



UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA
T U N J A

TÍTULO PROYECTO

Repositorio Digital de Trabajos de Grado para la Facultad de Ingeniería de Sistemas de la
Universidad Santo Tomás - Seccional Tunja

PROPONENTE(S)

Juan Andres Cabrera Suarez
2321515

Cristian David Pineda Molano
2330324

DIRECTOR

Juan Francisco Mendoza Moreno

Tunja
26/11/2025

Índice

1. Ficha de objetivos.....	4
2. Introducción	5
3. Planteamiento del problema.....	6
3.1 Árbol de problemas	7
4. Justificación.....	8
5. Objetivos	9
5.1 Objetivo general.....	9
5.2 Objetivos específicos.....	9
6. Marco teórico	9
7. Metodología.....	12
7.1 Fases del modelo de prototipado aplicado al proyecto	14
7.2 Aplicación práctica del modelo de prototipado en el proyecto	14
8. Análisis y diseño del sistema.....	16
8.1 Análisis de requerimientos	16
8.1.1 Requerimientos funcionales	16
8.1.2 Requerimientos no funcionales	17
8.2 Arquitectura del sistema	17
8.3 Diseño del sistema.....	19
8.3.2 Diagramas del sistema.....	20
8.4 Diseño de seguridad.....	25
8.5 Modelo de datos	26
9. Implementación del sistemas	26
9.1 Consideraciones generales de implementación	26
9.2 Implementación del backend.....	27
9.3 Implementación de frontend	27
9.4 Integraciones externas	28
9.5 Decisiones técnicas relevantes.....	28
10. Cumplimiento de objetivos y desarrollo realizado	29

10.1	Cumplimiento del objetivo 1: Análisis de requerimientos	29
10.2	Cumplimiento del objetivo 2: Implementación del repositorio digital	31
10.3	Cumplimiento de objetivo 3: Evaluación del sistema	37
11.	Conclusiones	40
11.1	Limitaciones del proyecto	40
11.2	Trabajos futuros y propuestas de mejora	41
Referencias		43

1. Ficha de objetivos

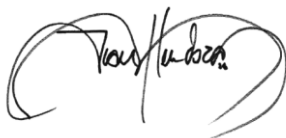
Título	Repositorio Digital de Trabajos de Grado para la Facultad de Ingeniería de Sistemas de la Universidad Santo Tomás - Seccional Tunja
Nombre Estudiante	Juan Andres Cabrera Suarez / Cristian David Pineda Molano
Correo electrónico estudiante	juan.cabrera@usantoto.edu.co / cristian.pinedam@usantoto.edu.co
Director	Juan Francisco Mendoza Moreno
Lugar de ejecución del proyecto	Tunja, Boyacá
Duración	4 meses
Palabras claves	Repositorio, digital, trabajo de grado, ingeniería, sistemas, universidad.



Firma del autor
Juan Andres Cabrera Suarez



Firma del autor
Cristian David Pineda Molano



Firma del director
Juan Francisco Mendoza Moreno

2. Introducción

En las instituciones de educación superior, los trabajos de grado constituyen una de las principales fuentes de producción académica y científica, ya que reflejan los procesos de investigación, innovación y desarrollo realizados por los estudiantes a lo largo de su formación profesional. Estos documentos no solo evidencian el cumplimiento de los requisitos académicos para optar a un título, sino que también representan un acervo de conocimiento valioso que puede ser reutilizado, consultado y ampliado por estudiantes, docentes e investigadores.

No obstante, en muchas facultades universitarias la gestión de estos trabajos se realiza de manera descentralizada, mediante archivos físicos, carpetas digitales aisladas o repositorios no estandarizados, lo que dificulta su organización, preservación y consulta. Esta situación genera problemas como la pérdida de información, la baja visibilidad de la producción académica, la duplicidad de temáticas y la ausencia de mecanismos eficientes de búsqueda y control de acceso. En consecuencia, se limita el aprovechamiento del conocimiento generado y se afectan procesos académicos y de investigación.

En la Facultad de Ingeniería de Sistemas de la Universidad Santo Tomás, seccional Tunja, se identificó la carencia de un repositorio digital institucional que permitiera centralizar los trabajos de grado y gestionarlos de forma estructurada, segura y accesible. Esta necesidad evidenció la importancia de contar con una plataforma tecnológica que facilitara el almacenamiento, la catalogación y la recuperación de la información, así como la preservación digital de los documentos y productos académicos.

En respuesta a esta problemática, el presente proyecto tuvo como propósito el desarrollo de un repositorio digital de trabajos de grado, implementado como un sistema web que integra funcionalidades de gestión, consulta y administración de la información académica. El sistema fue diseñado bajo una arquitectura cliente-servidor, incorporando mecanismos de autenticación, control de roles, búsqueda avanzada y gestión de anexos, con el fin de apoyar los procesos académicos, investigativos y administrativos de la facultad.

Este documento presenta el proceso de desarrollo del repositorio digital, abordando desde el análisis del problema y la definición de requerimientos, hasta el diseño, implementación y validación del sistema. Asimismo, se exponen los fundamentos teóricos que sustentan el proyecto, la metodología empleada, los resultados obtenidos y las conclusiones derivadas de la experiencia, con el objetivo de evidenciar el aporte del sistema a la gestión del conocimiento y a la modernización de los procesos institucionales.

3. Planteamiento del problema

La facultad de ingeniería de sistemas de la Universidad Santo Tomas – Seccional Tunja no contaba con un repositorio digital centralizado que permitiera almacenar, organizar y consultar de manera eficiente los trabajos de grado desarrollados por los estudiantes. Esta carencia generaba una dispersión considerable de la información en distintos formatos y medios, dificultando su acceso oportuno y limitando la preservación a largo plazo de la producción académica de la facultad.

La inexistencia de una plataforma tecnológica unificada restringía posibilidad de que estudiantes, docentes e investigadores accedieran de forma ágil a trabajos previos, afectando la continuidad académica, la visibilidad institucional y la generación de nuevo conocimiento. Esta situación incrementaba el riesgo de duplicidad de temáticas, la pérdida de insumos valiosos para procesos investigativos y la disminución del impacto de los resultados académicos producidos.

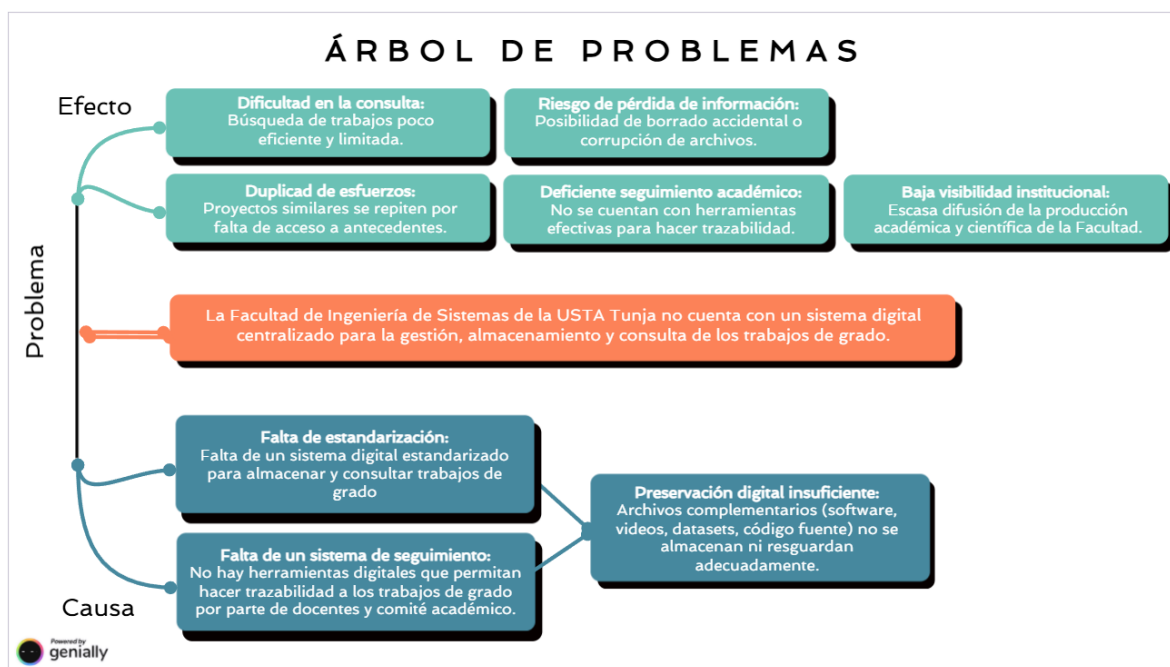
Entre las causas principales se identificó la falta de un sistema digital estandarizado que integrara funcionalidades de almacenamiento seguro, organización estructurada y mecanismo de búsqueda especializados. Como consecuencia, la facultad carecía de un medio confiable que garantizara la preservación digital de los trabajos de grado y de un mecanismo eficiente para su consulta y aprovechamiento.

Ante este escenario, se evidenció la necesidad de diseñar e implementar un repositorio digital institucional que centralizara, preservara y pusiera a disposición la producción académica de los estudiantes. Esta solución contribuyó a fortalecer la gestión documental, mejorar los procesos académicos y apoyar la toma de decisiones al asegurar la disponibilidad y trazabilidad del conocimiento generado en la facultad.

3.1 Árbol de problemas

Con el fin de estructurar de manera clara la problemática identificada, se elaboró un árbol de problemas que permite visualizar las principales causas y efectos asociados a la ausencia de un repositorio digital centralizado en la Facultad de Ingeniería de Sistemas. Esta herramienta facilitó la identificación del problema central, así como su relación con las deficiencias actuales en la gestión, preservación y consulta de los trabajos de grado, sirviendo como base para la formulación de los objetivos del proyecto.

Figura 1. Árbol de problemas del repositorio digital de trabajos de grado



Fuente: Elaboración propia.

En la Figura 1 se observa que el problema central corresponde a la inexistencia de un sistema digital centralizado para la gestión, almacenamiento y consulta de los trabajos de grado.

Entre las principales causas se identifican la falta de estandarización en los procesos de almacenamiento, la ausencia de un sistema de seguimiento académico y una preservación digital insuficiente de los productos derivados de los proyectos, como software, videos y conjuntos de datos.

Como efectos directos de esta problemática se evidencian la dificultad en la consulta de antecedentes, la duplicidad de esfuerzos investigativos, el riesgo de pérdida de información, el deficiente seguimiento académico y la baja visibilidad institucional de la producción académica.

Este análisis permitió establecer la necesidad de una solución tecnológica que centralizara la información, garantizara su preservación y facilitara su consulta, aspectos que orientaron la definición de los objetivos y el diseño del repositorio digital propuesto.

4. Justificación

El desarrollo del repositorio digital de trabajos de grado respondió a la necesidad identificada en la facultad de ingeniería de sistemas de contar con un sistema centralizado que permitiera organizar, preservar y consultar de manera eficiente la producción académica generada por los estudiantes. Durante la ejecución del proyecto se evidencio que la ausencia de una plataforma tecnológica adecuada ocasionaba dificultades en el acceso a información relevante, pérdida de insumos para procesos investigativos y limitaciones en la continuidad académica derivadas de la dispersión de los trabajos en distintos medios y formatos.

La implementación del repositorio aportó una solución integral al problema diagnosticado inicialmente, al consolidar un entorno digital que garantizo la conservación, disponibilidad y trazabilidad de los trabajos de grado. Asimismo, facilitó la consulta de antecedentes, redujo la duplicidad de temáticas y fortaleció tanto los procesos de investigación como la toma de decisiones dentro de la facultad.

El proyecto permitió generar un impacto positivo en la gestión documental institucional, puesto que la plataforma desarrollada contribuyó a mejorar la visibilidad del conocimiento producido por los estudiantes, promovió la organización académica y facilitó el acceso a información valiosa para docentes, estudiantes e investigadores. Además, el trabajo significó un aporte al fortalecimiento de las capacidades tecnológicas de la facultad, al integrar buenas prácticas de ingeniería de sistemas, criterios de calidad y mecanismos de preservación digital.

En conjunto, la realización del proyecto demostró ser pertinente y necesaria para la atender la problemática detectada, y constituyó un aporte significativo a la consolidación del acervo académico de la facultad y al mejoramiento de sus procesos investigativos y formativos.

5. Objetivos

5.1 Objetivo general

Desarrollar un repositorio digital que centralice, almacene y facilite el acceso a los trabajos de grado de la Facultad de Ingeniería de Sistemas de la Universidad Santo Tomás – Tunja incluyendo tanto documentos como productos generados, utilizando una metodología ágil, con el fin de apoyar la gestión, consulta y preservación a largo plazo de estos resultados académicos.

5.2 Objetivos específicos

Nro.	Objetivo específico
1	Analizar los requerimientos funcionales y no funcionales del repositorio mediante el uso del estándar IEEE 830 como formato principal para la redacción del SRS, con el fin de asegurar que cumpla con las necesidades de los usuarios, los estándares de almacenamiento y las políticas de acceso a la información, garantizando su eficiencia y seguridad.
2	Implementar una plataforma web que facilite la carga, gestión y consulta de los trabajos de grado, asegurando una estructura de datos organizada y un acceso intuitivo a la información mediante herramientas de búsqueda y clasificación.
3	Evaluar el desempeño del software de acuerdo con usabilidad, seguridad y desempeño, mediante pruebas de carga y de calidad en la codificación, con el uso de herramientas como sonarQube, JMeter o Selenium.

6. Marco teórico

6.1 Repositorios digitales institucionales

Los repositorios digitales institucionales son plataformas tecnológicas que permiten la recopilación, gestión y difusión de la producción académica de una institución educativa (Subirats & Minguillón, 2015). Estas herramientas ayudan a centralizar documentos, facilitar su recuperación y promover el acceso abierto a la producción intelectual de estudiantes y docentes (Pinfield et al., 2017).

Los repositorios digitales se estructuran con metadatos que describen los contenidos, lo que permite búsquedas más eficientes y precisas en comparación con sistemas no estandarizados. Además, favorecen la preservación de materiales digitales, incrementan la visibilidad institucional y mejoran la reutilización del conocimiento generado (Sowell, 2016).

6.2 Sistemas de información académicos

Un sistema de información académico es una solución orientada a la recolección, almacenamiento, procesamiento y presentación de datos relacionados con procesos educativos y de investigación dentro de una institución (Romero et al., 2018). Estos sistemas no solo soportan funciones administrativas, sino que también proporcionan información estratégica que optimiza la toma de decisiones y la gestión de recursos académicos.

En el contexto de gestión de trabajos de grado, un sistema de información permite organizar el ciclo completo de vida de los documentos, desde su registro hasta su consulta, mejorando la calidad de los procesos y reduciendo la dependencia de procedimientos manuales propensos a errores (Ali & Osman, 2020).

6.3 Gestión documental y preservación digital

La gestión documental es un enfoque sistemático para controlar el ciclo de vida de los documentos, desde su creación hasta su conservación o eliminación (Andrade et al., 2019). Incluye políticas, procedimientos y tecnologías que aseguran que los documentos sean accesibles y coherentes.

La preservación digital, como parte integral de la gestión documental, se ocupa de mantener la accesibilidad de la información en el tiempo pese a los cambios tecnológicos y de formatos (Chen et al., 2019). Esto requiere el uso de formatos estandarizados, sistemas de respaldo y mecanismos de integridad que resguarden la información de la pérdida o corrupción.

En trabajos de grado, la preservación digital es esencial para garantizar que futuras generaciones de estudiantes e investigadores puedan acceder a los resultados académicos e información asociada con valor histórico o formativo.

6.4 Arquitectura web cliente–servidor

La arquitectura cliente–servidor es un patrón de diseño de sistemas distribuidos donde se separan las responsabilidades entre el cliente (interfaz de usuario) y el servidor (lógica de negocio y almacenamiento de datos) (Bass et al., 2015). Este modelo facilita la escalabilidad, modularidad y mantenimiento de sistemas, lo cual lo hace apropiado para aplicaciones web modernas.

En este esquema, el cliente realiza solicitudes al servidor, el cual procesa la información, aplica reglas de negocio y responde con los datos requeridos. Esto permite que cada capa evolucione de forma independiente, mejorando la flexibilidad del sistema y su capacidad para integrar servicios externos, como almacenamiento en la nube o servicios de autenticación (Richardson & Ruby, 2019).

6.5 Metodologías de desarrollo de software: modelo de prototipado

Las metodologías de desarrollo de software proporcionan marcos estructurados para planear, ejecutar y controlar proyectos tecnológicos (Laanti et al., 2016). Dentro de estas, el modelo de prototipado es un enfoque iterativo que genera versiones tempranas del producto para validar requisitos y obtener retroalimentación continua de los usuarios (Wang & Ahmad, 2018).

Este modelo es particularmente útil cuando los requisitos no están completamente definidos desde el inicio o cuando se requiere validar aspectos de usabilidad y funcionalidad con frecuencia (Sommerville, 2016). El prototipado permite ajustar y mejorar continuamente el producto a partir de la retroalimentación de los usuarios, lo que aumenta la probabilidad de que el sistema final cumpla adecuadamente con las necesidades reales.

7. Metodología

Para el desarrollo del proyecto se empleó la metodología de Prototipado, la cual permitió construir el sistema de manera iterativa mediante la elaboración, evaluación y refinamiento continuo de prototipos funcionales. Este enfoque resultó apropiado debido a la necesidad de validar progresivamente los requerimientos del sistema y ajustar la solución conforme se obtenía retroalimentación de los usuarios interesados.

En la primera etapa, se recopilaron y refinaron los requisitos iniciales del sistema, lo que permitió establecer el alcance funcional y técnico del repositorio digital. Posteriormente, se elaboró un diseño rápido, orientado a construir representaciones preliminares de la interfaz y de la estructura general del sistema. Estas representaciones sirvieron como guía para la construcción del primer prototipo funcional.

El prototipo inicial fue evaluado por los usuarios implicados –principalmente docentes-- con el fin de verificar su utilidad, claridad y cumplimiento de los requerimientos establecidos. Con base en las observaciones obtenidas, se realizaron ajustes estructurales, correcciones y mejoras en la interfaz, las funcionalidades y los procesos internos del sistema.

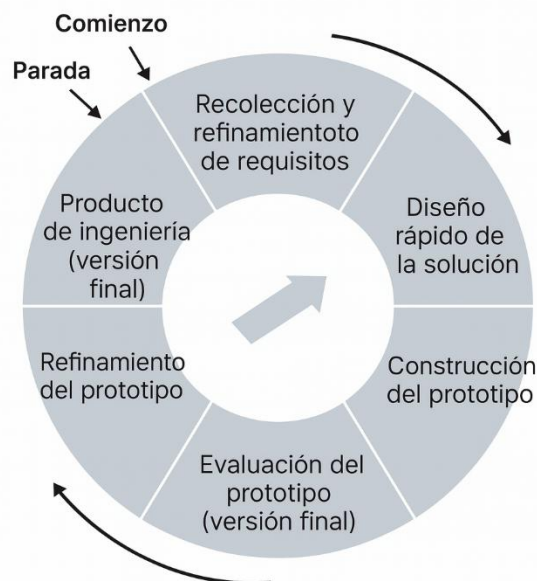
Este ciclo de construcción-evaluación-refinamiento se repitió hasta obtener un prototipo robusto que cumplió con los criterios de calidad, funcionalidad y usabilidad definidos para el proyecto. Finalmente, tras las iteraciones necesarias, se consolidó el producto de ingeniería, un sistema completamente funcional que cubrió las necesidades de almacenamiento, consulta y preservación digital de los trabajos de grado de la facultad.

El uso de la metodología de prototipado permitió disminuir la incertidumbre inicial del proyecto, aumentar la participación de los usuarios finales, corregir desviaciones antes de la implementación definitiva y asegurar que el sistema desarrollado respondiera efectivamente a las necesidades institucionales.

Con el fin de representar gráficamente el enfoque metodológico adoptado en el desarrollo del proyecto, en la **¡Error! No se encuentra el origen de la referencia.** se presenta el modelo de prototipado, el cual sirvió como base para la planificación y ejecución de las actividades. Este modelo permitió estructurar el proceso de construcción del repositorio digital mediante

ciclos iterativos de recolección de requisitos, diseño, construcción, evaluación y refinamiento, hasta alcanzar el producto final.

Figura 2. Modelo de desarrollo por prototipado aplicado al proyecto



Fuente: Elaboración propia.

En la **¡Error! No se encuentra el origen de la referencia.** se ilustra el modelo de prototipado aplicado al proyecto, el cual inicia con la recolección y refinamiento de los requisitos, fase en la que se identificaron las necesidades funcionales y no funcionales del repositorio digital a partir del análisis del proceso actual y la retroalimentación de los usuarios. Posteriormente, se desarrolló un diseño rápido de la solución, orientado a definir la estructura general del sistema, la organización de módulos y los primeros bocetos de interfaz.

A partir de este diseño se procedió a la construcción del prototipo, en la cual se implementaron versiones parciales del sistema que permitieron materializar las funcionalidades esenciales. Estas versiones fueron sometidas a una fase de evaluación, donde docentes y estudiantes interactuaron con el sistema y aportaron observaciones relacionadas con usabilidad, organización de la información y pertinencia de las funciones desarrolladas. Con base en los resultados de la evaluación, se realizó el refinamiento del prototipo, incorporando ajustes en la interfaz, mejoras en la lógica de negocio, optimización del modelo de datos y fortalecimiento de los mecanismos de seguridad. Este ciclo se repitió de manera

iterativa hasta consolidar el producto de ingeniería, correspondiente a la versión final del repositorio digital, completamente funcional y alineada con los requerimientos institucionales. La aplicación de este modelo permitió reducir la incertidumbre del proyecto, garantizar la participación de los usuarios finales y asegurar que la solución desarrollada respondiera efectivamente a las necesidades académicas identificadas.

7.1 Fases del modelo de prototipado aplicado al proyecto

El desarrollo del repositorio digital se estructuró en fases iterativas que integraron actividades de análisis, diseño, construcción y evaluación, las cuales se repitieron hasta consolidar el producto final. Las fases aplicadas fueron:

- Recopilación y análisis de requerimientos
- Diseño preliminar
- Construcción del prototipo
- Evaluación con usuarios
- Refinamiento
- Pruebas y validación

Cada fase generó entregables específicos que sirvieron como base para la siguiente iteración del proyecto.

7.2 Aplicación práctica del modelo de prototipado en el proyecto

Iteración 1: Prototipo inicial (definición y validación temprana)

En esta primera iteración se realizó el levantamiento de información con docentes y estudiantes, identificando los principales problemas asociados a la gestión de los trabajos de grado. A partir de este análisis se definieron los requerimientos funcionales y no funcionales iniciales, y se elaboraron representaciones preliminares del sistema, incluyendo bocetos de interfaz, diagramas de casos de uso y un modelo conceptual de datos.

Con base en este insumo se desarrolló un prototipo inicial enfocado en la estructura general del sistema, incorporando funcionalidades básicas como el acceso a la plataforma,

navegación principal y visualización general de trabajos. Este prototipo permitió validar la comprensión del problema, la organización de la información y la disposición de las pantallas.

Como resultado de esta iteración, se ajustaron los campos del registro de trabajos de grado, se redefinieron los roles de usuario y se reorganizó la estructura de navegación para separar claramente las funcionalidades administrativas de las funcionalidades de consulta.

Iteración 2: Prototipo funcional (consolidación de procesos)

En la segunda iteración se desarrollaron los módulos principales del sistema, incluyendo la gestión de usuarios, autenticación y autorización, el registro y edición de trabajos de grado, la administración de catálogos y la estructuración de la base de datos. Asimismo, se incorporaron validaciones de datos, control de sesiones y mecanismos de protección de rutas.

Este prototipo fue evaluado por usuarios interesados, quienes realizaron pruebas de uso y propusieron mejoras relacionadas con la facilidad de registro de información, el control de permisos y la organización visual de los resultados de búsqueda. Con base en esta retroalimentación, se optimizaron formularios, se fortaleció el control de roles y se incorporaron nuevos atributos para mejorar la clasificación de los trabajos.

Esta iteración permitió consolidar un sistema estable que cubría los procesos fundamentales del repositorio.

Iteración 3: Prototipo avanzado (optimización y validación final)

En la tercera iteración se integraron funcionalidades avanzadas como la carga y gestión de anexos, la búsqueda avanzada con filtros combinables, la paginación de resultados y la visualización detallada de trabajos de grado. Paralelamente, se realizaron pruebas de rendimiento, seguridad y calidad de código, con el fin de evaluar el comportamiento del sistema bajo condiciones controladas.

A partir de los resultados de estas pruebas se optimizaron consultas a la base de datos, se implementaron controles adicionales de validación y se fortalecieron los mecanismos de seguridad. Esta iteración concluyó con la consolidación del prototipo final, un sistema completamente funcional, alineado con los requerimientos institucionales y preparado para su uso en un entorno académico real.

8. Análisis y diseño del sistema

8.1 Análisis de requerimientos

El análisis de requerimientos tuvo como propósito identificar, estructurar y validar las necesidades funcionales y no funcionales del repositorio digital, con el fin de definir con precisión el alcance del sistema y servir como base para su diseño e implementación. Este proceso se realizó a partir del estudio del contexto institucional, la revisión de los procesos actuales de gestión de trabajos de grado y la interacción con usuarios potenciales, principalmente docentes y estudiantes.

Los requerimientos se documentaron mediante un Documento de Especificación de Requerimientos de Software (SRS), elaborado bajo lineamientos del estándar IEEE, el cual permitió formalizar las funcionalidades del sistema, las restricciones, los criterios de calidad y las reglas de negocio.

8.1.1 Requerimientos funcionales

Entre los principales requerimientos funcionales definidos se encuentran:

- RF-01. Permitir el registro y autenticación de usuarios.
- RF-02. Gestionar roles (administrador y usuario).
- RF-03. Registrar, consultar, actualizar y eliminar trabajos de grado.
- RF-04. Gestionar directores y codirectores.
- RF-05. Administrar catálogos académicos (líneas de investigación, niveles de formación, períodos, opciones de grado).
- RF-06. Permitir la carga y gestión de anexos asociados a los trabajos de grado.
- RF-07. Implementar un sistema de búsqueda avanzada con filtros combinables.
- RF-08. Generar visualización de estadísticas básicas del repositorio.

- RF-09. Restringir funcionalidades según el rol del usuario.

Estos requerimientos fueron validados durante las iteraciones del prototipo y sirvieron como base para el diseño de la arquitectura y los módulos del sistema.

8.1.2 Requerimientos no funcionales

Los requerimientos no funcionales definieron los atributos de calidad del sistema, entre los cuales se destacan:

Usabilidad: interfaz intuitiva, navegación clara y adaptación a distintos dispositivos.

Seguridad: autenticación mediante tokens, control de acceso por roles y protección de la información.

Rendimiento: tiempos de respuesta aceptables en consultas y búsquedas.

Escalabilidad: capacidad de soportar el crecimiento del volumen de trabajos y usuarios.

Mantenibilidad: arquitectura modular y uso de buenas prácticas de desarrollo.

Disponibilidad: acceso continuo al repositorio y protección ante pérdida de información.

8.2 Arquitectura del sistema

El repositorio digital fue diseñado bajo una arquitectura web cliente–servidor, estructurada en capas, con el objetivo de separar responsabilidades, facilitar el mantenimiento y permitir la escalabilidad del sistema. La arquitectura se compone de los siguientes niveles:

Capa de presentación (Frontend): responsable de la interacción con el usuario, visualización de información y captura de datos.

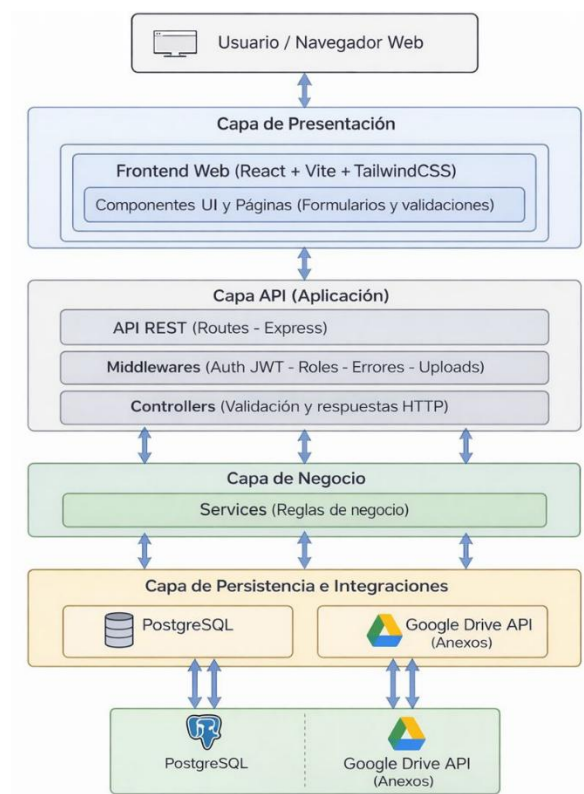
Capa de servicios (Backend): gestiona la lógica de negocio, autenticación, autorización, validaciones y exposición de servicios mediante una API REST.

Capa de persistencia: encargada del almacenamiento y recuperación de la información en la base de datos.

Servicios externos: integración con almacenamiento en la nube para la gestión de anexos.

Con el propósito de representar la estructura general del repositorio digital y la distribución de responsabilidades entre sus componentes, se diseñó una arquitectura por capas basada en el modelo cliente-servidor. En la Figura 3 se presenta el diagrama arquitectónico del sistema, el cual describe la interacción entre la capa de presentación, la capa de aplicación, la capa de negocio y la capa de persistencia e integraciones externas.

Figura 3. Arquitectura por capas del repositorio digital de trabajos de grado



Fuente: Elaboración propia.

En la Figura 3 se ilustra la arquitectura por capas del repositorio digital de trabajos de grado. En la parte superior se encuentra el usuario, quien accede al sistema a través de un navegador web. La interacción se realiza mediante la capa de presentación, implementada como una aplicación web, responsable de la visualización de la información, la captura de datos y la validación básica de formularios.

La capa de aplicación (API) actúa como intermediaria entre la interfaz y la lógica interna del sistema. En esta capa se gestionan los endpoints de la API REST, los controladores y los middlewares, los cuales permiten procesar las solicitudes, aplicar validaciones, controlar

errores y garantizar la seguridad mediante mecanismos de autenticación y autorización basados en roles.

La capa de negocio concentra las reglas fundamentales del sistema y los servicios encargados de orquestrar los procesos principales, tales como el registro y consulta de trabajos de grado, la gestión de usuarios, la administración de catálogos académicos y el manejo de anexos. Esta separación permitió desacoplar la lógica del sistema de los detalles de implementación, facilitando el mantenimiento y la escalabilidad.

Finalmente, la capa de persistencia e integraciones gestiona el almacenamiento y recuperación de la información. En esta capa se encuentra la base de datos relacional, encargada de preservar los registros académicos, así como la integración con un servicio externo de almacenamiento en la nube para la gestión de anexos. La comunicación entre la capa de negocio y esta capa garantiza la consistencia de los datos y la correcta vinculación de los documentos asociados a cada trabajo de grado.

Esta arquitectura permitió estructurar el sistema de manera modular, mejorar la mantenibilidad del software, reforzar la seguridad y facilitar la integración de nuevos servicios, asegurando que el repositorio digital respondiera de forma eficiente a los requerimientos institucionales.

8.3 Diseño del sistema

8.3.1 Diseño de módulos

El sistema fue organizado en módulos funcionales que agrupan responsabilidades específicas:

- Módulo de autenticación y usuarios
- Módulo de gestión de trabajos de grado
- Módulo de catálogos académicos
- Módulo de directores y codirectores
- Módulo de anexos
- Módulo de búsqueda y consulta
- Módulo administrativo

Esta división permitió estructurar el desarrollo, facilitar las pruebas y mejorar la mantenibilidad del software.

8.3.2 Diagramas del sistema

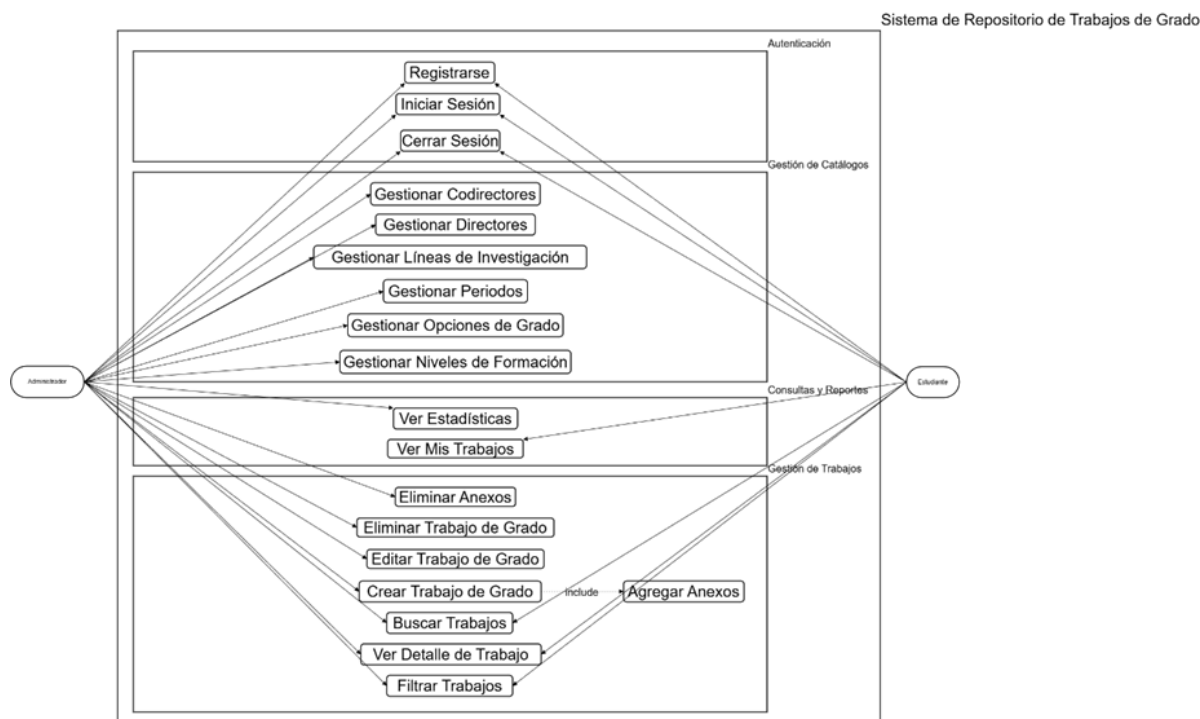
Para representar formalmente la estructura y el comportamiento del repositorio, se elaboraron diferentes diagramas UML:

a) Diagrama de casos de uso

Describe la interacción entre los actores (usuario y administrador) y el sistema. Permite identificar las funcionalidades principales, las restricciones por rol y los flujos generales de operación.

Con el fin de representar de manera gráfica las funcionalidades del repositorio digital y la interacción entre los diferentes tipos de usuarios y el sistema, se elaboró un diagrama de casos de uso. En la Figura 4 se presentan los principales procesos soportados por la plataforma, así como los actores involucrados en la gestión, consulta y administración de los trabajos de grado.

Figura 4. Diagrama de casos de uso del repositorio digital de trabajos de grado



Fuente: Elaboración propia.

En la Figura 4 se presenta el diagrama de casos de uso del repositorio digital, en el cual se identifican dos actores principales: Administrador y Usuario. Ambos interactúan con el sistema a través de los módulos de autenticación, gestión de trabajos y consulta de información, aunque con diferentes niveles de acceso y responsabilidad.

El actor Usuario puede registrarse, iniciar y cerrar sesión, consultar el repositorio mediante búsquedas y filtros, visualizar el detalle de los trabajos de grado y registrar nuevos trabajos, incluyendo la carga de anexos. Asimismo, tiene la posibilidad de editar la información de los trabajos que ha creado, lo que le permite mantener actualizados los registros asociados a sus proyectos académicos.

El actor Administrador, además de las funcionalidades disponibles para el usuario, dispone de permisos ampliados para la gestión integral del sistema. Entre sus principales casos de uso se encuentran la administración de catálogos académicos, tales como líneas de investigación, períodos académicos, opciones de grado y niveles de formación, así como la gestión de directores y codirectores. Estas funciones permiten garantizar la estandarización de la información y el correcto funcionamiento del repositorio.

Adicionalmente, el administrador cuenta con funcionalidades para la eliminación de trabajos de grado y anexos, y para la visualización de estadísticas del sistema, lo cual apoya los procesos de control, seguimiento académico y toma de decisiones institucionales.

El diagrama evidencia que los casos de uso relacionados con la gestión de trabajos de grado incluyen actividades como crear, editar, eliminar, buscar, filtrar y consultar el detalle de los registros. Asimismo, se observa la relación de inclusión entre la creación de trabajos de grado y la funcionalidad de agregar anexos, lo que indica que esta acción forma parte del proceso de registro completo de un trabajo.

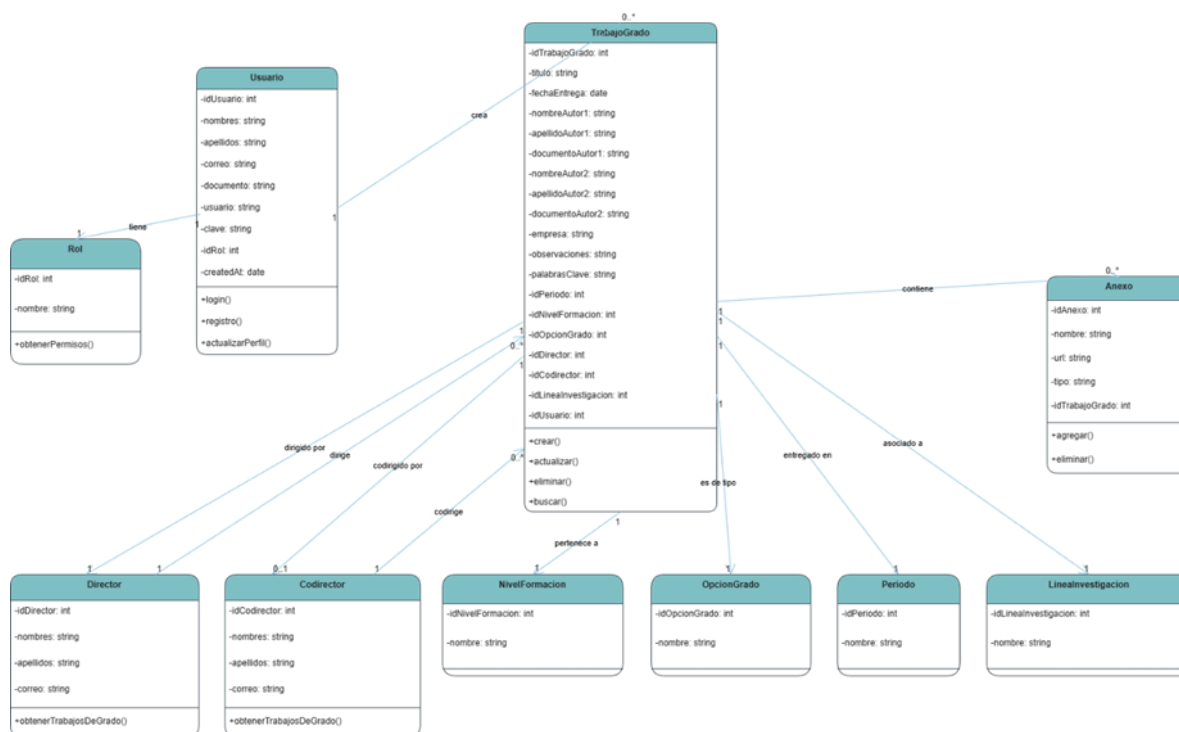
Este modelo permitió definir claramente el alcance funcional del sistema, establecer los permisos por tipo de usuario y servir como base para el diseño de los módulos del repositorio y la implementación de los endpoints de la API.

b) Diagrama de clases

Representa la estructura lógica del sistema, las entidades principales, sus atributos, métodos y relaciones. Este diagrama sirvió como base para la implementación de los modelos del sistema y la definición de relaciones en la base de datos.

Con el fin de representar la estructura lógica del sistema, las entidades principales, sus atributos y las relaciones entre ellas, se elaboró el diagrama de clases que se presenta en la Figura 5. Este modelo permitió definir formalmente los componentes fundamentales del repositorio digital y sirvió como base para la implementación de los modelos de datos y la lógica de negocio del sistema.

Figura 5. Diagrama de clases del repositorio digital de trabajos de grado



Fuente: Elaboración propia.

En la Figura 5 se presenta el diagrama de clases del repositorio digital, en el cual se identifican las entidades principales que conforman el sistema y sus relaciones. La clase Usuario representa a las personas que interactúan con la plataforma y contiene los atributos necesarios para su identificación, autenticación y control de estado. Esta clase se asocia con la clase Rol, la cual define los niveles de acceso y permisos dentro del sistema.

La clase central del modelo es TrabajoGrado, ya que concentra la información académica principal del repositorio. Esta clase almacena los datos generales del trabajo, como título, resumen, palabras clave, fechas y estado, y se relaciona con múltiples entidades que permiten su correcta clasificación. Entre estas se encuentran Director y Codirector, que representan los responsables académicos del proyecto; NivelFormación, OpciónGrado, Periodo y LíneaInvestigación, que estructuran los criterios académicos y temporales de los trabajos de grado.

Asimismo, la clase TrabajoGrado mantiene una relación directa con la clase Anexo, que modela los archivos asociados a cada trabajo, tales como documentos, software o material multimedia. Esta relación permite que un trabajo de grado pueda contar con múltiples anexos, garantizando la preservación y vinculación de los productos derivados del proyecto.

El diagrama también incluye los métodos principales asociados a cada clase, los cuales representan las operaciones que el sistema realiza sobre las entidades, tales como creación, consulta, actualización y eliminación de registros. Este modelo orientó la definición de las tablas de la base de datos, las relaciones entre ellas y la implementación de los servicios del backend.

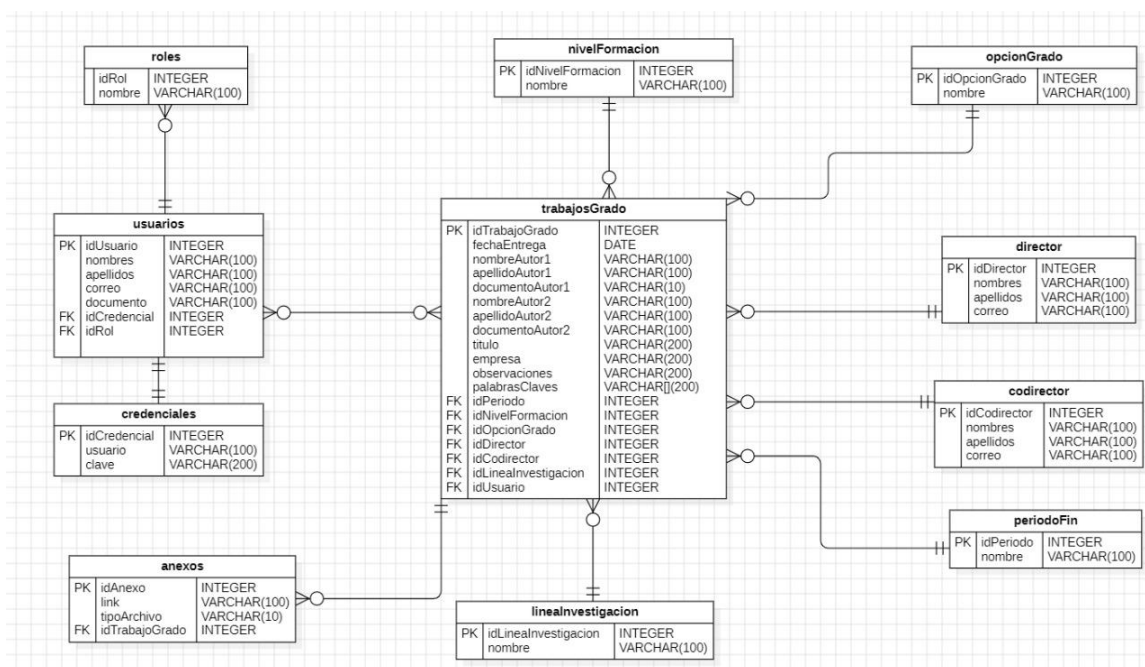
En conjunto, el diagrama de clases permitió establecer una visión clara de la estructura interna del sistema, asegurar la coherencia entre los módulos desarrollados y garantizar que la solución implementada respondiera de manera consistente a los requerimientos funcionales definidos.

c) Diagrama entidad–relación

Modela la estructura de la base de datos, describiendo las tablas, claves primarias, claves foráneas y cardinalidades. Este diagrama permitió garantizar integridad referencial y correcta organización de la información.

Con el objetivo de definir la estructura de la base de datos y las relaciones entre las entidades que soportan el funcionamiento del repositorio digital, se elaboró el diagrama entidad–relación que se presenta en la Figura 6. Este modelo permitió identificar las tablas principales, sus atributos, claves primarias y foráneas, así como las cardinalidades necesarias para garantizar la integridad y consistencia de la información almacenada.

Figura 6. Diagrama entidad–relación de la base de datos del repositorio digital



Fuente: Elaboración propia.

En la Figura 6 se presenta el diagrama entidad–relación de la base de datos del repositorio digital. El modelo se estructuró a partir de la entidad central trabajosgrado, la cual almacena la información principal de cada trabajo de grado, incluyendo datos generales, autores, fecha de entrega y metadatos académicos.

Esta entidad se relaciona con múltiples tablas de tipo catálogo, entre las que se encuentran nivelformacion, opciongrado, periodoFin y lineaInvestigacion, las cuales permiten clasificar los trabajos de grado de acuerdo con criterios académicos y temporales. Estas relaciones garantizan la estandarización de la información y facilitan la implementación de mecanismos de búsqueda y filtrado.

Asimismo, trabajosgrado se vincula con las entidades director y codirector, que almacenan la información de los responsables académicos del proyecto, permitiendo registrar y consultar los trabajos asociados a cada uno.

La entidad anexos se relaciona con trabajosgrado mediante una relación uno a muchos, lo que permite asociar múltiples archivos a un mismo trabajo de grado. Esta estructura soporta la gestión de documentos complementarios, tales como informes, software o material audiovisual.

El modelo también contempla las entidades usuarios, roles y credenciales, las cuales soportan los procesos de autenticación, autorización y control de acceso al sistema. La relación entre usuarios y roles define los permisos dentro de la plataforma, mientras que la tabla de credenciales almacena de forma segura la información necesaria para la validación de identidad.

En conjunto, este modelo de datos permitió asegurar la integridad referencial mediante el uso de claves primarias y foráneas, evitar la duplicidad de información, facilitar la escalabilidad del sistema y servir como base para la implementación de la capa de persistencia del repositorio digital.

8.4 Diseño de seguridad

La seguridad del sistema fue considerada un aspecto crítico debido al carácter académico e institucional de la información gestionada. Por esta razón, se diseñó un esquema de seguridad basado en los siguientes principios:

- **Autenticación:** validación de identidad mediante credenciales y generación de tokens de sesión.
- **Autorización:** control de acceso a funcionalidades según el rol del usuario.
- **Protección de datos:** cifrado de contraseñas, validación de entradas y control de archivos.
- **Seguridad de servicios:** protección de endpoints, control de sesiones y manejo de errores.

Se establecieron niveles de acceso diferenciados para usuarios y administradores, garantizando que únicamente personal autorizado pudiera realizar operaciones críticas como eliminación de registros o gestión de catálogos.

8.5 Modelo de datos

El modelo de datos fue diseñado para garantizar integridad, escalabilidad y consistencia de la información. Se estructuró en entidades principales que representan los elementos fundamentales del repositorio:

- Usuarios
- Roles
- Credenciales
- Trabajos de grado
- Directores
- Codirectores
- Anexos
- Niveles de formación
- Opciones de grado
- Períodos académicos
- Líneas de investigación

Las relaciones establecidas permitieron vincular cada trabajo de grado con su información académica, sus autores, sus responsables y sus productos asociados.

9. Implementación del sistemas

9.1 Consideraciones generales de implementación

La implementación del repositorio digital se realizó a partir del diseño definido en la fase de análisis, siguiendo la arquitectura cliente–servidor organizada por capas. El desarrollo se orientó a la construcción de un sistema modular, mantenible y escalable, que permitiera gestionar de manera eficiente los trabajos de grado y sus productos asociados.

El sistema fue implementado como una aplicación web distribuida, compuesta por una capa de presentación (frontend), una capa de aplicación (backend), una capa de negocio y una capa de persistencia, integrando además servicios externos para la gestión de anexos.

9.2 Implementación del backend

La capa backend fue desarrollada como una API REST, encargada de gestionar la lógica del sistema, la validación de datos, la seguridad, el acceso a la base de datos y la integración con servicios externos.

Se estructuró el backend en módulos que separan claramente responsabilidades:

- **Rutas:** definición de los endpoints y exposición de los servicios.
- **Controladores:** recepción de solicitudes, validación inicial y construcción de respuestas HTTP.
- **Servicios:** implementación de las reglas de negocio y orquestación de procesos.
- **Middlewares:** gestión de autenticación, autorización, validación de entradas, manejo de errores y carga de archivos.
- **Modelos:** representación de las entidades del sistema y sus relaciones.

Entre las principales funcionalidades implementadas en esta capa se encuentran:

- Autenticación de usuarios y generación de tokens de sesión.
- Control de acceso basado en roles.
- Gestión completa de trabajos de grado (registro, consulta, edición y eliminación).
- Administración de catálogos académicos.
- Gestión de directores y codirectores.
- Registro y control de anexos.
- Implementación de búsquedas avanzadas con filtros y paginación.

La API se diseñó para responder mediante estructuras de datos normalizadas, facilitando la comunicación con el frontend y garantizando la consistencia de la información.

9.3 Implementación de frontend

La capa de presentación se desarrolló como una aplicación web interactiva, orientada a facilitar el acceso, la consulta y la gestión de la información académica. Su implementación se enfocó en la construcción de interfaces claras, coherentes y usables.

El frontend se organizó en:

- **Componentes reutilizables:** botones, tablas, formularios, modales y notificaciones.
- **Páginas:** vistas principales para autenticación, consulta del repositorio, gestión de trabajos y panel administrativo.

- **Servicios de comunicación:** encargados del consumo de la API REST.
- **Gestión de estado:** control de sesión, datos de usuario y resultados de búsqueda.

Se implementaron interfaces diferenciadas según el rol del usuario, garantizando que las opciones disponibles correspondieran a sus permisos. Asimismo, se incorporaron mecanismos de validación en formularios, retroalimentación visual, manejo de errores y confirmaciones de acciones críticas.

La interfaz permitió realizar de manera intuitiva procesos como:

- Registro e inicio de sesión.
- Búsqueda y filtrado de trabajos de grado.
- Visualización detallada de registros.
- Creación y edición de trabajos.
- Gestión de anexos.
- Administración de catálogos y usuarios (rol administrador).

9.4 Integraciones externas

Como parte de la implementación del sistema, se integró un servicio externo de almacenamiento en la nube para la gestión de anexos asociados a los trabajos de grado.

Esta integración permitió:

- Almacenar archivos de gran tamaño sin sobrecargar el servidor principal.
- Centralizar los anexos en un repositorio seguro.
- Mantener en la base de datos únicamente los metadatos y referencias de los archivos.

El backend se encargó de gestionar la comunicación con el servicio externo, incluyendo la validación de archivos, el control de tipos y tamaños permitidos, y el registro de los identificadores correspondientes. De esta manera, el sistema garantizó la vinculación adecuada entre los trabajos de grado y sus productos asociados.

9.5 Decisiones técnicas relevantes

Durante la implementación se tomaron decisiones técnicas orientadas a mejorar la calidad, seguridad y sostenibilidad del sistema, entre las cuales se destacan:

- Separación por capas, para desacoplar la interfaz, la lógica de negocio y la persistencia.
- Uso de autenticación basada en tokens, para garantizar sesiones seguras y control de acceso.

- Modularización del backend, facilitando el mantenimiento y la ampliación de funcionalidades.
- Normalización del modelo de datos, evitando duplicidad de información y garantizando integridad referencial.
- Uso de servicios externos para anexos, reduciendo riesgos de pérdida de información y mejorando la escalabilidad.
- Implementación de validaciones en frontend y backend, para asegurar consistencia de datos y prevenir errores.
- Estandarización de respuestas de la API, facilitando la integración entre capas.

Estas decisiones permitieron construir un sistema robusto, alineado con los requerimientos institucionales y preparado para su evolución futura.

10. Cumplimiento de objetivos y desarrollo realizado

El presente capítulo describe de manera detallada las actividades realizadas para dar cumplimiento a cada uno de los objetivos específicos del proyecto, así como los resultados obtenidos, las evidencias generadas y los mecanismos utilizados para su verificación. Esta sección permite demostrar la correspondencia entre los objetivos planteados y el producto de ingeniería desarrollado.

10.1 Cumplimiento del objetivo 1: Análisis de requerimientos

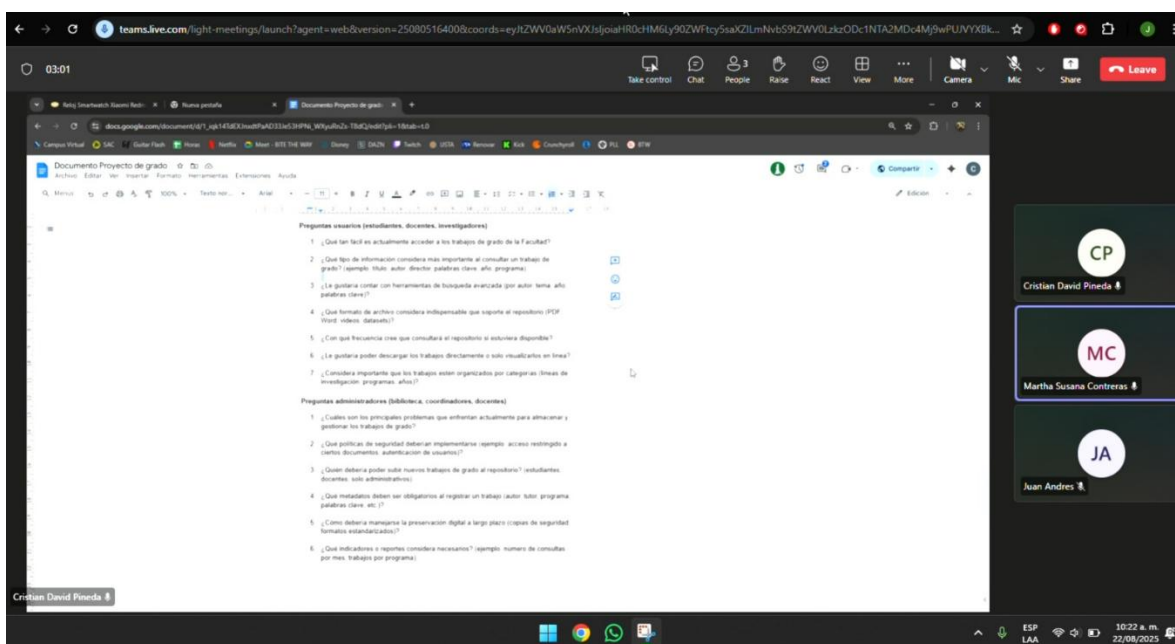
Objetivo: Analizar los requerimientos funcionales y no funcionales del repositorio mediante el uso del estándar IEEE 830 como formato principal para la redacción del SRS, con el fin de asegurar que cumpla con las necesidades de los usuarios, los estándares de almacenamiento y las políticas de acceso a la información, garantizando su eficiencia y seguridad.

Actividades realizadas y evidencias

Con el fin de dar cumplimiento a este primer objetivo, como primer paso se realizó un proceso de investigación en el cual se identificaron los principales actores y procesos académicos que hacen parte de la elaboración de un trabajo de grado. Una vez identificadas las partes involucradas, se inició un proceso de recolección de información, revisando el proceso actual, más específicamente el repositorio del CRAI de la Universidad Santo Tomás, analizando sus interfaces, la forma de representar la información, los mecanismos de

búsqueda y los demás componentes que lo integran. Adicionalmente, se llevaron a cabo consultas tanto a docentes como a estudiantes, lo que permitió dar inicio al proceso de levantamiento de requerimientos. Una de las entrevistas se realizó con la ingeniera Martha Susana Contreras Ortiz, en su calidad de docente de la Facultad de Ingeniería de Sistemas, en la cual se evidenciaron las principales necesidades para el desarrollo de este repositorio, como se puede observar en la Figura 7.

Figura 7. Entrevista de recolección de requerimientos para el repositorio digital



Fuente: Elaboración propia

Una vez recolectados los requerimientos, se inició el proceso de definición y documentación de estos, clasificándolos en funcionales y no funcionales, para finalmente elaborar el documento de Especificación de Requerimientos de Software (SRS), el cual se puede encontrar en el **ANEXO 1**. Este proceso se realizó con la intención de utilizar un estándar como el IEEE 830, con el fin de asegurar que se cumplieran las necesidades de los usuarios del repositorio, garantizando la eficiencia y seguridad de los datos almacenados. Cabe aclarar que, debido a la metodología de prototipado utilizada, en cada iteración los requerimientos fueron validados y refinados.

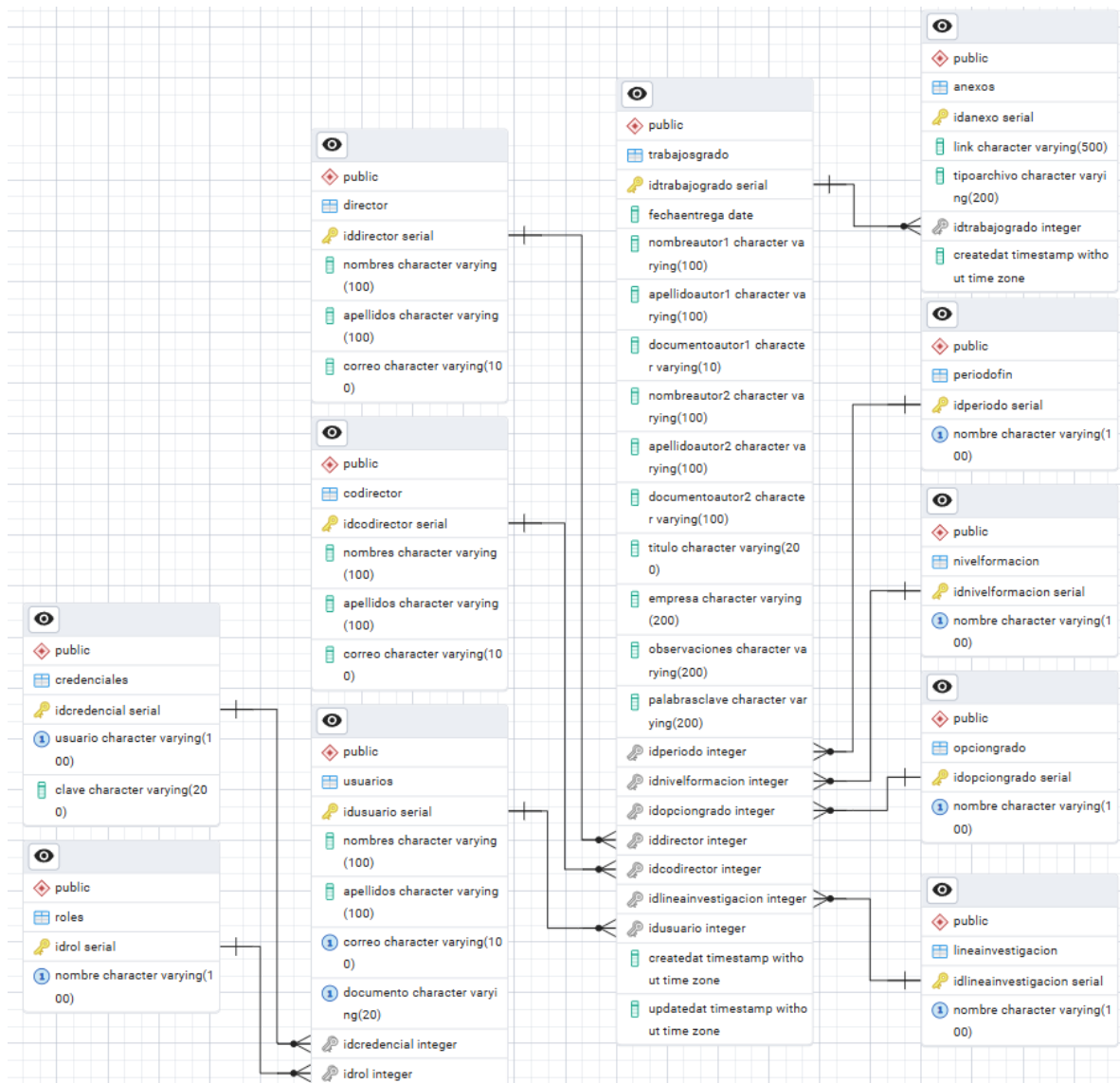
10.2 Cumplimiento del objetivo 2: Implementación del repositorio digital

Objetivo: Implementar una plataforma web que facilite la carga, gestión y consulta de los trabajos de grado, asegurando una estructura de datos organizada y un acceso intuitivo a la información mediante herramientas de búsqueda y clasificación.

Actividades realizadas y evidencias

Para esta etapa se analizaron los requerimientos ya establecidos y luego de una fase de investigación se logró encontrar que la arquitectura más adecuada para este proyecto era una arquitectura cliente - servidor por capas como se muestra en la Figura 3 esto por la necesidad de escalabilidad, seguridad y mantenimiento y el acceso de múltiples usuarios, así mismo la organización por capas (presentación, lógica de negocio y acceso a datos) favoreció una mejor estructuración del software, haciendo posible aislar responsabilidades específicas. Continuo a esto se empezó con la etapa de modelado, donde se realizaron tanto el modelo Entidad – Relación (ER) mostrado en la Figura 6 y el modelo de clases mostrado en la Figura 5, esto para diseñar la estructura de la base de datos, la estructura interna del sistema e identificar las entidades principales, sus atributos, las relaciones entre ellas y las cardinalidades, lo cual permitió organizar de manera correcta la información, garantizar la integridad de los datos y servir como base para la implementación de la base de datos relacional del repositorio y permitió definir las clases del software, sus atributos, métodos y relaciones, facilitando la organización del código, la implementación del backend y la correcta distribución de responsabilidades dentro del sistema, dando así momento para decidir que motor de base de datos se utilizaría para el proyecto dando como resultado la elección de PostgreSQL dando paso a la implementación de la base de datos relacional mostrada en la siguiente figura.

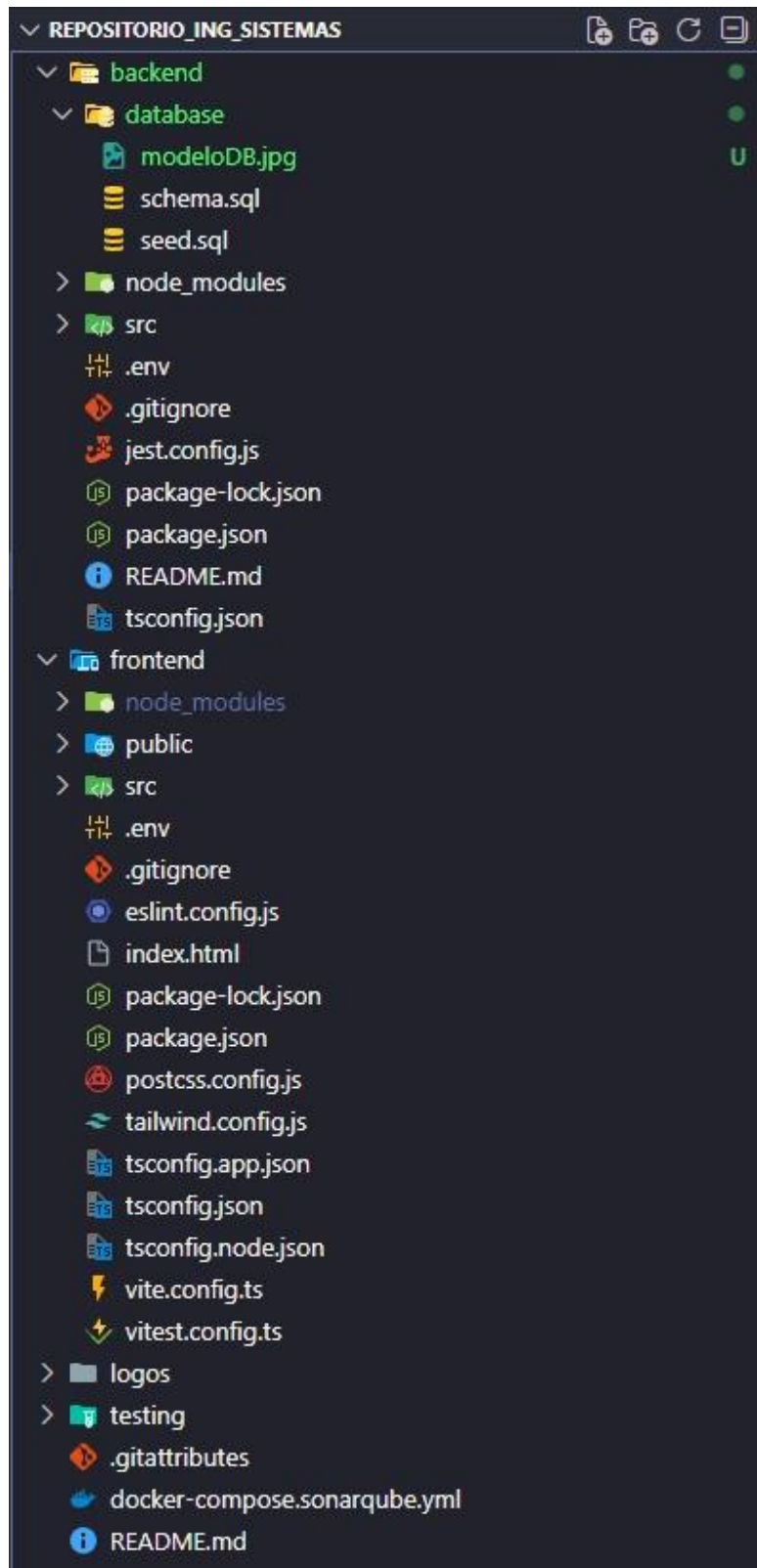
Figura 8. Esquema relacional de la base de datos



Fuente: Elaboración propia

Una vez esto terminado se dio paso al inicio del desarrollo del backend bajo el enfoque de servicios mediante una API REST encargada de gestionar la comunicación entre el frontend y la base de datos, así como de centralizar las reglas de negocio y los mecanismos de seguridad. Para el backend fueron seleccionadas tecnologías como TypeScript como lenguaje Node.js como entorno de ejecución, Express como framework web y JWT (JSON Web Token), Bcrypt para temas de autenticación y encriptación como métodos de seguridad. Esto por ser tecnologías complementarias entre ellas, posibilidad de escalabilidad, robustez y conocimiento de manejo dando como resultado una estructura de proyecto mostrada en la Figura 9.

Figura 9. Estructura del proyecto repositorio digital



Fuente: Elaboración propia

Simultáneamente se iba desarrollando el frontend, el cual fue realizado con tecnologías modernas como lo son React, Vite para la entorno de desarrollo y TailwindCSS como framework esto para tener un rendimiento adecuado, una experiencia de usuario agradable y tecnologías que permitan fácil actualización ya que está basado en componentes como lo podemos ver en las siguientes figuras.

Figura 10. Interfaz principal del repositorio digital de trabajos de grado

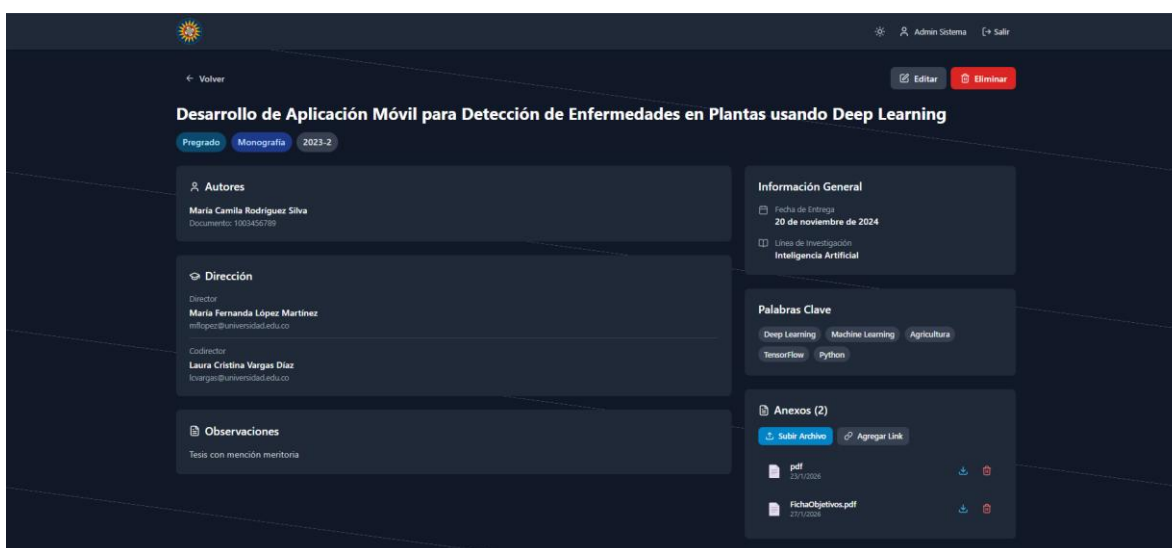


Fuente: Elaboración propia

Esta pantalla tiene como propósito brindar una presentación general del sistema y orientar al usuario sobre las funcionalidades disponibles, tales como el acceso al catálogo, la búsqueda avanzada, la visualización de estadísticas y la interacción con la comunidad académica.

En esta interfaz se integran elementos de navegación que permiten al usuario registrarse o iniciar sesión, lo cual constituye el punto de entrada a las funcionalidades completas del repositorio.

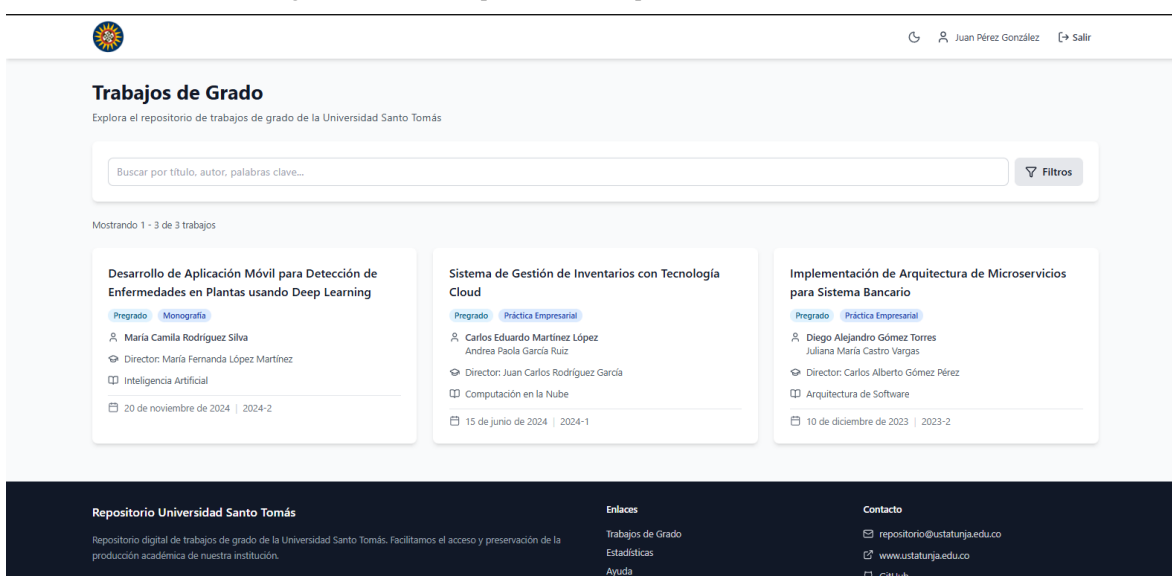
Figura 11. Vista detallada de un trabajo de grado desde el rol usuario



Fuente: Elaboración propia

En esta interfaz se presenta de forma estructurada toda la información asociada al trabajo, incluyendo título, nivel de formación, opción de grado y período académico. Asimismo, el sistema organiza los datos en secciones claramente diferenciadas: autores, dirección (director y codirector), información general, palabras clave, observaciones y anexos. Esta disposición facilita la lectura, comprensión y recuperación de la información académica.

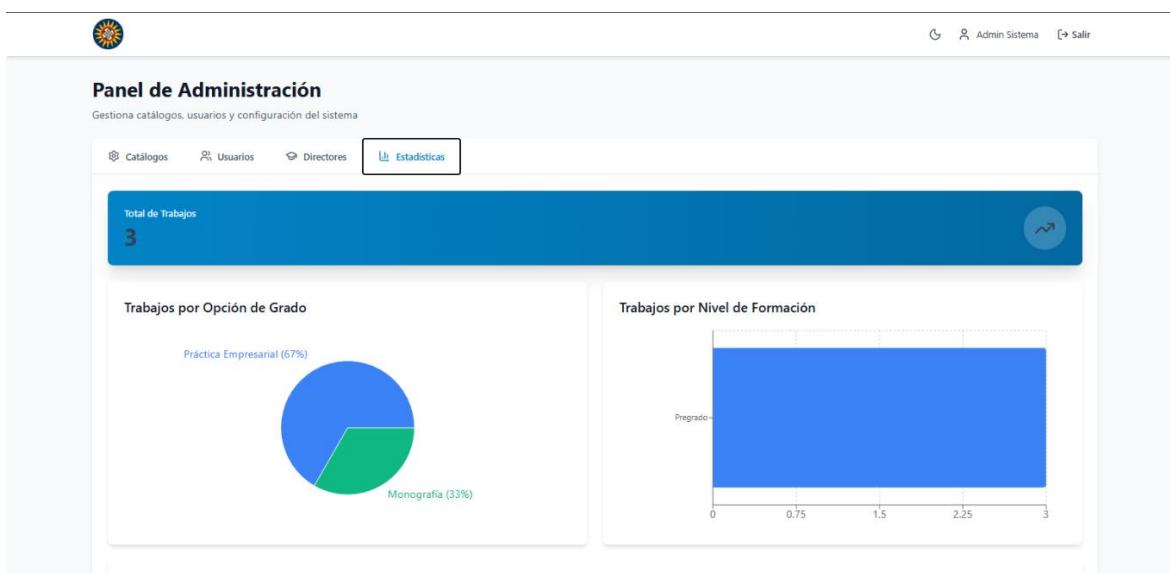
Figura 12. Vista de exploración del repositorio desde el rol usuario



Fuente: Elaboración propia

Esta vista permite consultar los trabajos de grado registrados, presentando la información de forma organizada y accesible. En esta pantalla se dispone un campo de búsqueda avanzada que permite localizar trabajos por título, autor o palabras clave, así como un conjunto de tarjetas que muestran los trabajos disponibles con información relevante como autores, directores, línea de investigación, opción de grado y período académico.

Figura 13. Panel de administración – Módulo de visualización de estadísticas del repositorio



Fuente: Elaboración propia

Esta interfaz permite al administrador visualizar información consolidada sobre los trabajos de grado almacenados en el sistema, facilitando el análisis de la producción académica institucional. En esta sección se presentan indicadores generales, como el total de trabajos registrados, así como gráficos que representan la distribución de los trabajos por opción de grado y por nivel de formación. Estas visualizaciones permiten identificar tendencias, áreas de mayor producción y características relevantes de los trabajos de grado.

Y todo esto dio como resultado una plataforma web que permite la carga, gestión y consulta de los trabajos de grado, asegurando una estructura de datos organizada y un acceso intuitivo a la información mediante herramientas de búsqueda y clasificación.

10.3 Cumplimiento de objetivo 3: Evaluación del sistema

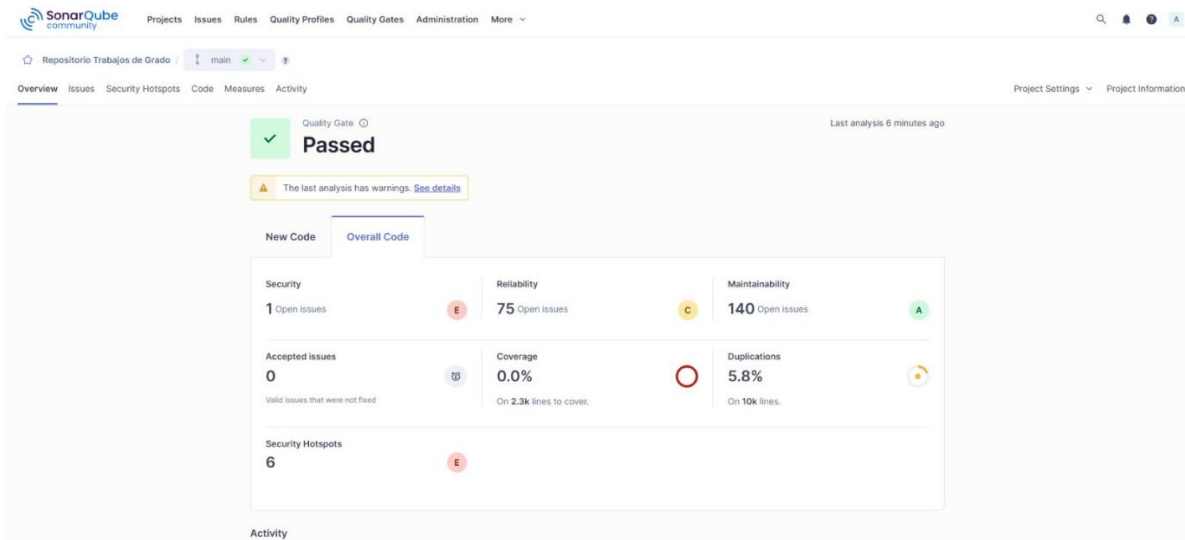
Objetivo: Evaluar el desempeño del software de acuerdo con usabilidad, seguridad y desempeño, mediante pruebas de carga y de calidad en la codificación, con el uso de herramientas como sonarQube, JMeter o Selenium.

Actividades realizadas y evidencias

Para dar cumplimiento a este objetivo, se desarrolló un proceso de evaluación del desempeño y la calidad del software, orientado principalmente a los atributos de usabilidad, seguridad y desempeño, mediante el análisis estructural del código fuente y la verificación de buenas prácticas de desarrollo. En una fase inicial, se realizó una revisión de herramientas comúnmente empleadas para este tipo de evaluaciones, entre ellas JMeter, Selenium y SonarQube. Sin embargo, teniendo en cuenta la naturaleza del proyecto —una plataforma web para la gestión y consulta de trabajos de grado— se determinó que SonarQube era la herramienta más adecuada, dado que permite realizar análisis profundo del código, detectar vulnerabilidades, medir la calidad interna del software y evaluar aspectos directamente relacionados con la seguridad, confiabilidad y mantenibilidad, sin necesidad de un entorno complejo de simulación de usuarios o automatización de interfaces.

Una vez seleccionada la herramienta, se procedió a la instalación y configuración de SonarQube Community Edition en un entorno local, creando un proyecto específico para el repositorio digital. Posteriormente, se integró el backend y el frontend al proceso de análisis mediante la ejecución del escáner de SonarQube, lo que permitió evaluar de forma conjunta la totalidad del sistema. Como resultado de este proceso, se generaron reportes automáticos que incluyen métricas de seguridad, confiabilidad, mantenibilidad, hotspots de seguridad, duplicación de código, tamaño del proyecto y deuda técnica, los cuales pueden observarse en la Figura 14 correspondientes al panel principal de resultados.

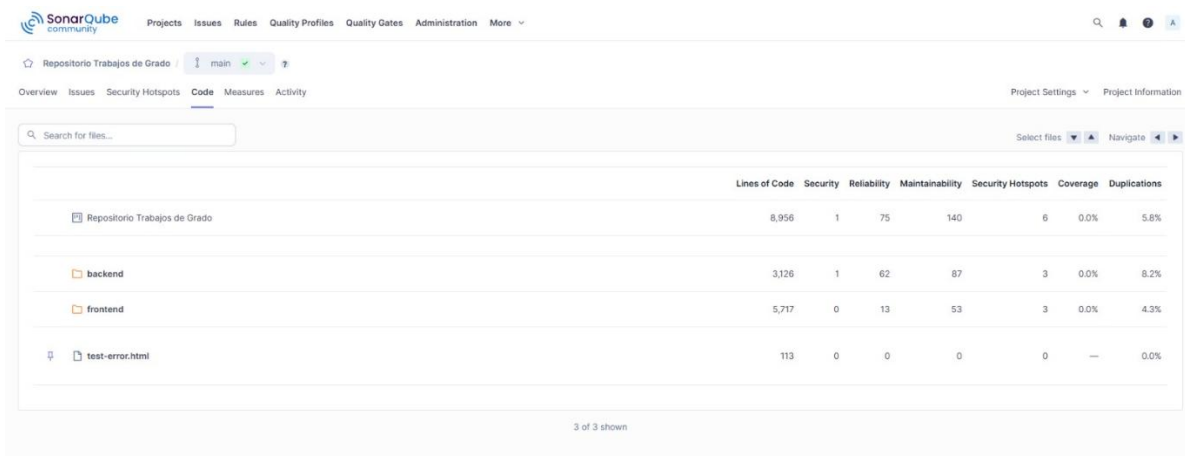
Figura 14. Panel principal de resultados de evaluación



Fuente: Elaboración propia

Los resultados obtenidos evidencian que el sistema superó el Quality Gate definido por la herramienta, lo cual se observa en la vista general del proyecto, donde se indica el estado “Passed”, validando que el software cumple con los criterios mínimos de calidad establecidos. Asimismo, se identificaron y clasificaron advertencias relacionadas con consistencia, mantenibilidad y confiabilidad. En la Figura 15 se presenta el resumen general del análisis, donde se visualiza el número total de líneas de código evaluadas, los problemas detectados por categoría y el porcentaje de duplicación, lo que permitió verificar que el proyecto mantiene una estructura organizada y con bajo nivel de redundancia.

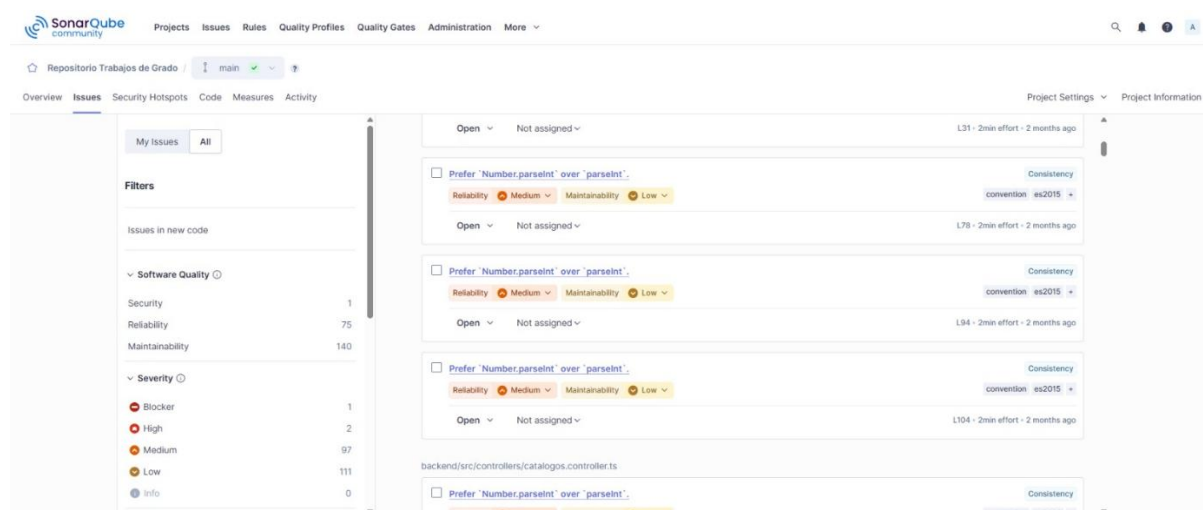
Figura 15. Resumen general del análisis por código



Fuente: Elaboración propia

De igual forma, a través de la sección de métricas y análisis de riesgo de la Figura 16, fue posible evaluar la distribución de la deuda técnica, la cobertura, la criticidad de los problemas detectados y su impacto sobre la seguridad y la confiabilidad del sistema. Estas visualizaciones facilitaron la interpretación del estado real del software, evidenciando que, si bien existen observaciones propias de un sistema en evolución, no se identificaron vulnerabilidades críticas que comprometieran la integridad o confidencialidad de la información. Además, los hotspots de seguridad permitieron revisar secciones sensibles del código, especialmente aquellas relacionadas con autenticación y validación de datos.

Figura 16. Sección de métricas y análisis de riesgo



Fuente: Elaboración propia

Este proceso de evaluación no solo permitió medir objetivamente la calidad del software, sino que también se utilizó como una herramienta de apoyo para la toma de decisiones durante las etapas finales de desarrollo, promoviendo la mejora continua del código y la adopción de buenas prácticas. En consecuencia, mediante el uso de SonarQube y el análisis de los resultados reflejados en las evidencias presentadas, se logró validar que el sistema cumple con condiciones adecuadas de seguridad, desempeño estructural y mantenibilidad, demostrando de forma objetiva el cumplimiento del objetivo planteado.

11. Conclusiones

Se desarrolló e implementó un repositorio digital de trabajos de grado para la Facultad de Ingeniería de Sistemas de la Universidad Santo Tomás – Tunja, que permite la gestión, almacenamiento y consulta centralizada de la producción académica, cumpliendo con el objetivo general planteado.

El sistema integra funcionalidades de registro, búsqueda avanzada, visualización detallada, gestión de anexos, administración de catálogos académicos y generación de estadísticas, consolidándose como una herramienta institucional para la preservación y visibilidad del conocimiento.

La implementación de catálogos académicos (niveles de formación, opciones de grado, períodos y líneas de investigación) permitió estandarizar la información almacenada en el repositorio, mejorando la organización, consistencia y trazabilidad de los trabajos de grado. Esto facilita los procesos de consulta, clasificación y análisis, reduciendo la duplicidad de esfuerzos y fortaleciendo el control académico.

El sistema incorpora un modelo de control de acceso basado en roles, diferenciando funcionalidades entre usuarios y administradores. Esta característica permitió garantizar la seguridad de la información, la correcta gestión de privilegios y la protección de los datos académicos almacenados.

La arquitectura cliente–servidor implementada, junto con el uso de tecnologías modernas de desarrollo web, permitió construir una plataforma escalable, mantenible y orientada a servicios, facilitando futuras extensiones del sistema.

El repositorio no solo cumple una función de almacenamiento, sino que se proyecta como un sistema de apoyo para la gestión académica y el análisis de la producción investigativa.

El repositorio digital desarrollado contribuye directamente al fortalecimiento de la visibilidad institucional, al facilitar el acceso a los trabajos de grado y promover su consulta por parte de estudiantes, docentes e investigadores.

Asimismo, establece una base tecnológica para la preservación digital de la memoria académica de la facultad.

11.1 Limitaciones del proyecto

A pesar de los resultados obtenidos, el proyecto presenta algunas limitaciones:

- El sistema fue implementado y validado en un entorno local académico controlado, por lo que no se evaluó su comportamiento bajo escenarios de alta concurrencia o cargas masivas de usuarios.

- Actualmente, los reportes estadísticos se enfocan en métricas básicas, sin incorporar análisis avanzados o generación automática de informes académicos.

11.2 Trabajos futuros y propuestas de mejora

Este punto es clave para mostrar que tu proyecto no se queda en una entrega, sino que tiene proyección real.

Escalabilidad institucional

Extender el repositorio digital para cubrir todas las facultades de la universidad, permitiendo su adopción como plataforma institucional de gestión de trabajos de grado.

Mejoras en búsqueda y análisis

Implementar motores de búsqueda avanzados con indexación semántica, filtrado inteligente y recomendaciones automáticas basadas en palabras clave y líneas de investigación.

Integración con plataformas institucionales

Integrar el sistema con plataformas académicas existentes, como sistemas de matrícula, bibliotecas digitales o repositorios institucionales, para automatizar el flujo de información.

Módulo de reportes académicos

Incorporar un módulo de generación de reportes exportables (PDF, Excel) que facilite la toma de decisiones por parte de coordinaciones académicas y comités curriculares.

Optimización de seguridad

Incluir mecanismos adicionales de seguridad como doble factor de autenticación, auditoría de acciones y monitoreo de accesos.

Referencias

Albert, W., & Tullis, T. (2019). *Measuring the user experience: Collecting, analyzing, and presenting usability metrics* (3rd ed.). Morgan Kaufmann.

Balaji, S., & Murugaiyan, M. S. (2018). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology & Business Management*, 30(1), 26–30.

Borgman, C. L. (2015). *Big data, little data, no data: Scholarship in the networked world*. MIT Press.

Chakraborty, S., & Bose, A. (2020). Web API security: Best practices and patterns. *Journal of Software Engineering Research and Development*, 7(4), 1–12.

Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of database systems* (7th ed.). Pearson.

ISO/IEC. (2017). *ISO/IEC 25010:2011/AMD1:2017 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE)*. International Organization for Standardization.

Lago, P., Soldani, J., Giallorenzo, S., & De Roover, C. (2019). Microservices: A systematic mapping study. *Journal of Systems and Software*, 150, 77–97. <https://doi.org/10.1016/j.jss.2019.01.016>

Lynch, C. A. (2016). Digital libraries and the challenges of digital preservation. *Communications of the ACM*, 59(7), 50–56. <https://doi.org/10.1145/2942426>

Myers, G. J., Sandler, C., & Badgett, T. (2019). *The art of software testing* (3rd ed.). Wiley.

Pressman, R. S., & Maxim, B. R. (2019). *Software engineering: A practitioner's approach* (8th ed.). McGraw-Hill Education.

Project Management Institute. (2021). *A guide to the project management body of knowledge (PMBOK guide)* (7th ed.). PMI.

Richardson, L., & Ruby, S. (2018). *RESTful web APIs: Services for a changing world* (2nd ed.). O'Reilly Media.

Sharma, T., & Jangra, A. (2018). Security issues and best practices in web application development. *International Journal of Computer Applications*, 182(31), 1–6.

Sommerville, I. (2015). *Software engineering* (10th ed.). Pearson.

Taylor, R. N., Medvidovic, N., & Dashofy, E. M. (2019). *Software architecture: Foundations, theory, and practice* (2nd ed.). Wiley.

VersionOne. (2020). 14th annual state of agile report. <https://stateofagile.com>

ANEXO 1

Especificación de requerimientos de software (SRS)

Repositorio digital de trabajos de grado para la facultad de Ingeniería de Sistemas – USTA Tunja

1. Introducción

1.1 Propósito

El presente documento tiene como propósito especificar de manera formal los requerimientos funcionales y no funcionales del sistema “Repositorio Digital de Trabajos de Grado”, desarrollado para la Facultad de Ingeniería de Sistemas de la Universidad Santo Tomás – Seccional Tunja. Este documento sirve como base para el diseño, implementación, validación y mantenimiento del sistema.

El SRS está dirigido a estudiantes, docentes, directivos académicos y desarrolladores, y proporciona una descripción completa del comportamiento esperado del sistema.

1.2 Alcance

El sistema permitirá centralizar, almacenar, organizar y consultar los trabajos de grado desarrollados por los estudiantes de la Facultad de Ingeniería de Sistemas. Proporcionará mecanismos de autenticación, control de acceso por roles, búsqueda avanzada, gestión de anexos y administración de catálogos académicos.

El repositorio estará disponible como plataforma web y permitirá preservar no solo documentos finales, sino también productos asociados como software, bases de datos y material multimedia.

1.3 Personal involucrado

Nombre	Juan Andrés Cabrera Suárez
Rol	Analista, diseñador y desarrollador
Profesión	Ingeniero de sistemas
Responsabilidad	Análisis, diseño, desarrollo y documentación del sistema.
Información de contacto	juan.cabrera@usantoto.edu.co

Nombre	Cristian David Pineda Molano
Rol	Analista, diseñador y desarrollador
Profesión	Ingeniero de sistemas

Responsabilidad	Análisis, diseño, desarrollo y documentación del sistema.
Información de contacto	cristian.pinedam@usantoto.edu.co

1.4 Definiciones, acrónimos y abreviaturas

Nombre	Descripción
SRS	Software Requirements Specification
API	Application Programming Interface
JWT	JSON Web Token
CRUD	Create, Read, Update, Delete
Usuario	Persona registrada en el sistema
Administrador	Usuario con privilegios de gestión total
Anexo	Archivo asociado a un trabajo de grado
Trabajo de grado	Producto académico final del estudiante

1.5 Referencias

- IEEE Std 830-1998 / ISO/IEC/IEEE 29148:2018
- Pressman & Maxim, Software Engineering (2015+)
- Sommerville, Software Engineering (2016)

2. Descripción general

2.1 Perspectiva del producto

El repositorio digital es un sistema web que opera bajo una arquitectura cliente–servidor organizada por capas. Se integra con una base de datos relacional y un servicio externo de almacenamiento en la nube para la gestión de anexos.

El sistema reemplaza procesos manuales y dispersos por un entorno centralizado, seguro y estructurado.

2.2 Funciones del producto

- Autenticación y gestión de usuarios
- Control de acceso basado en roles
- Registro y gestión de trabajos de grado
- Administración de catálogos académicos
- Carga y consulta de anexos
- Búsqueda avanzada con filtros
- Visualización de estadísticas
- Preservación de información académica

2.3 Características de los usuarios

Tipo de usuario	Descripción
Administrador	Gestiona usuarios, catálogos, trabajos y estadísticas
Usuario	Registra, edita y consulta trabajos de grado

2.4 Restricciones

- Acceso mediante navegador web
- Uso de base de datos relacional
- Control de seguridad obligatorio
- Almacenamiento externo para anexos
- Cumplimiento de políticas institucionales

2.5 Suposiciones y dependencias

- Conectividad a Internet
- Disponibilidad del servicio en la nube
- Acceso institucional a la plataforma
- Crecimiento progresivo del repositorio

3. Requerimientos funcionales

Identificación de requerimientos:	RF – 01
Nombre del requerimiento:	Registro y autenticación de usuarios
Características:	Los usuarios deberán registrarse e identificarse para acceder al sistema.
Descripción del requerimiento:	El sistema deberá permitir el registro de nuevos usuarios y la autenticación mediante credenciales, garantizando el acceso seguro a las funcionalidades según el rol asignado.
Requerimiento NO funcional:	• RNF-01 Usabilidad • RNF-02 Seguridad • RNF-04 Disponibilidad
Prioridad del requerimiento: Alta	

Identificación de requerimientos:	RF – 02
Nombre del requerimiento:	Gestión de roles de Usuario
Características:	El sistema deberá manejar diferentes roles de acceso: administrador y usuario.
Descripción del requerimiento:	El sistema permitirá asignar, modificar y controlar los roles de los usuarios, definiendo las funcionalidades disponibles para cada perfil.
Requerimiento NO funcional:	• RNF-02 Seguridad • RNF-06 Mantenibilidad
Prioridad del requerimiento: Alta	

Identificación de requerimientos:	RF – 03
Nombre del requerimiento:	Gestión de trabajos de grado
Características:	El sistema deberá permitir registrar, consultar, actualizar y eliminar trabajos de grado.
Descripción del requerimiento:	El sistema permitirá la administración completa de los trabajos de grado, incluyendo su información general, autores, clasificación académica y observaciones.
Requerimiento NO funcional:	• RNF-01 Usabilidad • RNF-03 Rendimiento • RNF-06 Mantenibilidad • RNF-08 Preservación
Prioridad del requerimiento: Alta	

Identificación de requerimientos:	RF – 04
Nombre del requerimiento:	Gestión de directores y codirectores
Características:	El sistema permitirá administrar la información de los responsables académicos.
Descripción del requerimiento:	El sistema permitirá registrar, consultar, actualizar y eliminar directores y codirectores asociados a los trabajos de grado.
Requerimiento NO funcional:	• RNF-01 Usabilidad • RNF-06 Mantenibilidad
Prioridad del requerimiento: Media	

Identificación de requerimientos:	RF – 05
Nombre del requerimiento:	Administración de catálogos académicos
Características:	El sistema permitirá gestionar líneas de investigación, niveles de formación, períodos académicos y opciones de grado.
Descripción del requerimiento:	El sistema permitirá la administración de los catálogos académicos necesarios para la clasificación y organización de los trabajos de grado.
Requerimiento NO funcional:	• RNF-06 Mantenibilidad • RNF-08 Preservación
Prioridad del requerimiento: Alta	

Identificación de requerimientos:	RF – 06
Nombre del requerimiento:	Gestión de anexos de trabajos de grado
Características:	El sistema permitirá cargar, consultar y eliminar anexos asociados a trabajos de grado.

Descripción del requerimiento:	El sistema permitirá la carga, almacenamiento, consulta y eliminación de archivos digitales vinculados a los trabajos de grado.
Requerimiento NO funcional:	• RNF-02 Seguridad • RNF-03 Rendimiento • RNF-08 Preservación
Prioridad del requerimiento: Alta	

Identificación de requerimientos:	RF – 07
Nombre del requerimiento:	Búsqueda avanzada de trabajos de grado
Características:	El sistema permitirá realizar búsquedas mediante múltiples filtros combinables.
Descripción del requerimiento:	El sistema permitirá consultar trabajos de grado aplicando filtros como título, autor, director, línea de investigación, período académico y nivel de formación.
Requerimiento NO funcional:	• RNF-01 Usabilidad • RNF-03 Rendimiento
Prioridad del requerimiento: Alta	

Identificación de requerimientos:	RF – 08
Nombre del requerimiento:	Visualización de estadísticas del repositorio
Características:	El sistema permitirá visualizar estadísticas básicas de los trabajos de grado.
Descripción del requerimiento:	El sistema mostrará indicadores y gráficos que representen la producción académica almacenada en el repositorio.
Requerimiento NO funcional:	• RNF-01 Usabilidad • RNF-03 Rendimiento
Prioridad del requerimiento: Media	

Identificación de requerimientos:	RF – 09
Nombre del requerimiento:	Restricción de funcionalidades según rol
Características:	El sistema limitará el acceso a funcionalidades según el rol del usuario.
Descripción del requerimiento:	El sistema garantizará que los usuarios solo puedan acceder a las funcionalidades correspondientes a su perfil.
Requerimiento NO funcional:	• RNF-02 Seguridad
Prioridad del requerimiento: Alta	

4. Requerimientos no funcionales

Identificación de requerimientos:	RNF - 01
Nombre del requerimiento:	Usabilidad del sistema
Características:	El sistema deberá ofrecer una interfaz clara, consistente y de fácil navegación.
Descripción del requerimiento:	El sistema deberá permitir que los usuarios interactúen con el repositorio de forma intuitiva, sin necesidad de conocimientos técnicos avanzados, garantizando una experiencia de uso clara, accesible y coherente.
Prioridad del requerimiento: Alta	

Identificación de requerimientos:	RNF - 02
Nombre del requerimiento:	Seguridad de la información
Características:	El sistema deberá garantizar la confidencialidad, integridad y control de acceso.

Descripción del requerimiento:	El sistema deberá implementar mecanismos de seguridad que protejan la información, controlen el acceso a los recursos y prevengan usos no autorizados.
Prioridad del requerimiento: Alta	

Identificación de requerimientos:	RNF - 03
Nombre del requerimiento:	Rendimiento del sistema
Características:	El sistema deberá garantizar tiempos de respuesta adecuados.
Descripción del requerimiento:	El sistema deberá asegurar un desempeño eficiente durante las operaciones principales, especialmente en procesos de búsqueda, consulta y carga de información.
Prioridad del requerimiento: Alta	

Identificación de requerimientos:	RNF - 04
Nombre del requerimiento:	Disponibilidad del sistema
Características:	El sistema deberá estar disponible de forma continua.
Descripción del requerimiento:	El sistema deberá garantizar el acceso continuo para los usuarios autorizados, minimizando interrupciones y asegurando recuperación ante fallos.
Prioridad del requerimiento: Media	

Identificación de requerimientos:	RNF - 05
Nombre del requerimiento:	Escalabilidad del sistema
Características:	El sistema deberá soportar crecimiento progresivo de usuarios y datos.

Descripción del requerimiento:	El sistema deberá permitir el incremento del número de usuarios, trabajos de grado y anexos sin afectar significativamente el rendimiento.
Prioridad del requerimiento: Media	

Identificación de requerimientos:	RNF - 06
Nombre del requerimiento:	Mantenibilidad del sistema
Características:	El sistema deberá estar basado en una arquitectura modular y documentada.
Descripción del requerimiento:	El sistema deberá facilitar su mantenimiento, corrección y evolución mediante una estructura modular alineada con buenas prácticas de desarrollo.
Prioridad del requerimiento: Media	

Identificación de requerimientos:	RNF - 07
Nombre del requerimiento:	Portabilidad del sistema
Características:	El sistema deberá ser accesible desde distintos navegadores.
Descripción del requerimiento:	El sistema deberá funcionar correctamente en los principales navegadores web, sin requerir instalaciones adicionales ni software propietario.
Prioridad del requerimiento: Media	

Identificación de requerimientos:	RNF - 08
Nombre del requerimiento:	Preservación e integridad de la información
Características:	El sistema deberá garantizar integridad, respaldo y disponibilidad de los datos.

Descripción del requerimiento:	El sistema deberá asegurar la preservación digital de la información almacenada, protegiéndola contra pérdidas, fallos y alteraciones.
Prioridad del requerimiento: Alta	

5. Requerimientos de interfaz

5.1 Interfaz de usuario

- Formularios de autenticación
- Panel principal
- Módulos diferenciados por rol
- Interfaces de carga y consulta
- Panel administrativo

5.2 Interfaz de hardware

Acceso mediante dispositivos con navegador web.

5.3 Interfaz de software

- Base de datos relacional
- API REST
- Servicio externo de almacenamiento

6. Requerimientos de datos

- Almacenamiento estructurado
- Integridad referencial
- Trazabilidad de registros
- Persistencia de anexos
- Auditoría básica de cambios

7. Atributos de calidad

- Seguridad
- Confiabilidad

- Escalabilidad
- Usabilidad
- Disponibilidad
- Mantenibilidad

8. Reglas de negocio

- Un trabajo debe estar asociado a un usuario.
- Solo el creador o administrador puede editar.
- Solo el administrador puede eliminar.
- Todo trabajo debe tener clasificación académica.
- Los anexos deben estar vinculados a un trabajo.

9. Criterios de aceptación

- Autenticación funcional
- CRUD completo operativo
- Búsqueda avanzada efectiva
- Control de roles validado
- Anexos correctamente asociados
- Integridad de datos garantizada