambios en los criterios.
RF043	Visualizar prompts actuales	Mostrar claramente los prompts actuales configurados para cada tipo de evaluación, facilitando su consulta y revisión.	promptId, textoPrompt, activo	Facilitar consulta rápida y transparente de los criterios usados.

Tabla 14. Requerimientos funcionales del módulo de configuración del modelo de IA

Funcionalidad de Gestión del consentimiento (Investigador)

Formulario inteligente para elaborar el consentimiento, validaciones y generación final.

ID	Elemento del dominio	Descripción (El sistema debe...)	Atributos (si aplica)	Objetivo
RF044	Crear caso de consentimiento informado	Permitir al investigador crear un nuevo caso, iniciando el llenado del formulario de consentimiento.	uid, nombreProyecto, fecha	Iniciar la construcción estructurada del documento.
RF045	Completar formulario	Mostrar un formulario con los campos requeridos para el consentimiento informado: introducción, info general, riesgos, etc.	campoId, valor	Guiar al investigador en la recolección completa de datos.
RF046	Validar campos obligatorios	Verificar que todos los campos necesarios del formulario hayan sido diligenciados antes de permitir continuar.	campoId, obligatorio (bool)	Evitar generación de documentos incompletos.
RF047	Mostrar errores de completitud	Notificar al investigador de los campos que estén incompletos o mal diligenciados.	campoId, error	Brindar retroalimentación inmediata durante el llenado.
RF048	Guardar borrador del consentimiento	Permitir guardar el progreso del formulario sin necesidad de completar todos los campos.	datosParciales	Facilitar trabajo progresivo sobre el documento.
RF049	Generar PDF del consentimiento	Generar un archivo PDF con los datos del formulario diligenciado, siguiendo el formato oficial del comité de ética.	fileUrl, version	Producir el documento final en formato estandarizado.
RF050	Almacenar PDF generado	Guardar automáticamente el PDF generado en el historial del investigador en el sistema MinIO.	filePath, uid, fechaGeneracion	Mantener registro histórico de los casos creados.
RF051	Consultar historial de casos	Mostrar al investigador una lista de casos creados con sus fechas, estados y enlaces de descarga.	casoId, fecha, estado	Permitir trazabilidad y gestión de sus propios consentimientos.

Tabla 15. Requerimientos funcionales del módulo de gestión del consentimiento para el investigador

Requerimientos de desempeño

Una vez establecidos los requerimientos funcionales, se definen los requerimientos de desempeño con el fin de garantizar que el sistema opere de forma estable, rápida y eficiente bajo condiciones reales de uso. Estos se clasifican en requerimientos estáticos, relacionados con la carga de usuarios conectados simultáneamente, y requerimientos dinámicos, enfocados en tiempos de respuesta y procesamiento.

Requerimientos Estáticos

- Usuarios concurrentes: El sistema deberá ser capaz de atender simultáneamente hasta 5 usuarios de tipo evaluador y 20 usuarios de tipo investigador, sin comprometer la estabilidad ni el rendimiento general de la plataforma.
- Criterio de medición: Durante pruebas de carga, el sistema deberá demostrar que puede mantener un funcionamiento correcto, sin errores ni caídas perceptibles, ante la concurrencia máxima establecida para ambos roles.

Requerimientos Dinámicos

- Latencia general: Acciones habituales como autenticación, registro, consulta de historial, carga de documentos y visualización de estadísticas deberán completarse en un tiempo menor o igual a 3 segundos por transacción, bajo condiciones normales.
- Evaluación mediante IA: El sistema deberá recibir una respuesta de evaluación por parte del modelo de lenguaje dentro de un tiempo máximo de 20 segundos, contados a partir del envío del documento a la API.
- Generación de PDF: La generación de un documento PDF del consentimiento informado deberá ejecutarse en menos de 10 segundos por solicitud.
- Envío de correos electrónicos: El sistema deberá procesar y enviar correos electrónicos (ya sea al investigador o al comité de ética) en un tiempo máximo de 15 segundos, sin incluir posibles retardos del proveedor de correo externo.

Criterios de medición para desempeño dinámico

- Las funcionalidades deberán cumplir con los tiempos establecidos en al menos el 90% de las ejecuciones durante pruebas de validación controladas.
- Las mediciones serán tomadas bajo condiciones equivalentes al entorno de producción (con carga simulada de usuarios concurrentes).

Atributos de calidad asociados

Atributo	Definición	Aplicación en la plataforma
Latencia	Tiempo entre una solicitud del usuario y la respuesta del sistema.	Menor o igual a 3s en operaciones comunes; ≤ 20s para evaluaciones por IA.

Tasa de procesamiento	Número de operaciones completadas con éxito en un intervalo de tiempo definido.	Capacidad para atender múltiples solicitudes de usuarios concurrentes.
Capacidad	Nivel de carga que el sistema puede soportar antes de afectar la latencia.	Soporte garantizado para 25 usuarios activos simultáneos (5 evaluadores + 20 investigadores).
Disponibilidad	Tiempo total durante el cual el sistema está accesible y funcional.	Disponibilidad esperada $\geq 95\%$ durante el uso operativo normal.

Tabla 16. Tabla de atributos de calidad

Restricciones de diseño

El diseño e implementación de la plataforma web está sujeto a diversas restricciones técnicas y decisiones arquitectónicas que guiarán el desarrollo del sistema. Estas restricciones se detallan a continuación y también se referencian en el Documento de Diseño del Software (SDD), donde se abordan con mayor profundidad.

Lenguajes de Programación

El sistema será desarrollado principalmente utilizando TypeScript en ambas capas (frontend y backend), con las siguientes herramientas específicas:

- Backend: TypeScript + Bun como runtime principal, por su rendimiento superior y velocidad de ejecución. No obstante, se continúa utilizando Node.js con npm en el entorno de pruebas debido a la compatibilidad con ciertas dependencias. El backend está estructurado con Express.js y sigue los principios de Clean Architecture para mantener una lógica desacoplada del framework o motor de persistencia.
- Frontend: TypeScript + React, organizado bajo el patrón de diseño Atomic Design. Este enfoque facilita la creación de interfaces modulares y mantenibles, separando los componentes en niveles jerárquicos (átomos, moléculas, organismos, plantillas y páginas). Se emplean herramientas modernas como Vite (para bundling eficiente), shadcn/ui (para componentes visuales accesibles) y TanStack Query (para manejo eficiente de datos en cliente).

Herramientas CASE (Análisis y Diseño)

Se utiliza Visual Paradigm como herramienta CASE principal, tanto para el modelado UML (casos de uso, clases, secuencia) como para el diseño del modelo de dominio. Esta herramienta facilita la documentación y permite mantener sincronía entre los modelos conceptuales y el código fuente.

Restricciones de Hardware y Software

Debido al enfoque de arquitectura basada en contenedores y al uso de herramientas modernas, se establecen las siguientes restricciones:

- Hardware mínimo para desarrollo local:
 - Procesador de al menos 4 núcleos.
 - Memoria RAM de mínimo 8 GB (recomendado 12 GB si se utilizan múltiples contenedores simultáneamente).
 - Espacio en disco libre superior a 10 GB para manejo de dependencias y persistencia de volúmenes de Docker.
- Software requerido:
 - Docker y Docker Compose para la construcción, despliegue y orquestación de contenedores.
 - MongoDB y MinIO, ambos ejecutados en contenedores independientes.
 - Navegadores modernos (Chrome, Edge o Firefox) compatibles con HTML5 para uso de la plataforma.

La plataforma fue desarrollada y probada bajo el paradigma cliente-servidor, utilizando contenedores para simular los distintos servicios distribuidos.

Diseño de la Arquitectura Final del Sistema

La plataforma adopta una arquitectura N-Tier, con separación explícita entre:

- Capa de presentación: React + Vite
- Capa de lógica de negocio: Express.js sobre Bun
- Capa de persistencia: MongoDB y MinIO

Cada capa está organizada siguiendo principios sólidos de diseño, y se han adoptado los siguientes patrones arquitectónicos:

- Clean Architecture en el backend: El backend está estructurado en capas independientes (controladores, casos de uso, entidades y servicios de infraestructura), lo que permite una alta cohesión interna, bajo acoplamiento y una mayor capacidad de testeo, mantenimiento y escalabilidad. Esta arquitectura promueve que la lógica de negocio sea independiente del framework, la base de datos o las librerías utilizadas.
- Atomic Design en el frontend: Se utiliza este patrón de diseño para la construcción modular de la interfaz de usuario, lo que permite una mejor reutilización, estructura clara y facilidad de mantenimiento visual.
- Arquitectura basada en contenedores: Cada módulo del sistema se ejecuta en un contenedor Docker independiente (frontend, backend, base de datos, almacenamiento), lo que permite una infraestructura modular, aislada y fácilmente replicable en distintos entornos.
- Comunicación entre servicios: Se emplea Axios como cliente HTTP para consumir APIs, permitiendo un manejo consistente de errores, encabezados y autenticación por token.
- Control de versiones: Se utiliza GitHub como sistema de gestión de versiones y control colaborativo del código fuente.

Construcción del Modelo de Bases de Datos

La base de datos utilizada es MongoDB, un sistema NoSQL document-oriented, elegido por su flexibilidad para almacenar estructuras JSON complejas, como los formularios de consentimiento y las evaluaciones generadas por IA.

- Las entidades del sistema se definen en esquemas estructurados que permiten consultas eficientes y escalables.
- Se ha documentado el modelo de dominio detallado (ver Anexo 2) y se definió el uso de claves foráneas mediante referencias entre colecciones (e.g., usuarios, casos, evaluaciones, normas éticas).

Atributos del sistema de software (No funcionales)

Los atributos de calidad no funcionales son esenciales para garantizar que la plataforma cumpla con las expectativas de rendimiento, disponibilidad, seguridad, mantenibilidad y portabilidad. A continuación, se presentan los principales atributos considerados en el diseño del sistema, incluyendo sus mecanismos de implementación y criterios de evaluación medibles.

Confiabilidad

El sistema debe operar de forma estable y predecible, incluso frente a errores o situaciones inesperadas. Se han establecido los siguientes mecanismos:

- Manejo de transacciones concurrentes: El backend gestionará múltiples solicitudes de usuarios evaluadores e investigadores, garantizando consistencia y evitando conflictos en el acceso concurrente a la base de datos.
- Tolerancia a fallos: Se implementan controladores de errores con try/catch centralizados en el backend (usando Bun), que aseguran la respuesta del sistema sin interrupciones ante fallas inesperadas.
- Persistencia de datos: Toda la información gestionada por la plataforma se almacena en MongoDB, con respaldo automático de colecciones clave (como usuarios, casos y evaluaciones). Además, se utiliza MinIO para almacenamiento de archivos, asegurando su persistencia con tolerancia a pérdidas de sesión o conectividad.
- Criterio de evaluación: El sistema debe manejar adecuadamente todos los errores identificados durante las pruebas controladas, evitando interrupciones en el servicio. Asimismo, se debe verificar que los datos críticos (casos, evaluaciones, usuarios) se mantengan íntegros y disponibles incluso ante fallos del sistema o interrupciones inesperadas.

Disponibilidad

La plataforma deberá estar disponible el 99% del tiempo durante la jornada laboral establecida, considerando la naturaleza institucional del sistema.

- Aplicaciones externas: El modelo de IA utilizado depende de servicios en la nube, cuya disponibilidad puede afectar el tiempo de evaluación. Se recomienda su uso durante horarios con disponibilidad garantizada.
- Comunicación: Se usan protocolos TCP/IP para garantizar el acceso constante entre los componentes del sistema. Aunque no se ha implementado HTTPS en el entorno actual, se considera una medida obligatoria para entornos de producción.
- Tiempo administrador: El evaluador es el actor principal que gestiona evaluaciones. Se requiere que su disponibilidad esté alineada con el ciclo de evaluación del Comité.

- Criterio de evaluación: Registro de tiempo activo del sistema en relación con la disponibilidad programada.

Seguridad

El sistema manipula información sensible, como evaluaciones éticas y datos de investigadores, por lo tanto se han definido medidas rigurosas de protección:

- Cuentas para usuarios: Se implementa autenticación basada en JWT. El sistema sólo crea automáticamente un usuario evaluador y permite el registro de investigadores por parte del sistema.
- Encriptación: Las contraseñas se almacenan usando bcrypt con salting. El acceso a la API está protegido con headers de autorización JWT.
- Criterio de evaluación: Validación de credenciales, rutas protegidas, y que ningún dato sensible (como tokens o contraseñas) se transmita sin cifrado.

Mantenibilidad

La mantenibilidad del sistema ha sido considerada desde el diseño inicial:

- Modularización: El proyecto sigue Clean Architecture en el backend (capas desacopladas: dominio, aplicación, infraestructura y presentación) y Atomic Design en el frontend (componentes: átomos, moléculas, organismos, plantillas y páginas).
- Reutilización: Los componentes son diseñados para ser fácilmente reutilizables, tanto en el frontend como en los servicios del backend (por ejemplo, middleware de autenticación, validadores, adaptadores).
- Criterio de evaluación: Verificación mediante revisión de código y pruebas de refactorización sin afectar otras funcionalidades.

Portabilidad

La plataforma está diseñada para ejecutarse en diferentes entornos de hardware y software, tanto en desarrollo local como en producción institucional.

- Contenedores Docker: Toda la solución (frontend, backend, base de datos, almacenamiento) está contenida en servicios Docker orquestados por Docker Compose, lo que permite replicar el entorno fácilmente.
- Compatibilidad Multiplataforma: El sistema ha sido probado en entornos Windows y Linux y es accesible desde navegadores modernos como Chrome, Edge y Firefox.
- Criterio de evaluación: Ejecución exitosa en entornos heterogéneos sin necesidad de configuración adicional.

Requerimientos de la base de datos

La plataforma utilizará MongoDB como sistema de gestión de bases de datos NoSQL, debido a su capacidad para manejar grandes volúmenes de datos textuales y su estructura flexible basada en documentos. A continuación, se detallan los principales aspectos técnicos y requerimientos asociados al uso de esta base de datos, según las buenas prácticas de diseño y desempeño.

Tipos de datos almacenados

El sistema almacenará datos principalmente en formato JSON, correspondientes a formularios de consentimiento informado, evaluaciones, usuarios y prompts de configuración.

Los tipos de datos más utilizados incluyen:

- String: Para campos como nombre del proyecto, justificación, correo, etc.
- Date: Para registrar fechas de creación, evaluación, envío de resultados, etc.
- Boolean: Para estados como aprobado o no aprobado.
- Number: Para versiones, códigos, tiempo promedio, etc.
- ObjectId: Para establecer referencias entre documentos (ej. usuarios, casos, evaluaciones).

Tipo de consultas utilizadas

La mayoría de las consultas serán de tipo búsqueda, filtrado y actualización por ID, especialmente desde los servicios DAO en el backend.

Se utilizarán técnicas equivalentes a los prepared statements, mediante las operaciones seguras que proporciona Mongoose (ODM para MongoDB), evitando la inserción de sentencias maliciosas y asegurando la integridad del sistema.

Consultas frecuentes incluyen:

- Búsqueda por UID o ID del caso.
- Filtrado por estado de evaluación.
- Ordenamiento por fecha.

Indexación de los datos

Para optimizar el rendimiento en tiempo real, se implementarán índices en campos críticos como:

- `_id` (clave primaria automática de MongoDB).
- `uid` (relación entre entidades).
- `status` y `fecha` (para evaluaciones y estadísticas).

Estos índices permitirán acelerar el acceso a registros específicos y evitar demoras en las operaciones frecuentes del evaluador.

Utilización de Primary Key

Cada documento almacenado en MongoDB cuenta con una clave primaria única (`_id`), que actúa como identificador global dentro de la base.

Adicionalmente, se usan referencias explícitas (ObjectId) para mantener la integridad entre entidades como usuarios, casos y evaluaciones, lo que facilita la navegación y evita ambigüedades durante el acceso a datos relacionados.

Frecuencia de acceso

Se estima una frecuencia de acceso moderada a alta, con una mayor carga en las operaciones de lectura por parte de los evaluadores, y escrituras periódicas por parte de los investigadores.

Con base en los requerimientos de desempeño, el sistema debe ser capaz de manejar sin inconvenientes:

- Hasta 20 usuarios investigadores creando y almacenando documentos simultáneamente.
- Hasta 5 evaluadores consultando estadísticas y generando evaluaciones en paralelo.

La arquitectura N-Tier y el uso de Mongoose como middleware de acceso a datos ayudan a reducir la carga sobre la base y evitan consultas innecesarias.

Contenedorización y replicabilidad

MongoDB será desplegado dentro de un contenedor Docker, lo que permite garantizar entornos de ejecución replicables tanto en desarrollo como en producción. Las configuraciones de volumen aseguran la persistencia de datos, y el uso de variables de entorno facilita la personalización de credenciales y puertos.

4. PROCESO INGENIERÍA DE REQUERIMIENTOS

Durante el proceso de administración de requerimientos se generarán los siguientes artefactos clave:

1. **Historias de Usuario**
Las historias de usuario documentarán de manera simple y clara las funcionalidades que los usuarios finales (los miembros del comité de ética) necesitan. Estas historias facilitarán la comprensión y priorización de las tareas en función de su valor y urgencia, permitiendo adaptar el desarrollo a los cambios y nuevos requerimientos.
2. **Uso de click up**
Se utilizará la plataforma ClickUp para gestionar la trazabilidad de los requerimientos desde su definición inicial hasta su implementación final. A través de esta herramienta, se organizarán matrices que incluirán el ID del requerimiento, descripción, prioridad, estado, módulo afectado y casos de uso relacionados. Esto permitirá controlar los cambios en las historias de usuario, vincular tareas específicas y evaluar su impacto en el resto del sistema de forma centralizada y colaborativa.
3. **Técnicas y Métodos Utilizados para la Especificación de Requerimientos**
Para la construcción de esta especificación, se utilizaron diversas técnicas, como entrevistas con los miembros del comité de ética para captar sus necesidades, talleres de requerimientos para definir las funcionalidades clave de la plataforma, y análisis de documentación existente (políticas éticas y normativas del HUSI). Adicionalmente, se utilizaron prototipos (*mockups*) iniciales para validar las funcionalidades principales e historias de usuario que describen las interacciones específicas entre los usuarios y el sistema.
4. **Gestión de Cambios y Aprobación de Requerimientos**
En cuanto al proceso de administración de requerimientos, se comenzará con la detección de nuevas necesidades o cambios por parte de los clientes o los desarrolladores. Los clientes

podrán solicitar modificaciones o nuevas funcionalidades durante el desarrollo, mientras que los desarrolladores identificarán mejoras o ajustes necesarios en función del progreso del proyecto.

En general, el proceso de levantamiento de requerimientos será iterativo y su flujo de actividades estará definido de la siguiente manera:

1. **Solicitud de Cambio o Nuevo Requerimiento:** Los clientes o desarrolladores pueden proponer nuevas historias de usuario o solicitar cambios a las historias existentes.
2. **Análisis y Evaluación del Requerimiento:** El equipo de desarrollo analizará la viabilidad del nuevo requerimiento o cambio propuesto, evaluando su impacto en el sistema y los recursos necesarios para implementarlo. En esta fase, se actualizarán las matrices de trazabilidad para reflejar cualquier modificación.
3. **Aprobación del Cambio:** Si el cambio es viable, se presentará una propuesta formal al director de tesis para su aprobación o rechazo, y en caso de que sea aprobada, se presentará la propuesta formal al HUSI para verificar también su aprobación. Una vez aprobado, el nuevo requerimiento se incluirá en el backlog y se priorizará para su desarrollo en un sprint posterior.
4. **Actualización de Artefactos:** Tras la aprobación del cambio, se actualizarán las historias de usuario y las matrices de trazabilidad, asegurando que todos los elementos afectados estén correctamente documentados y alineados con los objetivos del sistema.
5. **Desarrollo e Implementación:** Los desarrolladores implementarán los cambios aprobados, siguiendo las historias de usuario actualizadas y asegurando que la nueva funcionalidad cumpla con los requerimientos definidos.
6. **Verificación y Validación:** Una vez completada la implementación, se realizará una verificación para asegurar que los cambios cumplen con las especificaciones solicitadas. El cliente revisará y validará el producto para confirmar que los nuevos requerimientos se han implementado correctamente.

Este proceso iterativo permitirá que los cambios se controlen y gestionen de manera efectiva a lo largo del desarrollo, manteniendo la calidad del sistema. Este proceso está alineado con la sección de Control de Requerimientos del SPMP, que detalla la periodicidad de revisiones y la documentación formal del proceso de administración de cambios para asegurar una correcta trazabilidad y localización de los requerimientos.

5. PROCESO DE VERIFICACIÓN

En esta sección se describe detalladamente el proceso planificado para la verificación y validación tanto de esta especificación como de los requerimientos individuales del sistema. Este proceso busca

asegurar que el desarrollo de la plataforma cumpla con los estándares de calidad, funcionalidad, seguridad y rendimiento establecidos, y que satisfaga efectivamente las necesidades del Comité de Ética del Hospital Universitario San Ignacio.

El enfoque adoptado contempla diversas estrategias de prueba orientadas a cubrir desde la interacción entre componentes hasta el comportamiento del sistema bajo condiciones reales de uso. A continuación, se describen las principales actividades que componen este proceso.

- *Pruebas de integración*

Se desarrollarán pruebas de integración utilizando las herramientas Jest y Supertest. Estas pruebas se enfocan en validar la correcta comunicación entre los distintos módulos del backend, como los controladores, casos de uso, repositorios de datos y servicios externos. El objetivo es asegurar que el sistema responde de forma coherente y sin errores ante distintas combinaciones de entradas y flujos de interacción.

- *Pruebas end-to-end (E2E)*

Se aplicarán pruebas end-to-end mediante el framework Playwright. Estas pruebas validarán el funcionamiento completo del sistema desde la perspectiva del usuario final, incluyendo flujos como el registro, carga de documentos, evaluación con IA, consulta de resultados y envío de correos. Se cubrirán tanto los flujos principales como escenarios alternativos, garantizando así una experiencia de uso robusta y sin interrupciones.

- *Pruebas de carga y desempeño*

El sistema será sometido a pruebas de carga utilizando la herramienta Artillery, con el fin de evaluar su comportamiento bajo condiciones de uso intensivo. Se simulará un entorno con hasta 25 usuarios concurrentes (20 investigadores y 5 evaluadores), realizando operaciones críticas como autenticación, carga de archivos y evaluación de documentos. Estas pruebas permitirán identificar cuellos de botella, medir los tiempos de respuesta y validar la estabilidad del sistema.

- *Pruebas de usabilidad y aceptación*

Se llevarán a cabo sesiones de validación con usuarios finales (miembros del Comité de Ética) para comprobar que la plataforma cumple con los requerimientos establecidos y es fácil de usar. Estas pruebas permitirán identificar oportunidades de mejora en la experiencia del usuario y asegurar que la solución desarrollada responde adecuadamente a sus necesidades y expectativas.

Verificación de los requerimientos

Cada requerimiento funcional y no funcional será validado mediante criterios claros y verificables:

- Cobertura en los escenarios de prueba definidos.
- Evidencia de cumplimiento por medio de reportes de ejecución.
- Retroalimentación recibida durante las sesiones de validación.
- Trazabilidad con los casos de uso y funcionalidades documentadas.

Documentación del proceso

Todas las actividades de prueba, resultados, métricas e incidencias serán documentadas en el Documento de Pruebas del Proyecto. Este documento incluirá:

- Descripción de cada tipo de prueba ejecutada.
- Escenarios cubiertos y resultados obtenidos.
- Registro de defectos detectados y acciones correctivas aplicadas.
- Tiempos promedio de ejecución, tasas de éxito y estabilidad del sistema.

Este proceso integral de verificación permitirá asegurar que el sistema desarrollado es confiable, funcional, seguro y alineado con los objetivos planteados desde el inicio del proyecto.

6. ANEXOS

1. Diagrama de casos de uso del sistema

 DIAGRAMACASODEUSO.png


2. Anexo 2. Modelo de datos del sistemas

 ModeloDatos.png

3. Mockups de la plataforma

<https://www.figma.com/design/jbRBFfrlk5xtaC1fzWJ3HP/Tesis-mockups?node-id=8-2&t=zSxhsUNzBNw8NtFb-1>

4. Requerimientos funcionales

 Requerimientos-finales.xlsx