



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
“ESCOM”



INGENIERÍA DE SOFTWARE

PROYECTO FINAL

Sistema de generación de grafos de conocimiento a partir de noticias

Informe técnico final 6CV3

García García Aram Jesua

Hernández Díaz Roberto Ángel

Hernández Jiménez Irmin

Toral Hernández Leonardo Javier

Trejo Flores Johann Daniel

Profesor: Gabriel Hurtado Avilés

Fecha de entrega: 11 de junio de 2025

Índice

Resumen Ejecutivo del Proyecto.....	1
Contexto del Problema	1
Solución Propuesta.....	1
Características Técnicas Principales	1
Arquitectura del Sistema	1
Funcionalidades Core	2
Beneficios del Sistema	2
Para los Usuarios.....	2
Para la Sociedad.....	3
Calidad y Métricas del Software	3
Resultados Obtenidos	3
Limitaciones Identificadas.....	4
Impacto Proyectado.....	4
Conclusión Estratégica	4
Decisiones finales de diseño y arquitectura implementada	5
Diagrama de Robustez	5
Modelo de Interfaz	6
Diagrama de clases de diseño.....	10
Patrones de diseño implementados.....	11
Singleton	11
Repository	12
Arquitectura del Sistema.....	13
Diagrama de Despliegue	14
Desafíos durante implementación, dockerización y desarrollo móvil	15

Lecciones aprendidas en este proyecto	16
Lecciones Técnicas Fundamentales.....	16
1. Complejidad de la Arquitectura de Microservicios	16
2. Optimización Prematura vs. Escalabilidad	17
Lecciones de Ingeniería de Software.....	17
3. Importancia de la Abstracción en NLP	17
Lecciones de Gestión de Proyecto	17
4. Iteración Rápida vs. Calidad	17
5. Importancia de la Diversidad en el Equipo	17
Lecciones de Investigación y Desarrollo.....	18
6. Colaboración Academia-Industria	18
Trabajo a futuro y mejoras.....	18
Expansión Lingüística	18
Soporte Multidioma Avanzado	18
Optimización y Performance.....	19
Arquitectura de Alto Rendimiento	19
Inteligencia Aumentada	19
Procesamiento Semántico Avanzado	19

Tabla de figuras

Figura 1. Diagrama de robustez del sistema.	5
Figura 2. Interfaces para el caso de uso "Sign Up & Login".	6
Figura 3. Interfaces para el caso de uso "Filtrar noticias y Visualización de grafo".	7
Figura 4. Interfaces para el caso de uso "Actividades de un Usuario Autenticado" (Parte 1).	8
Figura 5. Interfaces para el caso de uso "Actividades de un Usuario Autenticado" (Parte 2).	9
Figura 6. Interfaces para el caso de uso "Actividades de un Administrador".....	10
Figura 7. Diagrama de clases.....	11
Figura 8. Diagrama de Patrón de Diseño.	13
Figura 9. Diagrama de la arquitectura del sistema.	13
Figura 10. Diagrama de despliegue del sistema.	15

Resumen Ejecutivo del Proyecto

Sistema de generación de grafos de conocimiento a partir de Artículos Periodísticos

Contexto del Problema

Las nuevas generaciones enfrentan serias dificultades para mantenerse informadas a través de fuentes periodísticas tradicionales debido a:

- Preferencia por contenido visual y breve (videos, infografías).
- Abandono rápido de fuentes que no proporcionan información inmediata.
- Mayor exposición a desinformación por falta de consulta en fuentes confiables.
- Pérdida de atención hacia medios periodísticos de calidad.

Solución Propuesta

Desarrollo de una **aplicación web innovadora** que utiliza técnicas de Procesamiento de Lenguaje Natural (PLN) para transformar artículos periodísticos en grafos de conocimiento visuales e interactivos.

Características Técnicas Principales

Arquitectura del Sistema

- **Patrón de microservicios** con contenedores Docker.
- **Frontend:** React/TypeScript con interfaces diferenciadas.
- **Backend:** Spring Boot con API REST.

- **Bases de datos híbridas:** PostgreSQL (datos estructurados) + Neo4j (relaciones semánticas).

Funcionalidades Core

1. **Ingesta múltiple de contenido:** URLs, archivos PDF/DOC/TXT, y texto manual.
2. **Procesamiento NLP avanzado:** Identificación de entidades (personas, organizaciones, lugares) y sus relaciones usando Stanford CoreNLP.
3. **Generación automática de grafos:** Construcción incremental de grafos de conocimiento en Neo4j.
4. **Visualización interactiva:** Interfaz intuitiva con menos de 5 clics para acceder a funcionalidades.
5. **Sistema de autenticación:** Gestión de usuarios con roles diferenciados (administradores/usuarios regulares).

Beneficios del Sistema

Para los Usuarios

- **Comprensión rápida:** Visualización gráfica de entidades clave y sus relaciones.
- **Acceso inmediato:** Identificación de información relevante en segundos.
- **Experiencia mejorada:** Interfaz responsive y personalizable (temas claro/oscuro).
- **Organización temporal:** Clasificación automática de noticias por relevancia temporal.

Para la Sociedad

- **Combate la desinformación:** Fomenta el uso de fuentes periodísticas confiables.
- **Promueve pensamiento crítico:** Facilita el análisis de conexiones entre eventos y actores.
- **Democratiza el acceso:** Disponible como aplicación web accesible desde cualquier dispositivo.

Calidad y Métricas del Software

El sistema implementa estándares de calidad empresarial:

- **Fiabilidad:** Manejo robusto de excepciones y errores.
- **Extensibilidad:** Arquitectura basada en principios SOLID.
- **Portabilidad:** Contenerización completa con Docker.
- **Comprobabilidad:** Funciones atómicas con pruebas unitarias e integración.

Resultados Obtenidos

- **Sistema funcional completo** con todas las funcionalidades especificadas.
- **Interfaz intuitiva** validada con usuarios reales.
- **Procesamiento efectivo** de entidades y relaciones en textos periodísticos.
- **Visualización exitosa** de grafos de conocimiento complejos.
- **Arquitectura escalable** preparada para crecimiento futuro.

Limitaciones Identificadas

- **Idioma:** Actualmente limitado al inglés (Stanford CoreNLP).
- **Recursos:** Alto consumo computacional del sistema NLP.
- **Precisión:** Algunas complicaciones en el entendimiento de estructuras periodísticas complejas.

Impacto Proyectado

Este sistema representa una **solución innovadora** al problema crítico de la desinformación en la era digital, proporcionando una herramienta que:

- Moderniza la forma de consumir noticias.
- Mantiene la calidad del periodismo tradicional.
- Adapta el contenido a las preferencias de consumo actuales.
- Fomenta una cultura de información verificada y análisis crítico.

Conclusión Estratégica

El proyecto demuestra la viabilidad técnica y social de combinar tecnologías de PLN con visualización interactiva para crear una experiencia de consumo de noticias más eficiente, confiable y atractiva para las nuevas generaciones, contribuyendo significativamente a la lucha contra la desinformación.

Decisiones finales de diseño y arquitectura implementada

A manera de introducción de esta sección, se procede a describir que la decisión final de diseño y de la arquitectura implementada, fue exactamente la misma a la definida en la parte de diseño del proyecto, por lo que se puede determinar que no existió en ningún momento algún tipo de desvío tanto en la implementación del diseño como de la arquitectura.

Una vez planteado lo anterior, se prosigue a dar un breve resumen de lo que conforma la parte de diseño y arquitectura del sistema que describe con exactitud la estructura y funcionamiento del proyecto en su etapa final.

Diagrama de Robustez

En la siguiente Figura 1 se muestra el diagrama de robustez para este sistema. Este diagrama muestra la interacción entre los actores del sistema, las interfaces de comunicación, los controladores que brindan las funcionalidades al sistema y las entidades que el sistema utiliza.



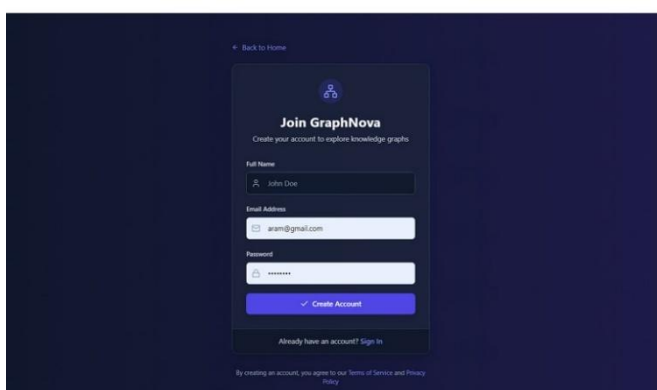
Figura 1. Diagrama de robustez del sistema.

Modelo de Interfaz

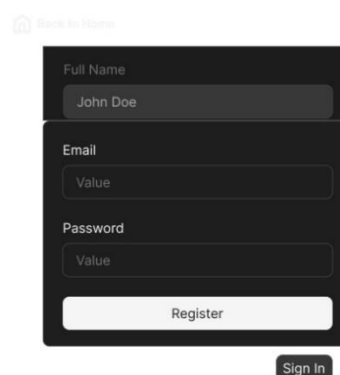
En las Figura 2, Figura 3, Figura 4, Figura 5 y Figura 6 se pueden ver todas las vistas que se encuentran involucradas en cumplir con las funciones que se solicitan por cada Caso de Uso existente en el sistema, que buscan cubrir con las necesidades solicitadas (requerimientos).

USE CASE (SIGN UP & LOGIN)

SIGN UP

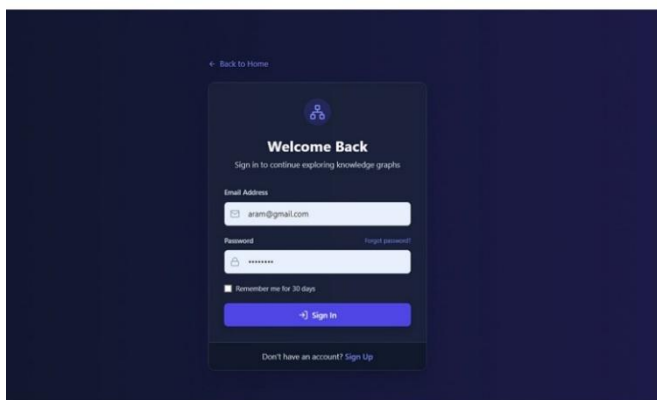


Mobile app sign-up screen for GraphNova. The screen has a dark blue background. At the top, there is a 'Back to Home' link. Below it is a 'Join GraphNova' header with a subtext 'Create your account to explore knowledge graphs'. The form includes fields for 'Full Name' (with a person icon), 'Email Address' (with an email icon), and 'Password' (with a lock icon). A 'Create Account' button is at the bottom. Below the button, there is a link 'Already have an account? Sign In'. At the very bottom, there is a small disclaimer: 'By creating an account, you agree to our Terms of Service and Privacy Policy'.

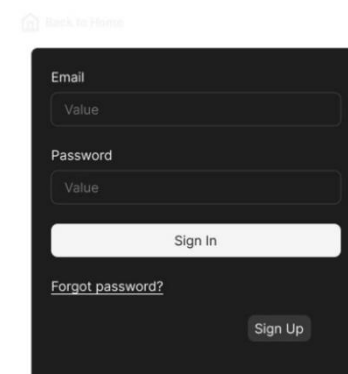


Web browser sign-up form for GraphNova. The form is on a white background. At the top, there is a 'Back to Home' link. The form includes fields for 'Full Name', 'Email', and 'Password'. A 'Register' button is at the bottom. Below the button, there is a 'Sign In' button.

LOGIN



Mobile app login screen for GraphNova. The screen has a dark blue background. At the top, there is a 'Back to Home' link. Below it is a 'Welcome Back' header with a subtext 'Sign in to continue exploring knowledge graphs'. The form includes fields for 'Email Address' (with an email icon) and 'Password' (with a lock icon). There is a 'Remember me for 30 days' checkbox. A 'Sign In' button is at the bottom. Below the button, there is a link 'Don't have an account? Sign Up'.

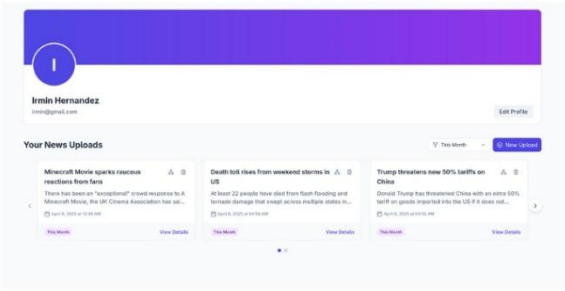


Web browser login form for GraphNova. The form is on a white background. At the top, there is a 'Back to Home' link. The form includes fields for 'Email' and 'Password'. A 'Sign In' button is at the bottom. Below the button, there is a 'Sign Up' button and a link 'Forgot password?'.

Figura 2. Interfaces para el caso de uso "Sign Up & Login".

USE CASE (KNOWLEDGE GRAPH & NEWS FILTERING)

NEWS FILTER



GRAPH VIEW

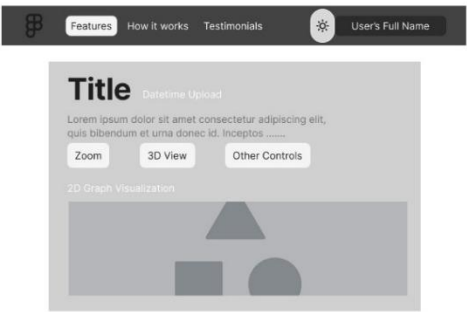
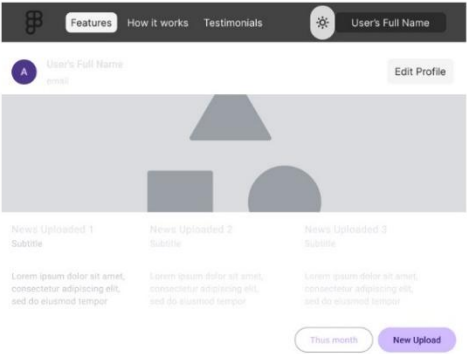
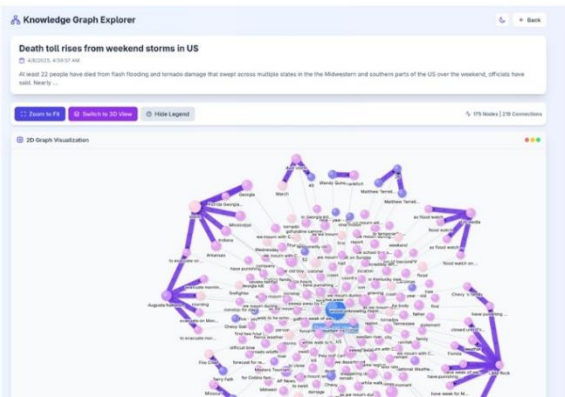
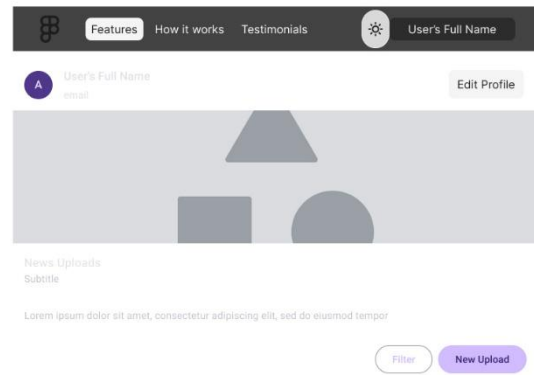
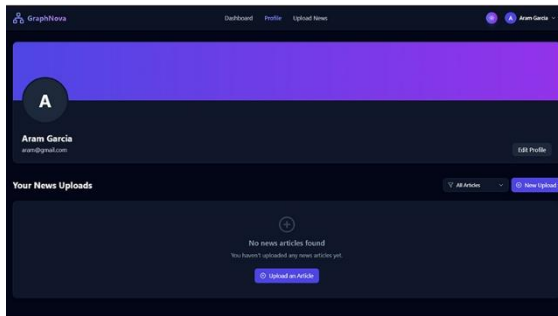


Figura 3. Interfaces para el caso de uso "Filtrar noticias y Visualización de grafo".

USE CASE (USER ACTIVITIES)

PROFILE



MODIFY PROFILE

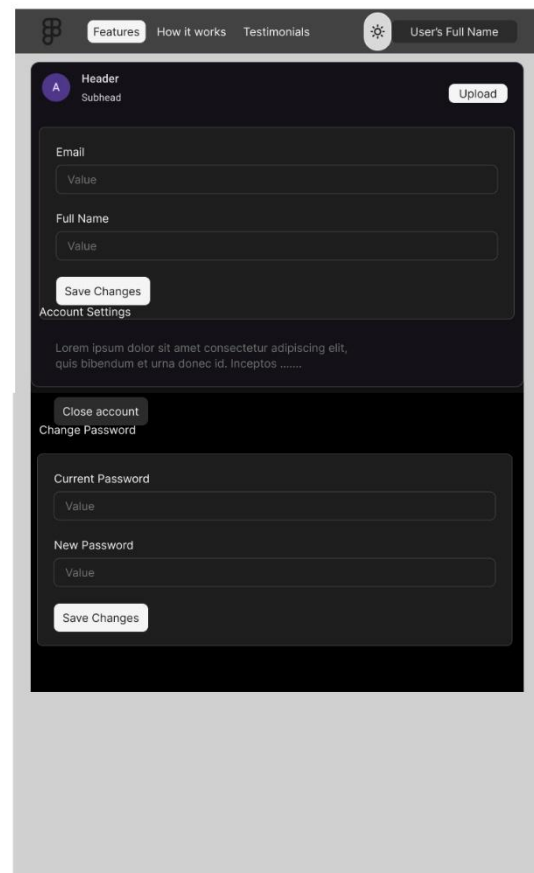
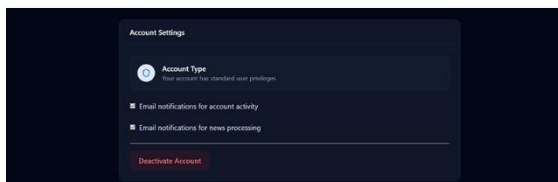
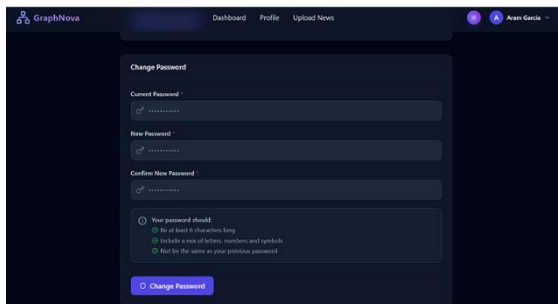
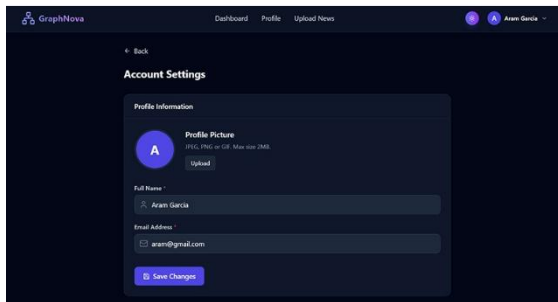


Figura 4. Interfaces para el caso de uso "Actividades de un Usuario Autenticado" (Parte 1).

UPLOAD NEWS (BY URL)

The screenshot shows the 'Upload News Article' form in the GraphNova application. The form is titled 'Upload News Article' and includes a sub-header 'Share news content for processing and knowledge graph generation'. Below this, there is a note: 'Our system will analyze your content, extract entities and relationships, and generate a knowledge graph. Processing may take a few minutes.' The form has three tabs: 'URL', 'Text', and 'File', with 'URL' currently selected. The 'URL' tab contains a 'Title' field with a placeholder 'Enter a title for this news article' and a 'News Article URL' field with a placeholder 'https://www.cnn.com/2019/07/01/...'. An 'Upload & Process' button is at the bottom.

[Back to Home](#)

The mockup shows the 'Upload News Article' form with a dark theme. It includes a 'Title' input field and a 'News Article URL' input field. An 'Upload & Process' button is located at the bottom of the form.

UPLOAD NEWS (BY TEXT)

The screenshot shows the 'Upload News Article' form in the GraphNova application. The form is titled 'Upload News Article' and includes a sub-header 'Share news content for processing and knowledge graph generation'. Below this, there is a note: 'Our system will analyze your content, extract entities and relationships, and generate a knowledge graph. Processing may take a few minutes.' The form has three tabs: 'URL', 'Text', and 'File', with 'Text' currently selected. The 'Text' tab contains a 'Title' field with a placeholder 'Enter a title for this news article' and a 'News Content' field with a placeholder 'Paste or type the full news article content here'. An 'Upload & Process' button is at the bottom.

[Back to Home](#)

The mockup shows the 'Upload News Article' form with a dark theme. It includes a 'Title' input field and a 'News Article Content' text area. An 'Upload & Process' button is located at the bottom of the form.

UPLOAD NEWS (BY FILE)

The screenshot shows the 'Upload News Article' form in the GraphNova application. The form is titled 'Upload News Article' and includes a sub-header 'Share news content for processing and knowledge graph generation'. Below this, there is a note: 'Our system will analyze your content, extract entities and relationships, and generate a knowledge graph. Processing may take a few minutes.' The form has three tabs: 'URL', 'Text', and 'File', with 'File' currently selected. The 'File' tab contains a 'Title' field with a placeholder 'Enter a title for this news article' and an 'Upload File' section with a placeholder 'Click to upload a file' and a file icon. An 'Upload & Process' button is at the bottom.

Upload & Process

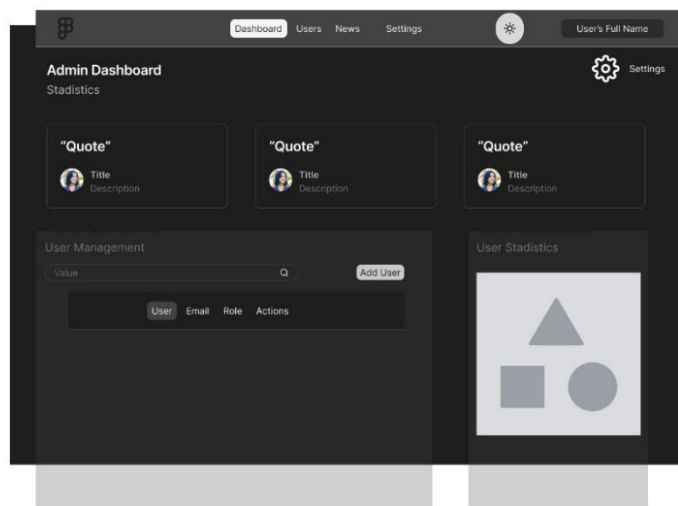
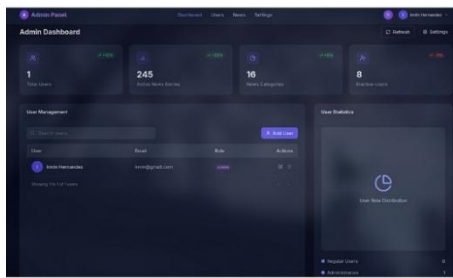
[Back to Home](#)

The mockup shows the 'Upload News Article' form with a dark theme. It includes a 'Title' input field and an 'Upload File' section with a file icon and a placeholder 'Value'. An 'Upload & Process' button is located at the bottom of the form.

Figura 5. Interfaces para el caso de uso "Actividades de un Usuario Autenticado" (Parte 2).

USE CASE (ADMIN ACTIVITIES)

USER MANAGEMENT



PROFILE

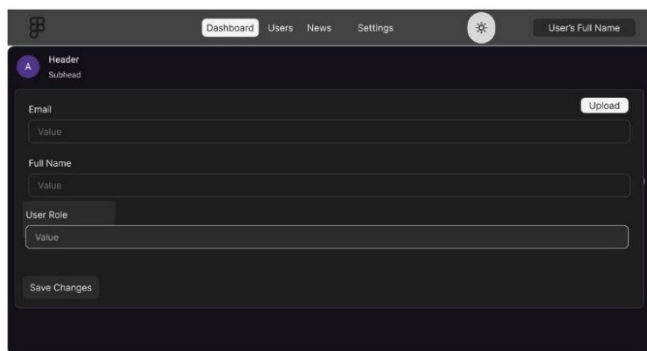
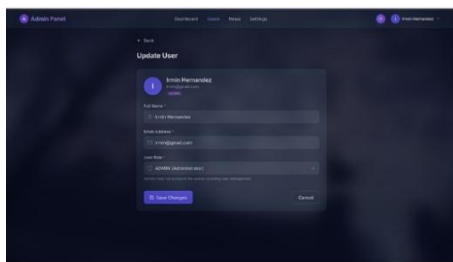


Figura 6. Interfaces para el caso de uso "Actividades de un Administrador".

Diagrama de clases de diseño

En la Figura 7 se muestran las clases utilizadas para la implementación del sistema, así como una descripción detallada de los constructores, métodos y atributos de cada clase.

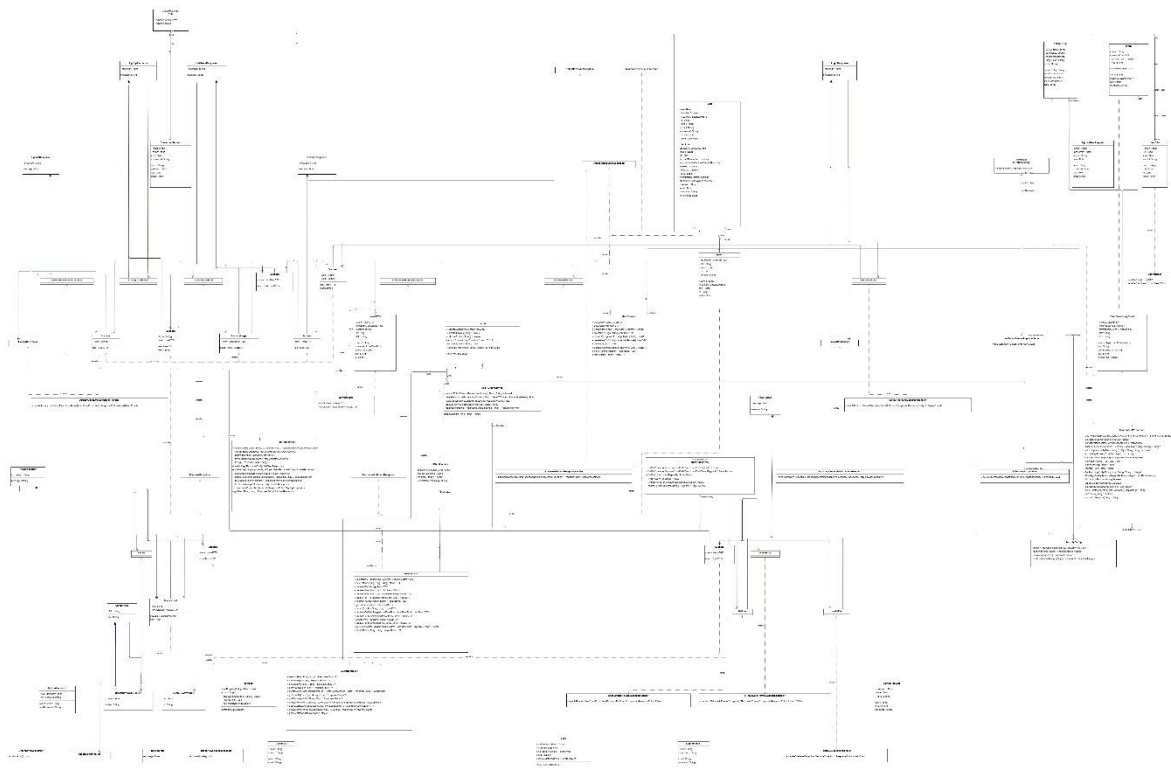


Figura 7. Diagrama de clases.

Patrones de diseño implementados

Para el desarrollo del sistema de generación de grafos de conocimiento a partir de noticias, se han implementado los siguientes patrones de diseño (los cuales se pueden ver en la Figura 8 como parte de un diagrama que los describe de una mejor manera):

Singleton

El patrón Singleton garantiza que una clase tenga una única instancia y proporciona un punto de acceso global a ella.

Aplicación en el sistema:

Se utiliza para instanciar clases que deben tener un solo punto de control, como la clase que gestiona la conexión a la base de datos de grafos (Neo4jService) y la clase que gestiona los tokens de seguridad (JwtUtil).

Beneficios:

- Previene la creación de múltiples instancias que podrían provocar inconsistencias en la gestión de datos o de seguridad.
- Mejora el control de acceso a recursos compartidos.
- Optimiza el uso de memoria y recursos del sistema.

Repository

El patrón Repository proporciona una capa intermedia entre la lógica de negocio y la capa de persistencia, permitiendo manejar las operaciones de acceso a datos de manera abstracta y desacoplada.

Aplicación en el sistema:

Se aplica en la interacción con las bases de datos (PostgreSQL y Neo4j), a través de repositorios como UserRepository y NewsRepository, facilitando las operaciones CRUD (Crear, Leer, Actualizar, Eliminar).

Beneficios:

- Aísla la lógica de acceso a datos, permitiendo cambios en la base de datos o en el framework de persistencia sin afectar la lógica del negocio.
- Facilita el mantenimiento y la evolución del sistema.
- Mejora la capacidad de realizar pruebas unitarias, ya que se pueden simular fácilmente las capas de persistencia.

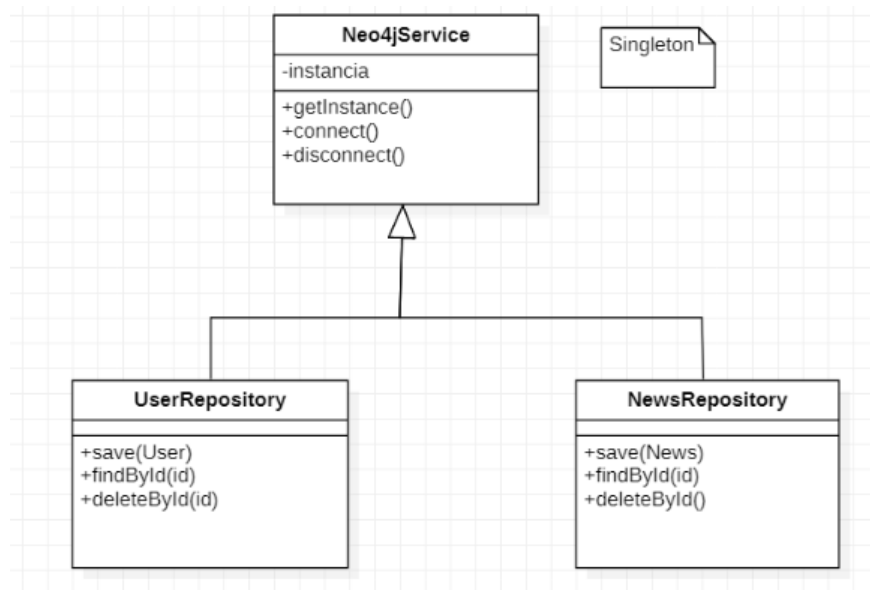


Figura 8. Diagrama de Patrón de Diseño.

Arquitectura del Sistema

El patrón arquitectónico que se decidió adaptar para el desarrollo del proyecto es un enfoque de cliente servidor para la versión actual del sistema, pero preparado para ser adaptado a un entorno de microservicios en el que se facilita la comunicación entre servicios a través de una API Rest estandarizada.

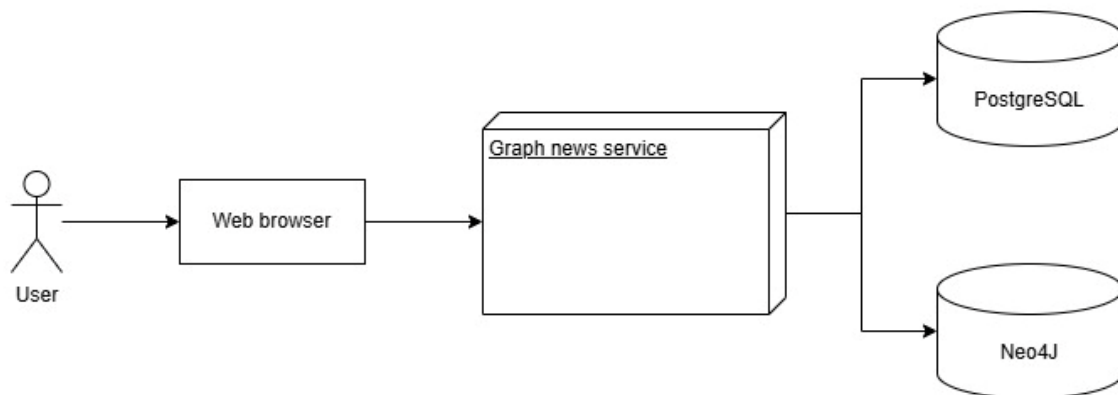


Figura 9. Diagrama de la arquitectura del sistema.

En la Figura 9 anterior se muestra un diagrama a alto nivel de la decisión de la arquitectura a utilizar, donde la aplicación web ejecutada en el navegador funciona

como el cliente, y la aplicación del servicio que se ejecuta de forma remota trabaja como el servidor, aunque este servidor en versiones futuras podría ser escalado a una arquitectura de microservicios gracias a la estructura adoptada por el sistema.

Diagrama de Despliegue

En la Figura 10 siguiente se muestra el diagrama de despliegue del sistema. En este diagrama se muestra como los componentes que componen al proyecto, siendo estos:

- **Base de datos relacional:** Instancia de Postgres que almacena datos estructurados correspondientes a los usuarios del sistema y a las noticias.
- **Base de datos no relacional (orientada a grafos):** Instancia de Neo4J que almacena las entidades y relaciones que se encuentran en una nota periodística.
- **Servicio web:** Es la aplicación java que maneja todas las funcionalidades del sistema por medio de una API Rest. Esta aplicación a su vez utiliza un navegador interno del contenedor para acceder a internet para técnicas de web scrapping.
- **Aplicación web:** Es la aplicación que se visualiza en el navegador, con todas las funcionalidades a las que un usuario puede acceder.

Este diagrama pretende ser una representación a bloques de los contenedores con los que cuenta el despliegue del sistema en su totalidad. Cada bloque es una instancia de un contenedor Docker, y su interacción entre contenedores se da mediante una interfaz virtual del contenedor, interfaz por la cual otros contenedores pueden comunicarse con esta aplicación mediante el puerto especificado.

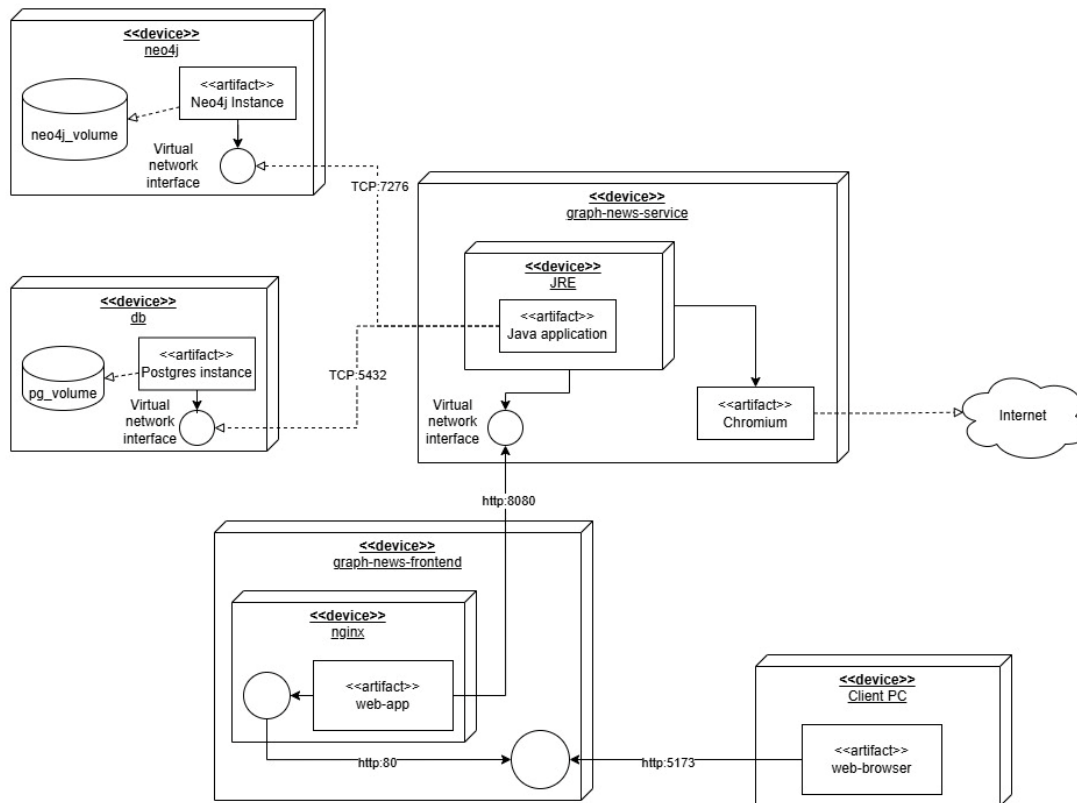


Figura 10. Diagrama de despliegue del sistema.

Desafíos durante implementación, dockerización y desarrollo móvil

El desafío que el equipo de desarrollo tuvo que enfrentar en el momento de implementación y desarrollo del aplicativo, tiene lugar en la creación del sistema en una aplicación web, pues aunque se utilizaron herramientas y lenguajes absolutamente compatibles entre sí que permitían facilidades para implementaciones en otros entornos operativos, existió una complicación.

Dicho problema o reto presentado fue la construcción de formularios en la aplicación móvil, pues como ya se mencionó, aunque el sistema estuvo desarrollado en una parte considerable en el framework React, y el framework que permite el desarrollo de aplicaciones móviles es ReactNative, que son compatibles entre sí dando

facilidades para la construcción de módulos y que mantengan la misma estructura visual y de manejo de datos, fue muy complicado dicho desarrollo.

Aun así, con suficiente tiempo para lograr una basta comprensión de ambas herramientas fue como se solucionó dicho reto, pues sólo era cuestión de entender las diferencias que existían entre sí de ambos frameworks descritos anteriormente.

También, es importante mencionar que durante la implementación del sistema existieron complicaciones en el tiempo que se estuvo trabajando con el PLN (Procesamiento del Lenguaje Natural), en donde en un principio al utilizarlo para incluirlo en el sistema, fue complicado pues le era difícil procesar notas periodísticas en primera instancia, posteriormente tuvo complicaciones al identificar conceptos muy simples.

Finalmente, cuando se logró entrenar con éxito, ahora cuando el módulo que se encargaba de activar el PLN era llamado, los dispositivos en los que se ejecutaba el proyecto presentaban un alto consumo de recursos, llegando a acercarse a un consumo de todos los recursos del 80%.

Para la solución de estos problemas radicó en la correcta implementación de esta herramienta y en la simplificación de su uso, para que en el momento de su activación, se procurara llegar al extremo mínimo del rango existente en la máxima cantidad de recursos que se pueden consumir al ejecutar un proceso en un dispositivo.

Lecciones aprendidas en este proyecto

Lecciones Técnicas Fundamentales

1. Complejidad de la Arquitectura de Microservicios

- **Aprendizaje:** La separación de responsabilidades facilita el mantenimiento pero incrementa la complejidad operacional

- **Aplicación:** Implementar observabilidad desde el día uno (logging distribuido, métricas, trazabilidad)

2. Optimización Prematura vs. Escalabilidad

- **Error inicial:** Sobredimensionamiento de la infraestructura NLP
- **Corrección:** Implementación de perfilado continuo y optimización basada en métricas reales

Lecciones de Ingeniería de Software

3. Importancia de la Abstracción en NLP

- **Desafío:** Acoplamiento fuerte con Stanford CoreNLP
- **Solución:** Creación de interfaces abstractas para procesadores de lenguaje
- **Beneficio:** Facilita el intercambio de bibliotecas NLP sin refactorización masiva

Lecciones de Gestión de Proyecto

4. Iteración Rápida vs. Calidad

- **Balance:** Implementar MVP funcional antes que sistema perfecto
- **Estrategia:** Technical debt consciente con plan de remediación
- **Resultado:** Feedback temprano permitió ajustes en los diseños arquitectónicos críticos.

5. Importancia de la Diversidad en el Equipo

- **Observación:** Diferentes perspectivas mejoraron el diseño UX
- **Implementación:** Inclusión de perfiles periodísticos en el proceso de diseño

- **Impacto:** Interfaz más intuitiva para el usuario final

Lecciones de Investigación y Desarrollo

6. Colaboración Academia-Industria

- **Beneficio:** Acceso a investigación de vanguardia y validación académica
- **Desafío:** Equilibrar rigor académico con necesidades comerciales
- **Resultado:** Publicación que valida el enfoque y atrae talento

Trabajo a futuro y mejoras

Expansión Lingüística

Soporte Multiidioma Avanzado

- **Implementación de SpaCy para español:** Migración gradual desde Stanford CoreNLP hacia modelos más eficientes.
- **Integración de modelos transformer multilingües:**
 - XLM-RoBERTa para procesamiento simultáneo de múltiples idiomas.
 - mBERT para tareas de clasificación y análisis semántico.
- **Fine-tuning especializado:** Entrenamiento con corpus específicos de noticias en español.
- **Validación cruzada:** Creación de datasets de evaluación para medir precisión por idioma
- **Detección automática de idioma:** Implementación de clasificadores para procesamiento automático sin intervención manual

Optimización y Performance

Arquitectura de Alto Rendimiento

- **Procesamiento distribuido:**
 - Implementación de Apache Kafka para cola de mensajes asíncrona
 - Microservicios especializados por tipo de procesamiento (NER, relaciones, grafos)
 - Auto-scaling con Kubernetes basado en carga de trabajo
- **Caching inteligente multinivel:**
 - Redis para resultados de NLP frecuentemente consultados
 - CDN para contenido estático y visualizaciones pre-renderizadas
 - Cache semántico para evitar reprocesamiento de contenido similar

Inteligencia Aumentada

Procesamiento Semántico Avanzado

- **Análisis de estructura periodística:**
 - Algoritmos de segmentación automática (titular, entradilla, desarrollo, conclusión)
 - Identificación de fuentes y citas mediante análisis sintáctico
 - Detección de sesgos y tonalidad emocional
- **Construcción de grafos temporales:**
 - Análisis de evolución de eventos a través del tiempo
 - Detección de patrones recurrentes en coberturas mediáticas
 - Predicción de tendencias basada en grafos históricos

- **Integración con fuentes externas:**
 - APIs de bases de datos factuales (DBpedia, Wikidata)
 - Verificación automática de hechos contra fuentes confiables
 - Enrichment de entidades con información contextual