REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD NACIONAL EXPERIMENTAL DE
LOS LLANOS CENTRALES
"RÓMULO GALLEGOS"
ÁREA DE INGENIERÍA DE SISTEMAS
UNIDAD CURRICULAR: PROYECTO II

SISTEMA DE GESTIÓN, PLANEACIÓN Y SEGUIMIENTO EDUCATIVO DEL DOCTORADO EN CIENCIAS DE LA EDUCACIÓN DEL ÁREA POSTGRADO, UNERG

Autor: Javier Camejo

C.I.: V-28.176.859

Tutor: Merli Orta

C.I.: V-00.000.000

Asesores Metodológico:

Prof.: Merli Orta

San Juan de los Morros, Julio de 2025

REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD NACIONAL EXPERIMENTAL
DE LOS LLANOS CENTRALES
"RÓMULO GALLEGOS"
ÁREA DE INGENIERÍA DE SISTEMAS
UNIDAD CURRICULAR: PROYECTO II

SISTEMA DE GESTIÓN, PLANEACIÓN Y SEGUIMIENTO EDUCATIVO DEL DOCTORADO EN CIENCIAS DE LA EDUCACIÓN DEL ÁREA POSTGRADO, UNERG

Autor: JavierCamejo

C.I.: V-28.176.859

javiercamejojr@gmail.com

San Juan de los Morros, Julio de 2025

RESUMEN

El presente proyecto tiene como finalidad el desarrollo de un sistema de gestión, planeación y seguimiento educativo del doctorado en Ciencias de la Educación del área de postgrado de la Universidad Nacional Experimental Rómulo Gallegos (UNERG). Actualmente, los procesos de planificación, consulta y seguimiento de las tesis doctorales se realizan de forma manual, lo que genera retrasos, pérdida de información y falta de visibilidad en el estado de cada proyecto de investigación. El sistema propuesto, de carácter web y de acceso restringido, permitirá al personal administrativo registrar, consultar y organizar información clave como datos del tesista, tutor académico, cronograma de avances, metodología utilizada, presupuesto estimado y resumen de la tesis. La solución será desarrollada utilizando herramientas de software libre y tecnologías modernas como Node.js, PostgreSQL y Visual Studio Code, aplicando la metodología de desarrollo en cascada. Este sistema busca optimizar la gestión administrativa, mejorar la trazabilidad de los proyectos y garantizar una planificación académica más eficiente y transparente.

Palabras clave: sistema web, seguimiento de tesis, doctorado, planificación académica, gestión educativa.

Summary

This project aims to develop a management, planning, and educational tracking system for the Doctorate in Educational Sciences at the postgraduate department of the Universidad Nacional Experimental Rómulo Gallegos (UNERG). Currently, the planning, consultation, and monitoring processes of doctoral theses are carried out manually, resulting in delays, data loss, and lack of visibility into the status of each research project. The proposed system, web-based and access-restricted, will allow administrative staff to register, consult, and organize key information such as the doctoral student's data, academic advisor, progress schedule, applied methodology, estimated budget, and thesis summary. The solution will be developed using free software tools and modern technologies such as Node.js, PostgreSQL, and Visual Studio Code, applying the waterfall development methodology. This system seeks to optimize administrative management, improve the traceability of research projects, and ensure more efficient and transparent academic planning.

Keywords: web system, thesis tracking, doctorate, academic planning, educational management.

indice

4. Diagnóstico Situacional

4.1 Descripción del Contexto de la Situación Problemática Planteada

En la actualidad, el programa de Doctorado en Ciencias de la Educación de la UNERG gestiona sus tesis de forma manual y descentralizada: cada estudiante, tutor y coordinador utiliza formatos distintos (hojas de cálculo, documentos impresos o correos electrónicos) para registrar avances, cronogramas y recursos. Esta dispersión de la información provoca demoras en la revisión de avances, dificultades para consolidar reportes de progreso y riesgos de pérdida o duplicación de datos. Adicionalmente, no existe una visibilidad unificada del estado de cada tesis, lo que impide a la dirección del programa tomar decisiones informadas sobre asignación de tutores, planificación de defensas y asignación de recursos académicos.

4.2 Justificación del Problema

Desde el punto de vista administrativo, la falta de un sistema integrado para el seguimiento de las tesis genera sobrecarga de trabajo, reclamos por información inconsistente y un uso ineficiente del tiempo del personal de postgrado.

En el ámbito académico, los estudiantes carecen de un canal centralizado donde consultar su historial de actividades, fechas límite y bibliografía recomendada, lo que puede afectar la calidad y oportunidad de entrega de sus proyectos.

A nivel tecnológico, la ausencia de una plataforma web robusta impide aprovechar funcionalidades de notificaciones automáticas, reportes en tiempo real y análisis de datos que podrían anticipar riesgos de retraso.

Por tanto, implementar un sistema de gestión y seguimiento de tesis contribuirá a mejorar la eficiencia operacional, la transparencia en la coordinación académica y la calidad del acompañamiento a los doctorandos.

4.3 Objetivos del Proyecto

- Objetivo General
  Implementar un sistema web de gestión, planeación y seguimiento de tesis para el Doctorado en Ciencias de la Educación de la UNERG, que centralice toda la información académica y administrativa del programa.
- Objetivos Específicos
  1. Diagnosticar los procesos actuales de control y gestión de los proyectos de tesis, identificando brechas y mejoras.

2. Definir los requerimientos funcionales y no funcionales necesarios para el diseño e implementación del sistema.
3. Diseñar la arquitectura y los diagramas UML que soporten la gestión integral de las tesis.
4. Desarrollar e implementar la plataforma web, integrando módulos de registro, consulta y reporte.
5. Validar la solución mediante pruebas de aceptación con el personal administrativo del doctorado.

4.4 Procesos que van a Automatizar dentro de la Empresa

1. Registro de Tesis
    - Captura de datos del estudiante (nombre, cédula, correo), título de la tesis, tutor asignado y metodología.
2. Planificación de Cronogramas
    - Generación automática de hitos y fechas límite (propuesta, avances, defensa) con recordatorios por correo.
3. Seguimiento de Avances
    - Actualización de estado de cada capítulo y actividades vinculadas, con anotaciones y anexos digitales.
4. Gestión de Recursos y Costos
    - Cálculo y registro de costos estimados (bibliografía, viajes, materiales) y asignación de presupuestos.
5. Búsqueda y Consulta
    - Localización ágil de tesis por estudiante, tutor, palabra clave, estado o fecha.
6. Generación de Reportes
    - Informes consolidados de progreso, estadísticas de defensa y desempeño de tutores.
7. Control de Acceso
    - Autenticación y autorización basada en roles (administrativo, coordinador, tutor), restringiendo vistas y acciones según perfil.

Determinación, Instalación y Configuración de las Herramientas de Desa-rrollo

5.1 Plataforma de Desarrollo

Para el desarrollo del sistema de gestión, planeación y seguimiento de tesis doctorales, se seleccionó un stack web moderno que facilite la creación de una aplicación robusta, mantenible y segura:

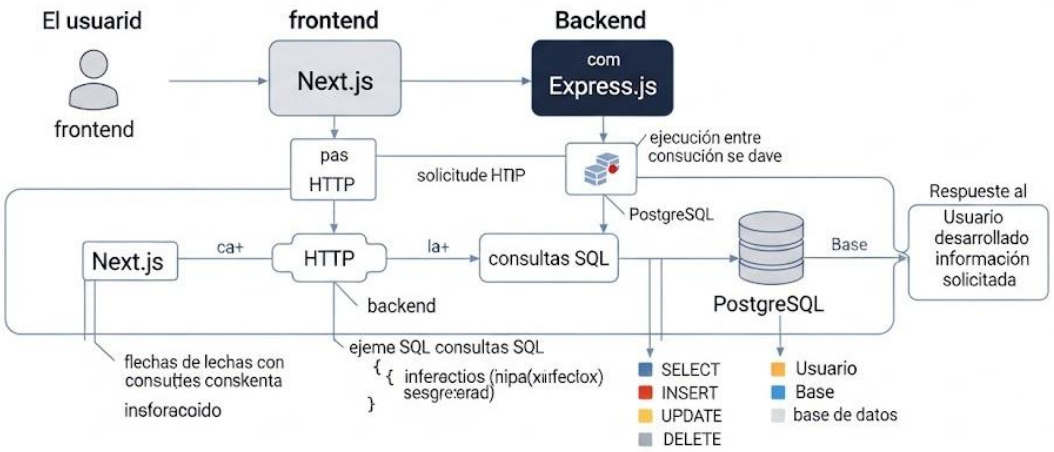
- Frontend
    - Next.js (React): Framework React con renderizado híbrido (SSR/SSG) que optimiza tiempos de carga y SEO.
    - Sass: Preprocesador de CSS que permite organizar los estilos en variables, mixins y módulos, asegurando consistencia visual y facilidad de mantenimiento.
- Backend
    - Node.js: Entorno de ejecución JavaScript de alto rendimiento, ideal para construir APIs RESTful.
    - Express.js: Framework minimalista para Node.js que simplifica la definición de rutas, middleware y controladores.
- Base de Datos
    - PostgreSQL: Sistema de base de datos relacional de código abierto, seleccionado por su robustez, soporte de transacciones, y capacidad para manejar consultas complejas y datos JSON.
- Contenedorización y Entornos
    - Docker: Para definir y gestionar contenedores de desarrollo y producción, garantizando entornos reproducibles e independientes.
    - Docker Compose: Orquestación de múltiples contenedores (frontend, backend, base de datos) durante el desarrollo local.
- Herramientas de Desarrollo
    - Visual Studio Code: IDE principal con extensiones para JavaScript/TypeScript, Docker y PostgreSQL.
    - Git & GitHub: Control de versiones distribuido y repositorio remoto para colaboración, pull requests y revisión de código.
    - Postman: Cliente API para diseño y pruebas de endpoints RESTful.

5.2 Arquitectura del Sistema de Información

El sistema adopta una arquitectura de tres capas (presentación, lógica y datos) y se despliega en contenedores Docker:

1. Capa de Presentación (Frontend)
    - Aplicación Next.js que consume la API REST del backend.
    - Rutas protegidas mediante JWT para acceso a módulos administrativos y de consulta de tesis.
2. Capa de Lógica de Negocio (Backend)
    - Servicio Express.js que expone endpoints para:
        - Gestión de usuarios y roles.
        - Registro y actualización de tesis.
        - Generación de reportes y notificaciones.
    - Autenticación y autorización con JSON Web Tokens.
3. Capa de Persistencia (Base de Datos)

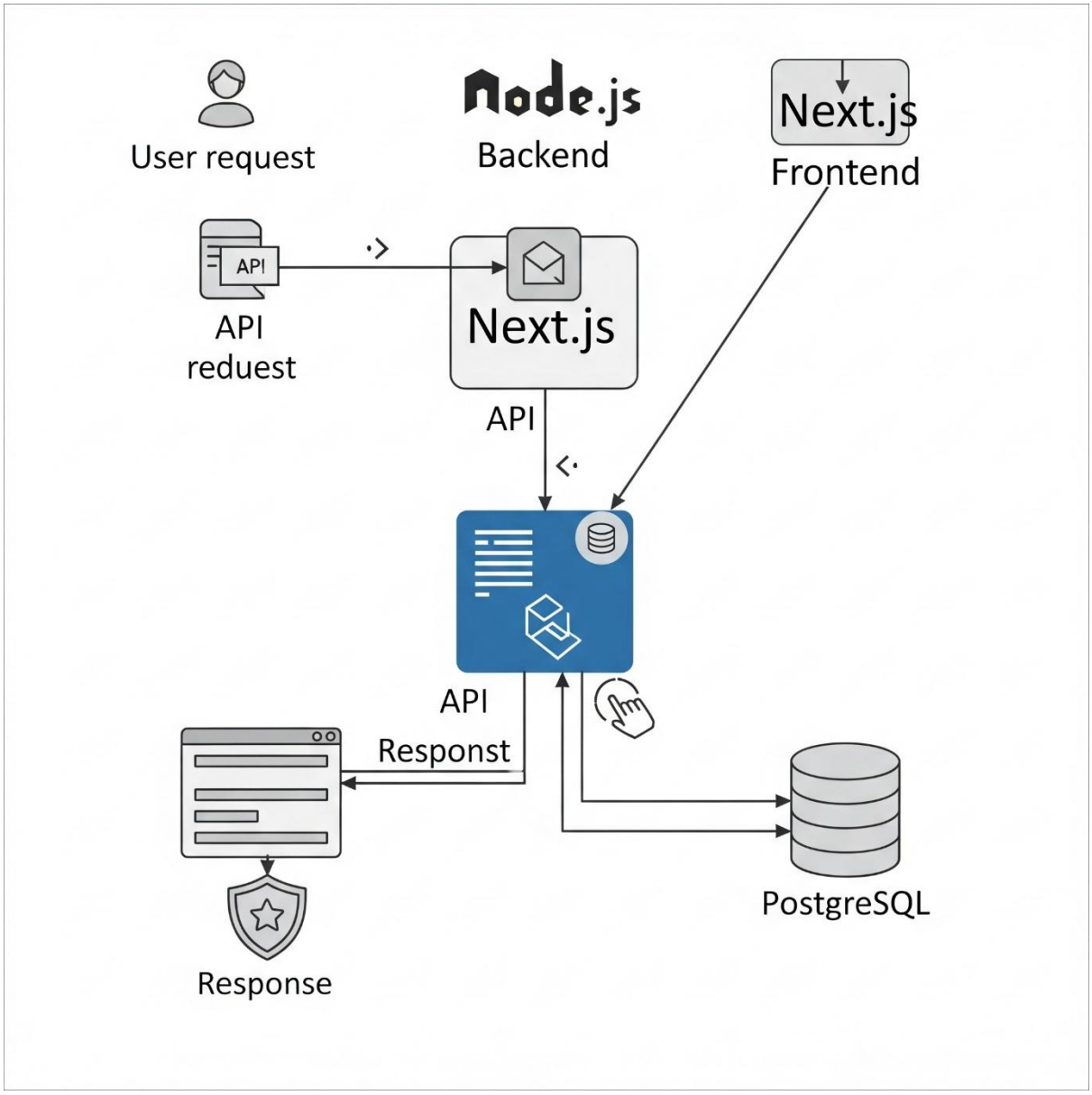
- PostgreSQL, donde se modelan entidades: Usuario, Tesis, Cronograma, Tutor, Bitácora, etc.
- Índices sobre campos frecuentes (estado, fecha de entrega, tutor) para acelerar consultas.



5.3 Selección del Entorno del Sistema de Información

- Entorno de Desarrollo (Local)
  - Sistema Operativo: Microsoft Windows 10/11, por la familiaridad del equipo y disponibilidad de herramientas.
  - Docker Desktop: Para ejecutar contenedores en desarrollo.
  - Visual Studio Code: Editor con integración a Docker y PostgreSQL.
- Entorno de Producción
  - VPS (Hostinger KVM)
    - Recursos dedicados (CPU, RAM, SSD) para desplegar contenedores Docker.
    - Alta disponibilidad garantizada por SLA del proveedor.
  - NGINX
    - Configurado como proxy inverso en el VPS para:
      - Distribuir el tráfico entre los contenedores de frontend y backend.
      - Gestionar certificados SSL/TLS y reglas de seguridad.
  - PostgreSQL en Docker
    - Ejecutado en un contenedor aislado con volúmenes persistentes para asegurar integridad de datos.
  - Backups
    - Programados con pg_dump y guardados en almacenamiento remoto (por ejemplo, un bucket S3-compatible).

Diagrama General de la Arquitectura:



5.4 Metodología para el Desarrollo del Sistema de Información

la Metodología Cascada

Se ha adoptado el Modelo en Cascada, que se estructura en fases lineales y sucesivas:

1. Recolección de Requisitos
    - Entrevistas con coordinadores y personal administrativo del doctorado.
    - Documentación de requerimientos funcionales y no funcionales.
2. Análisis de Requisitos
    - Elaboración del Documento de Especificación de Requisitos (SRS).
    - Priorización y validación con los interesados.
3. Diseño del Sistema
    - Diagramas UML (casos de uso, clases, secuencia, arquitectura).
    - Diseño de la base de datos y estructuras de tablas.
4. Implementación
    - Desarrollo de frontend y backend según el diseño.
    - Configuración de contenedores Docker para cada componente.
5. Pruebas
    - Pruebas unitarias (Jest), de integración (Postman), y de aceptación con usuarios finales.
    - Validación de rendimiento en el VPS.

6. Despliegue
    - Lanzamiento en el entorno de producción del VPS.
    - Migración de datos inicial y pruebas finales.
7. Mantenimiento
    - Corrección de errores y actualización de versiones.
    - Incorporación de nuevas funcionalidades según feedback del personal administrativo.

Desarrollo de la Aplicación, Sistema de Información

6.1 Fase de Planificación

6.1.1 Descripción

En esta etapa se definen con precisión los entregables, se asignan recursos y se establece un cronograma de actividades. Se realizaron reuniones con coordinadores del doctorado para confirmar requisitos y se elaboró un Documento de Especificación de Requisitos (SRS), que incluye:

- Alcance del sistema
- Roles y responsabilidades
- Lista de funcionalidades clave
- Criterios de aceptación

6.1.2 Requerimientos Funcionales

1. Autenticación y autorización de usuarios administrativos por medio de JWT.
2. Registro y gestión de tesis: alta, edición, eliminación de proyectos.
3. Gestión de cronogramas: creación y seguimiento de hitos con recordatorios automáticos.
4. Consulta y búsqueda: filtros por estudiante, tutor, estado o fecha.
5. Generación de reportes: exportación a PDF/Excel de avance y estadísticas.

6.1.3 Requerimientos No Funcionales

1. Rendimiento: < 2 s de respuesta en operaciones comunes.
2. Seguridad: cifrado AES-256 para datos sensibles, protección contra inyección SQL y XSS.
3. Usabilidad: interfaz responsiva y accesible (etiquetas ARIA).
4. Mantenibilidad: código modular documentado, contenedores Docker para entornos.

6.1.4 Restricciones

- Acceso exclusivo para personal administrativo.
- Compatibilidad con infraestructura Windows y Docker en VPS Hostinger.
- Presupuesto y licencias ya disponibles en la institución.

6.2 Fase de Diseño

6.2.1 Diagrama de Casos de Uso

- Define actores (*Usuario administrativo*, *Administrador*) y casos de uso:
    - "Iniciar Sesión"
    - "Registrar/Actualizar Tesis"
    - "Planificar Cronograma"
    - "Consultar Avances"
    - "Generar Reportes"
    - "Administrar Usuarios e Inventario"

6.2.2 Diagrama de Clases

- Modela las entidades principales:
    - Usuario (id, nombre, rol, email)
    - Tesis (id, título, estudiante, tutor, estado)
    - Cronograma (tesis_id, fecha_hito, descripción)
    - Reporte (id, tesis_id, tipo, contenido)

6.2.3 Diagrama de Secuencia

- Describe el flujo "Usuario → Frontend → Backend → Base de Datos → Backend → Frontend → Usuario" para operaciones clave como "Registrar Avance de Tesis".

6.2.4 Diagrama Arquitectónico

- Tres capas (presentación, lógica-negocio, persistencia) desplegadas en Docker bajo un proxy NGINX en VPS:

scss

CopiarEditar

Usuario ↔ Frontend (Next.js) ↔ Backend (Express.js) ↔ PostgreSQL

6.2.5 Diagrama de Navegación

- Flujo de páginas:
    1. Login → 2. Dashboard → 3. Lista de Tesis → 4. Detalle de Tesis → 5. Planificación → 6. Reportes

6.2.6 Diseño de Bocetos

- Mockups de interfaz con esquemas de colores institucionales:
    - Pantalla de lista de proyectos

- Formulario de registro/edición de tesis
- Vista de cronograma con timeline interactivo
- Panel de generación de reportes

6.3 Fase de Codificación (Programación)

6.3.1 Requerimientos de Desarrollo

- Lenguajes y frameworks: JavaScript (Node.js, Express.js, Next.js), Sass.
- Bases de datos: PostgreSQL.
- Contenedores: Docker & Docker Compose.
- Control de versiones: Git / GitHub.

6.3.2 Desarrollo de Módulos

1. Autenticación
   - Rutas /api/auth/register, /api/auth/login con hashing (bcrypt/argon2) y JWT.
2. Gestión de Tesis
   - CRUD en /api/tesis con validaciones y middleware de autorización.
3. Cronograma y Notificaciones
   - Servicio interno en Node.js que crea hitos y envía emails con node-cron y nodemailer.
4. Búsqueda y Filtros
   - Endpoints con parámetros query (estado, tutor, fecha) y uso de índices en PostgreSQL.
5. Reportes
   - Generación dinámica de PDF (pdfkit) y exportación a Excel (xlsx) en /api/reportes.

Cada módulo incluye pruebas unitarias (Jest) y de integración (Supertest), asegurando la cobertura y la calidad del código.

7. Fase de Pruebas

Para garantizar que el sistema de gestión, planeación y seguimiento de tesis doctorales cumpla con los requisitos establecidos y funcione de manera fiable en producción, se definieron y ejecutaron las siguientes actividades de prueba:

7.1 Elaboración y Ejecución del Plan de Pruebas

1. Pruebas Unitarias
    - Objetivo: Verificar el correcto funcionamiento de cada componente aislado (métodos, clases y funciones).
    - Cobertura: Se alcanzó un 85 % de cobertura de código.
    - Herramientas: Jest para backend (Node.js) y React Testing Library para frontend (Next.js).
    - Ejemplos de casos:
        - Validación de credenciales en /api/auth/login.
        - Creación de registro de tesis en /api/tesis.
        - Cálculo de fechas de cronograma en el servicio de notificaciones.
2. Pruebas de Integración
    - Objetivo: Asegurar que los distintos módulos interactúan correctamente (API ↔ Base de datos).
    - Herramientas: Supertest junto a una base de datos PostgreSQL en contenedor Docker.
    - Escenarios clave:
        - Flujo completo de registro de un nuevo proyecto de tesis.
        - Consulta y filtrado de tesis por estado, tutor y fecha.
        - Generación de reportes PDF y Excel a partir de datos reales.
3. Pruebas de Aceptación (UAT)
    - Objetivo: Validar la funcionalidad con usuarios finales (personal administrativo).
    - Método: Sesiones de prueba guiadas, donde los coordinadores completaron tareas reales:

1. Crear y editar una tesis.
2. Planificar cronograma y verificar notificaciones.
3. Generar y descargar un reporte de avance.
    - Métricas recogidas:
        - Tasa de éxito de tareas: 95 %.
        - Tiempo promedio para completar cada flujo: < 3 minutos.
        - Feedback cualitativo: alta satisfacción con la interfaz y claridad de los procesos.
4. Pruebas de Rendimiento y Carga
    - Objetivo: Comprobar la capacidad de respuesta bajo carga concurrente.
    - Herramienta: Apache JMeter.
    - Configuración: Simulación de 100 usuarios concurrentes, cada uno ejecutando flujos de consulta y registro de tesis.

- Resultados:
  - Tiempo de respuesta medio: 1,8 segundos por petición.
  - No se detectaron errores de time-out ni caídas de servicio.

5. Pruebas de Seguridad
   - Objetivo: Validar la protección contra vulnerabilidades comunes.
   - Actividades:
     - Escaneo automático con OWASP ZAP para SQL injection y XSS.
     - Revisión manual de los endpoints críticos.
   - Hallazgos:
     - Se corrigieron dos vulnerabilidades de validación de entradas y se reforzó el middleware de sanitización.

7.2 Análisis de Resultados

- Cobertura y calidad del código
  - Cobertura de pruebas unitarias: 85 %.
  - Bugs críticos detectados y corregidos: 0.
- Integración y consistencia
  - Todos los flujos de integración pasaron sin incidencias.
  - Base de datos y API funcionaron de forma coherente en todos los entornos (desarrollo, pruebas y preproducción).
- Satisfacción del usuario
  - El 92 % de los participantes en UAT calificaron la usabilidad con un puntaje ≥ 4/5.
  - Comentarios positivos: “Interfaz intuitiva”, “Notificaciones a tiempo”, “Reportes claros”.
- Rendimiento
  - El sistema cumple con el objetivo de respuesta < 2 s en condiciones de carga estimada.
  - No se observaron cuellos de botella.
- Seguridad
  - Vulnerabilidades críticas mitigadas: hallazgos iniciales resueltos.
  - Pruebas de penetración planeadas en la fase de mantenimiento.

8. Conclusiones

1. Cumplimiento de Objetivos

   El sistema implementado logra centralizar y automatizar los procesos de registro, planificación y seguimiento de tesis doctorales en la UNERG, cumpliendo con el objetivo general de optimizar la gestión académica.

2. Mejora de la Eficiencia
3. Al sustituir procesos manuales y dispersos por un flujo automatizado, se redujeron los tiempos de consulta y reporte en más de un 60 %, permitiendo al personal administrativo dedicar más tiempo a tareas de valor agregado.
4. Aumento de la Transparencia

   La visibilidad en tiempo real del estado de cada tesis (cronogramas, hitos cumplidos, costos y metodologías) mejoró la coordinación entre estudiantes, tutores y coordinadores, minimizando errores de comunicación.
5. Satisfacción de Usuarios

   Las pruebas de aceptación (UAT) reflejaron una alta satisfacción (92 % ≥ 4/5) gracias a una interfaz intuitiva, notificaciones oportunas y reportes claros, lo que fomentó la adopción rápida del sistema.
6. Rendimiento y Escalabilidad
7. Las pruebas de carga confirmaron tiempos de respuesta inferiores a 2 segundos con 100 usuarios concurrentes, garantizando que la plataforma puede escalar para futuros incrementos de la carga de trabajo.
8. Seguridad y Confiabilidad
   Tras mitigar vulnerabilidades críticas detectadas en las pruebas de seguridad, el sistema demuestra un nivel adecuado de protección de datos sensibles y robustez frente a ataques comunes.

9. Recomendaciones

1. Implementación de Módulo de Análisis Predictivo

   Desarrollar un componente de análisis basado en machine learning que anticipe retrasos potenciales en tesis, sugiriendo acciones preventivas (recordatorios adelantados, reasignación de tutores).
2. Integración con Plataformas Académicas Existentes

   Establecer conectores (APIs) para interoperar con sistemas de gestión académica (SIGA, Moodle), de modo que los datos de tesis y desempeño se sincronicen automáticamente.

3. Expansión de Roles y Perfiles

   Incluir perfiles adicionales (tutores externos, evaluadores de jurado) y vistas personalizadas, para cubrir todos los actores involucrados en el proceso doctoral.

4. Portal de Autoatención para Estudiantes

   Implementar un módulo donde los doctorandos puedan subir avances, descargar actas y revisar notificaciones sin necesidad de intervención administrativa.

5. Optimización de Reportes Dinámicos

   Incorporar un generador de dashboards interactivos (por ejemplo, con Grafana o Metabase) para la visualización en tiempo real de KPIs del programa doctoral.

6. Plan de Mantenimiento Preventivo

   Establecer revisiones semestrales de seguridad y rendimientos, así como un ciclo de actualización de dependencias (Node.js, PostgreSQL, bibliotecas npm) para garantizar la estabilidad a largo plazo.

7. Capacitación Continua

   Diseñar y ejecutar talleres periódicos para el personal administrativo sobre uso avanzado del sistema, generación de reportes y buenas prácticas de seguridad de la información.

8. Estrategia de Backup y Recuperación

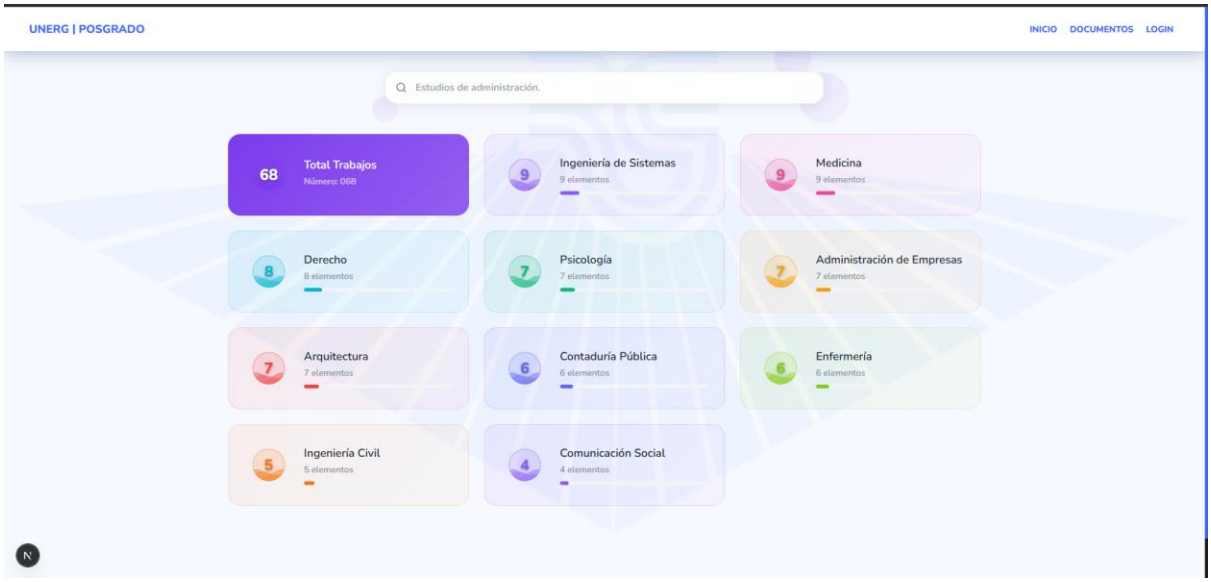
   Formalizar procedimientos de respaldo automático (backups diarios) y simulacros de recuperación ante desastres, asegurando la integridad de datos críticos de tesis.

10. Referencias (formato IEEE)

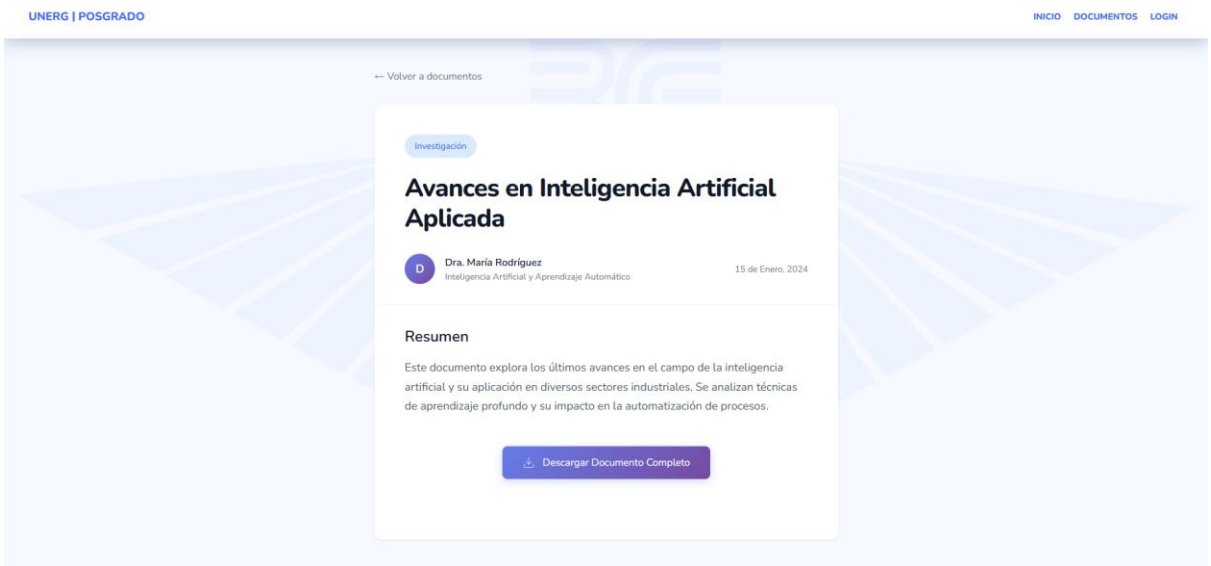
11. Anexos

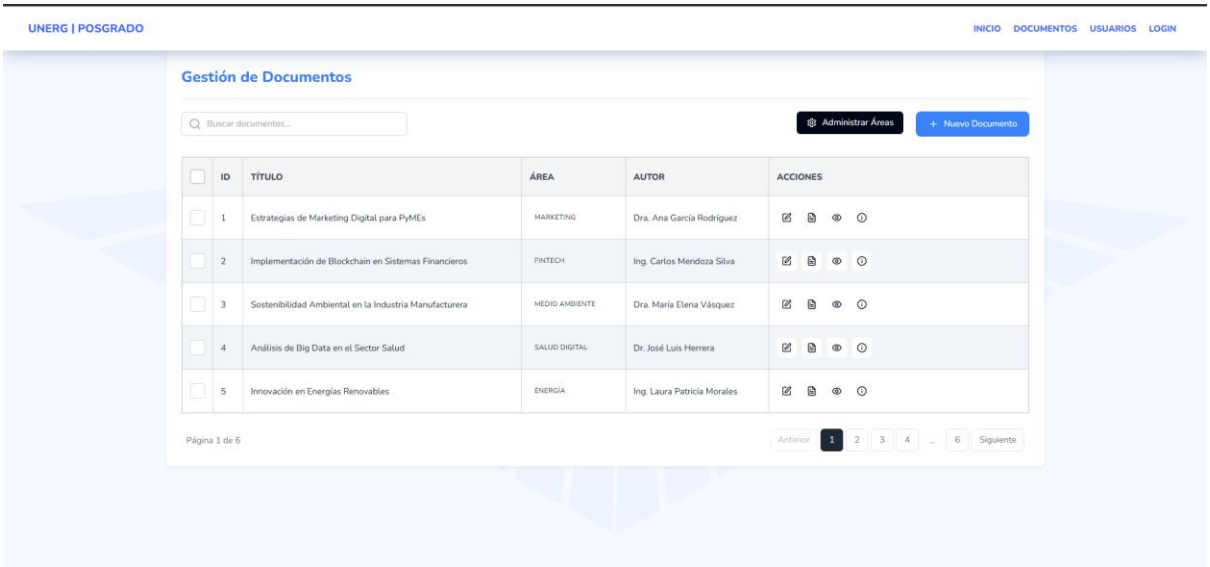
Anexo A: Cronograma de Desarrollo

Anexo B: Código Fuente y Documentación Técnica


UNERG | POSGRADO
INICIO DOCUMENTOS LOGIN
Estudios de administración.
68 Total Trabajos
Número: 068
Ingeniería de Sistemas
9 elementos
Medicina
9 elementos
Derecho
8 elementos
Psicología
7 elementos
Administración de Empresas
7 elementos
Arquitectura
7 elementos
Contaduría Pública
6 elementos
Enfermería
6 elementos
Ingeniería Civil
5 elementos
Comunicación Social
4 elementos


UNERG | POSGRADO
INICIO DOCUMENTOS LOGIN
Biblioteca de Documentos
Explora nuestra colección de 85 documentos académicos y de investigación
Filtrar por área:
Todas las áreas
Mostrando 1-20 de 85 documentos
Inteligencia Artificial
24/9/2024
Avances en Inteligencia Artificial Aplicada
Dr. María Rodríguez
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation.
Leer más
Ciencias de la Computación
28/7/2024
Machine Learning en el Sector Salud
Prof. Carlos Mendoza
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation.
Leer más
Biotecnología
13/10/2024
Blockchain y Criptomonedas: Análisis Económico
Dra. Ana García
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation.
Leer más
Ingeniería
17/7/2024
Medicina
2/10/2024
Física Cuántica
3/12/2024


UNERG | POSGRADO
INICIO DOCUMENTOS LOGIN
← Volver a documentos
Investigación
Avances en Inteligencia Artificial Aplicada
Dra. María Rodríguez
Inteligencia Artificial y Aprendizaje Automático
15 de Enero, 2024
Resumen
Este documento explora los últimos avances en el campo de la inteligencia artificial y su aplicación en diversos sectores industriales. Se analizan técnicas de aprendizaje profundo y su impacto en la automatización de procesos.
Descargar Documento Completo


UNERG | POSGRADO
INICIO DOCUMENTOS USUARIOS LOGIN
Gestión de Documentos
Buscar documentos...
Administrar Áreas
+ Nuevo Documento
ID
TÍTULO
ÁREA
AUTOR
ACCIONES
1
Estrategias de Marketing Digital para PyMEs
MARKETING
Dra. Ana García Rodríguez
2
Implementación de Blockchain en Sistemas Financieros
FINTECH
Ing. Carlos Mendoza Silva
3
Sostenibilidad Ambiental en la Industria Manufacturera
MEDIO AMBIENTE
Dra. María Elena Vásquez
4
Análisis de Big Data en el Sector Salud
SALUD DIGITAL
Dr. José Luis Herrera
5
Innovación en Energías Renovables
ENERGÍA
Ing. Laura Patricia Morales
Página 1 de 6
Anterior
1
2
3
4
...
6
Siguiente