Universidad de las Ciencias Informáticas

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Título: Módulo de Gestión de Configuración para el Sistema de Información de la Dirección de Gestión Tecnológica de la Universidad de las Ciencias Informáticas.

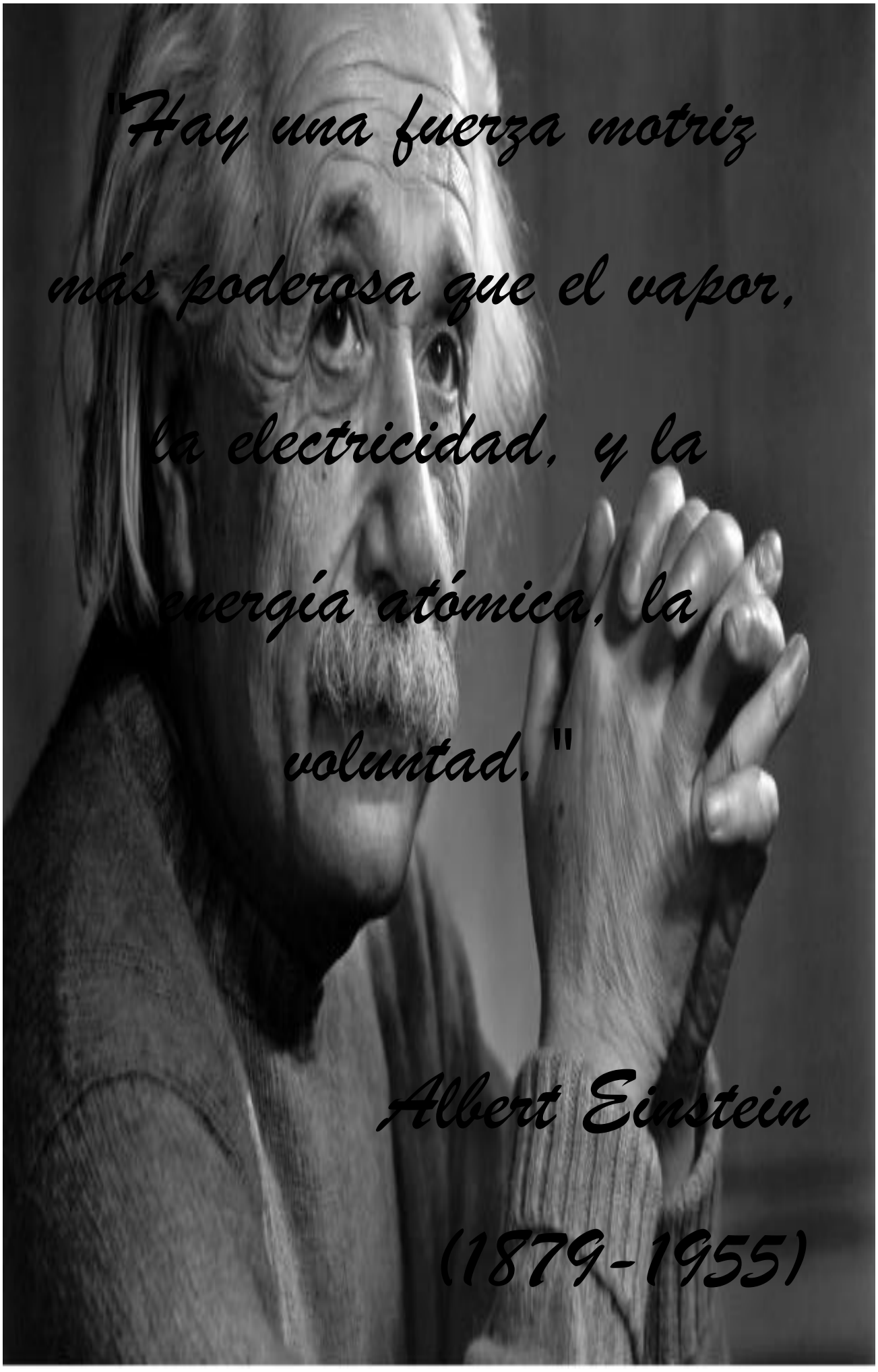
Autor: Aarom Ramsés Cárdenas Hernández

Tutores: Lic. Lester Rodríguez Vallejo

Lic. Angel Fabra Torres

La Habana, 8 de diciembre de 2023

“Año 64 de la Revolución”


"Hay una fuerza motriz
más poderosa que el vapor,
la electricidad, y la
energía atómica, la
voluntad."
Albert Einstein
(1879-1955)

DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título Módulo de Gestión de Configuración para el Sistema de Información de la Dirección de Gestión Tecnológica de la Universidad de las Ciencias Informáticas concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma la presente a los 8 días del mes de diciembre del año 2023.

Aarom Ramsés Cárdenas Hernández

_____

_____ Firma del Tutor

_____ Firma del Tutor

DATOS DE CONTACTO

Lester Rodríguez Vallejo (lesterr@uci.cu): Licenciado en Ciencias de la Computación en el año 2003. Profesor Asistente con 19 años de experiencia impartiendo docencia. Ha asumido diferentes roles en la docencia como: Jefe de Colectivo de asignatura a nivel de facultad, Jefe de Colectivo de Asignatura a nivel de Universidad, Jefe de la disciplina de Técnicas de Programación a nivel de Universidad, Jefe del Departamento Docente de Programación, Sistemas Digitales e Ingeniería de Software, Jefe del Departamento de Programación y Sistemas Digitales, Vicedecano de Formación. Actualmente es director de Gestión de Tecnología de la universidad. Ha estado vinculado a proyectos de desarrollo de software. Es miembro del grupo de investigación de Reconocimiento Lógico Combinatorio de Patrones de la universidad, tiene publicaciones científicas, ha participado en eventos, ha sido miembro de varios tribunales de defensa de tesis de diploma, tiene la Certificación en Administración de Base de Datos de la Microsoft.

Ángel Fabra Torres (afabra@uci.cu): Licenciado en Ciencias de la Computación en el año 2004. Profesor Asistente con 18 años de experiencia impartiendo docencia. Ha asumido diferentes roles en la docencia como: Jefe de Colectivo de asignatura a nivel de facultad, Jefe de Colectivo de Asignatura a nivel de universidad, Profesor Principal de Año, Jefe del Departamento Docente de Programación. Actualmente es Subdirector de Gestión de Tecnología de la universidad. Ha estado vinculado a proyectos de desarrollo de software como el desarrollo del Sistema de Gestión Hospitalaria XAVIA HIS, ocupando los roles de desarrollador y jefe de módulo. Ha participado en eventos y ha sido miembro de varios tribunales de defensa de tesis de diploma. Es graduado por la Academia CISCO en Cisco Certified Network Associate (CCNA)

AGRADECIMIENTOS

Después de ocho años, ocho años llenos de experiencias, risas y enseñanzas, ha llegado finalmente el día. Sería imperdonable no expresar mi gratitud a todos aquellos que han estado en mi camino durante este tiempo, quienes, de una forma u otra, hicieron posible este momento. Un agradecimiento especial a mi madre y a mi abuelo, quienes siempre creyeron en mí y sabían que, a pesar de mis desvíos, alcanzaría este sueño. Gracias a ellos soy quien soy hoy. También a mi abuela, que estoy seguro me observa desde el cielo con una sonrisa radiante y la mayor de las alegrías al verme lograr esto.

No puedo dejar de agradecer a mi esposa, mi compañera de vida y familia, que fue esencial para mi crecimiento personal y madurez. Sin ella, no estaría donde estoy hoy; ella ha transformado mi pasado en una mera anécdota. Su constante presencia, felicidad y apoyo han sido fundamentales en todo momento.

Un reconocimiento a todos mis amigos que me apoyaron en este viaje: Daniel, Gustavo, Mackay, mis hermanos de otras madres, quienes forman una parte significativa y son inolvidables recuerdos de mi paso por la universidad. A Jesús, compañero de trabajo, amigo y profesor, que me motivó a estudiar y es en gran parte responsable de mi éxito, pues su impulso y deseo de conocimiento enriquecieron mis habilidades como programador. A Enrique, que, a pesar de conocerlo por menos de un año, ha sido como mi hermano pequeño el cual tengo que enseñar y ayudar incentivándome a mejorar poco a poco.

Gracias también a Milena, Richard, José Carlos, Daniel, Kevin, Nathalí, Carlos, Roxana, Camila, Randy, y todos los que hicieron mi tiempo en la UCI más placentero. A mis profesores, que a pesar de los desafíos que les presenté, siempre confiaron en mi potencial. A mi tutor el cual además de ayudarme con todo este proceso de la tesis y estos meses de locura, en un momento de este trayecto como estudiante sus enseñanzas fueron cruciales gracias a esas clases de Programación 2 que ayudaron y despertaron ese deseo de estudio el cual aún atesoro. Todos ustedes ocupan un lugar especial en mi memoria y mi corazón.

DEDICATORIA

A mi madre y mis abuelos, quienes siempre confiaron en mí y me guiaron para que hoy sea la persona que soy. A mi esposa, que me ha brindado su amor incondicional, gracias a la cual estoy aquí cumpliendo este sueño.

RESUMEN

La adecuada gestión de la configuración es fundamental en cualquier sistema modular de gestión de información. La Dirección de Gestión Tecnológica (DGT) de la Universidad de las Ciencias Informáticas (UCI), como parte de su misión, requiere implementar un correcto control de la configuración de su Sistema de Información. Para abordar esta necesidad, se desarrolla un Módulo de Gestión de Configuración diseñado específicamente para la DGT. Este módulo permite el registro y seguimiento en tiempo real de usuarios, roles, permisos y otras configuraciones del sistema. Además, proporciona una interfaz web para que los administradores puedan verificar y gestionar las configuraciones de forma remota. La propuesta moderniza y optimiza el proceso de asignación dinámica de roles y permisos, mejorando la operatividad y reduciendo la probabilidad de errores. Se implementa con Python, Django y NextJs, aplicando la metodología AUP UCI. Se emplean patrones de diseño GOF y GRASP para robustecer el sistema. Mediante rigurosas pruebas se garantiza su correcto funcionamiento. En conclusión, este módulo de configuración incrementa la eficiencia en la DGT, apoyando su misión de modernización tecnológica en la UCI.

Palabras Clave: Configuración, Registro, Seguimiento, Sistema de Información, Interfaz Web

ABSTRACT

Proper configuration management is fundamental in any modular information management system. The Directorate of Technological Management (DGT) of the University of Computer Sciences (UCI), as part of its mission, requires the implementation of correct control of the configuration of its Information System. To address this need, a Configuration Management Module is developed, specifically designed for the DGT. This module allows the registration and real-time monitoring of users, roles, permissions, and other system configurations. Additionally, it provides a web interface for administrators to remotely check and manage configurations. The proposal modernizes and optimizes the process of dynamic role and permission allocation, improving operability and reducing the likelihood of errors. It is implemented with Python, Django, and NextJs, applying the UCI AUP methodology. GOF and GRASP design patterns are used to strengthen the system. Its correct functioning is guaranteed through rigorous testing. In conclusion, this configuration module increases efficiency in the DGT, supporting its mission of technological modernization at the UCI.

Keywords: Configuration, Registration, Monitoring, Information System, Web Interface

Índice

Índice de Tablas

Índice de Figuras

Introducción

Actualmente el mundo de la informática evoluciona aceleradamente, provocando de forma considerable impactos positivos en diferentes esferas de la sociedad como el gobierno, las telecomunicaciones, el deporte, la salud, entre otras, al proveer herramientas, tecnologías y metodologías que facilitan, agilizan y optimizan su gestión. Potenciando la automatización de procesos que antes se hacían manualmente en varias instituciones y logrando en un período relativamente corto una eficiencia casi exponencial.

Precisamente la utilización de las Tecnologías de la Información y las Comunicaciones (TIC) permiten que tareas complejas, laboriosas y propensas a errores humanos puedan ser realizadas de forma más eficiente y precisa por sistemas informáticos. El empleo de estas herramientas ha dado un giro radical a las formas y métodos clásicos de administrar una empresa, estas agregan valor a las actividades operacionales y de gestión empresarial en general y permite a las empresas obtener ventajas competitivas, permanecer en el mercado y centrarse en su negocio. (Riquelme, 2018)

En la implementación de un sistema informático intervienen muchos factores, uno de los principales es el factor humano. Por esta razón es necesario hacer una planeación estratégica tomando en cuenta las necesidades presentes y futuras de la empresa, así como una investigación preliminar y un estudio de factibilidad del proyecto que se desea. Dentro de las áreas beneficiadas por la digitalización se encuentra la de gestión de recursos tecnológicos, con la informatización de esta se ha logrado, un mejor seguimiento y control de sus recursos.

En Cuba existen centros de investigación que se dedican al desarrollo de aplicaciones informáticas para contribuir a la informatización del país. Una de estas instituciones es la Universidad de las Ciencias Informáticas (UCI), la cual tiene entre sus proyecciones, desarrollar capacidades productivas que generen productos y servicios informáticos de calidad, sostenibles y con alto valor agregado, según se describe en sus objetivos estratégicos.

La UCI tiene como misión contribuir a la transformación digital de la sociedad cubana, mediante la formación integral y continua de profesionales de las ciencias informáticas comprometidos con su Patria, así como la producción y comercialización de productos y servicios informáticos, aplicando ciencia e innovación, con proyección internacional y responsabilidad social para la sostenibilidad de la nación socialista. Para llevar a cabo esta misión la universidad cuenta con una estructura dividida en varias áreas, entre las

cuales se encuentran las que ofrecen servicios que garantizan el funcionamiento interno de dicho centro. Una de estas áreas es la Dirección de Gestión Tecnológica (DGT).

La DGT de la UCI tiene como misión mantener y modernizar la infraestructura tecnológica de la institución. Su objetivo es impulsar los procesos de formación, producción e investigación, proporcionando tecnología de punta de forma estable. Para ello realiza nuevas inversiones que permiten implementar servicios de excelencia, garantizando la fiabilidad, estabilidad, seguridad e invulnerabilidad de las tecnologías. De esta forma, se avanza hacia la soberanía tecnológica.

La DGT busca garantizar un escenario óptimo para que la universidad pueda alcanzar sus objetivos fundamentales en los procesos de enseñanza, investigación y extensión, se esfuerza por mantener y mejorar constantemente la infraestructura tecnológica de la universidad para impulsar su trabajo académico a la vanguardia. Para cumplir con esta misión la Dirección está organizada en tres grupos de trabajo especializados en Servicios Técnicos, un grupo de Gestión Económica y Compras, un Grupo de Aseguramiento y el Centro de Llamadas.

A su vez, ejecuta varios procesos amparados en procedimientos establecidos para gestionar eficientemente los recursos tecnológicos. Dentro de estos procedimientos destacan: PE-042-01-Gestión de incidencias del Grupo de Asistencia Técnica (GAT), PE-042-02 Taller, PE-042-03-Revisión de órdenes de servicio, PE-042-04-Local de Piezas Recuperadas, PE-042-05-Local de Diagnóstico, PE-042-06 Gestión de compras, PE-042-07-Autorizar y asignar partes y piezas, PE-042-09-Pago de facturas de compras y servicios, PE-042-10-Confección de órdenes de servicio y PE-042-16-Procedimiento de baja.

La implementación rigurosa de estos procedimientos exige una gestión óptima de los recursos tecnológicos disponibles y genera abundante documentación. Actualmente está en desarrollo un Sistema de Gestión de Información Tecnológica, concebido de manera modular, teniendo en cuenta las áreas en las que está estructurada la dirección, para informatizar los procesos y procedimientos que se ejecutan en cada una de ellas.

Este sistema garantiza la informatización de los procesos mencionados, donde el núcleo del sistema es la base para que exista un correcto funcionamiento en las operaciones que se realizan entre ellos, lo cual garantiza la integración entre los módulos y la gestión de elementos comunes como son: la seguridad y los nomencladores. El Sistema de Información en cuestión no cuenta actualmente con un módulo para este fin.

La falta de digitalización y automatización de los procesos y procedimientos implica:

- ✓ Existe un riesgo latente de pérdida de datos e información contenida en los documentos, debido al deterioro físico que sufren por el paso del tiempo y la exposición a condiciones ambientales adversas como la humedad. Esto puede ocasionar daños irreparables en los soportes, dificultando o impidiendo la recuperación de la información (los documentos que se manejan son de gran importancia para la dirección y deben estar guardados al menos por cinco (5) años).

- ✓ El procesamiento manual de los datos implica una alta probabilidad de errores humanos durante la manipulación y registro de la información (el llenado de la información la realizan diferentes personas en dependencia del documento). La introducción errónea de datos repercute negativamente en la calidad y utilidad de la información recabada.

- ✓ El área trabaja con estadísticas complejas, difíciles de adquirir que implican la revisión manual de documentación diversa y en cantidades elevadas. Esto conlleva a que en ocasiones los datos que se obtienen tengan errores y que se produzcan atrasos en la entrega de la información.

- ✓ Se hace muy complejo el control de los recursos tecnológicos por parte de dirección, ya que en términos de tecnología todas las áreas de la UCI están implicadas.

Teniendo en cuenta la situación problemática antes descrita, se identifica como **problema de la investigación:** ¿Cómo gestionar las configuraciones del Sistema de Gestión de Información de la Dirección de Gestión Tecnológica?

Objeto de estudio: Gestión de la configuración de Sistemas de Gestión de Información.

Objetivo general: Desarrollar el módulo de Gestión de la configuración para el Sistema de Gestión de Información de la DGT que permita gestionar las configuraciones a través de una interfaz web.

Campo de acción: Configuración del Sistema de Gestión de Información de la DGT.

Preguntas científicas:

1. ¿Cuáles son los fundamentos teóricos relacionados con el estado del arte de los procesos de Gestión de la configuración de Sistemas modulares de Gestión de Información?

2. ¿Qué características debe cumplir el proceso de Gestión de la configuración de Sistemas modulares de Gestión de Información para el apoyo en la toma de decisiones en la Dirección de Gestión Tecnológica de la Universidad de las Ciencias Informáticas para que cumpla con las necesidades de los clientes?
3. ¿Cuáles son las tecnologías, metodologías y herramientas que más se ajustan al desarrollo de la propuesta de solución?
4. ¿Cómo desarrollar el proceso de Gestión de la configuración de Sistemas modulares de Gestión de Información para el apoyo a la toma de decisiones en la DGT de la UCI de manera que contribuya con una perspectiva histórica de los datos?
5. ¿Cómo garantizar el correcto funcionamiento del proceso de Gestión de la configuración de Sistemas modulares de Gestión de Información para el apoyo a la toma de decisiones en la DGT de la UCI?

A continuación, se proponen las siguientes **tareas de investigación:**

Tareas de investigación:

- ✓ Estudio de los referentes teóricos y metodológicos de la Gestión de la configuración de Sistemas modulares de Gestión de Información y selección de la metodología y herramientas para la elaboración del marco teórico de la investigación para llevar a cabo el desarrollo del sistema informático.
- ✓ Análisis y diseño del módulo para la clasificación de las características y funcionalidades del sistema y obtener un mejor entendimiento de ésta.
- ✓ Diseño de una Base de Datos para gestionar la información.
- ✓ Implementación del sistema para gestionar la Gestión de la configuración de Sistemas modulares de Gestión de Información en la DGT de la UCI.
- ✓ Elaboración del diseño de casos de prueba partiendo del flujo de trabajo de cada uno de los requisitos identificados.
- ✓ Implementación y análisis de las pruebas para proporcionar información objetiva e independiente sobre la calidad del producto.

Los **métodos científicos** a utilizar en la investigación estuvieron determinados por el objetivo general y las tareas de investigación previstas.

A **nivel teórico** serán utilizados los métodos:

- ✓ **Histórico-Lógico:** se estudió la trayectoria histórica real del fenómeno y las tendencias del uso actual de sistemas de Gestión de la configuración para Sistemas de Gestión de Información con el objetivo de seleccionar las

características más apropiadas para darle cumplimiento al objetivo general de la investigación.

- ✓ **Analítico-Sintético:** se realizó el análisis de los documentos, las publicaciones, bibliografías y en general toda la información relacionada con los sistemas de Gestión de la configuración para Sistemas de Gestión de Información que hacen posible la elaboración de conclusiones tanto prácticas como teóricas y con esto reflejar los elementos más importantes y necesarios en el desarrollo del módulo.

- ✓ **Modelación:** Se utilizó para la representación de los procesos que enmarcan el objeto de estudio, del modelo lógico y físico de los artefactos correspondientes al ciclo de vida del software, ayudando a dar cumplimiento a las tareas de diseño de los procesos involucrados en la solución.

Los **métodos empíricos** ayudan a complementar el seguimiento a la evolución del proceso que se estudia, los utilizados en esta investigación son:

- ✓ **Observación:** La observación permitió adquirir un mejor entendimiento sobre cómo operan actualmente los sistemas y módulos utilizados para sistemas de Gestión de la configuración para Sistemas de Gestión de Información. Esto proporcionará información útil para el diseño de nuevos sistemas.

Para una mejor comprensión de la investigación, se definió una estructura capitular que aporta el grado de organización necesario y facilita la lectura del documento.

Capítulo 1. Fundamentación Teórica.

1.1. Introducción.

En el presente capítulo se analizaron los conceptos fundamentales relacionados al proceso de Gestión de la configuración de Sistemas modulares de Gestión de Información, con el objetivo de dar soporte teórico a la investigación. Además, se realiza un detallado estudio de módulos de gestión de la configuración a nivel internacional y nacional. De igual manera, se destacan las principales características de la metodología utilizada en la investigación, así como de las tecnologías y herramientas a emplear en la construcción de la propuesta de solución.

1.2 Descripción del Objeto de estudio

Dentro del vasto ecosistema de sistemas web, el módulo de configuración emerge como una piedra angular para la administración y adaptabilidad de dichos sistemas. Esencialmente, este módulo actúa como un tablero de control, proporcionando al administrador una serie de herramientas y funcionalidades que permiten personalizar y adaptar el sistema según las necesidades específicas del entorno en el que opera. Más que simples ajustes, un módulo de configuración adecuado es capaz de moldear el comportamiento y las características del sistema, garantizando su versatilidad frente a distintos requerimientos.

Esta adaptabilidad adquiere especial importancia en contextos donde los sistemas web se componen de múltiples módulos o componentes. En estos entornos modulares, la cohesión entre diferentes componentes es vital. Es aquí donde el módulo de configuración actúa como un puente, integrando y asegurando que todos los módulos trabajen armónicamente, evitando redundancias y posibles conflictos.

La seguridad, una preocupación omnipresente en el ámbito digital, es otra de las facetas en las que el módulo de configuración demuestra su valía. Desde este módulo, es posible establecer protocolos de seguridad, controlar niveles de acceso, definir roles y responsabilidades y, en general, instaurar barreras que protejan la integridad y privacidad de la información gestionada.

En el contexto de la UCI, la infraestructura tecnológica existente respalda plenamente los proyectos de desarrollo de software previstos. La universidad está equipada con computadoras en red, unidades de respaldo de energía (UPS), y una variedad de dispositivos que van desde laptops hasta dispositivos móviles. Cada uno de estos elementos, necesitan atención técnica especializada para su adquisición, instalación, configuración y mantenimiento.

Esta importante labor recae en la DGT, que está subordinada a la Vicerrectoría de Tecnología. El objetivo principal de la DGT es “brindar aseguramiento tecnológico a los procesos fundamentales de la universidad, con agilidad y profesionalidad, garantizando la organización y el control de los recursos, logrando un impacto perceptible en la universidad. “La decisión de crear un módulo de configuración propio, en lugar de adaptar uno preexistente, se basa en la necesidad de un ajuste preciso.

Cada sistema tiene sus particularidades, y mientras un módulo externo podría cubrir algunas de estas necesidades, quizás no se adapte perfectamente a todas ellas. Es especialmente relevante cuando se considera que este módulo no es solo un anexo, sino que se establece como la base sobre la cual se construye todo el sistema. Al ser el núcleo, debe ser sólido, coherente y totalmente alineado con los objetivos y características del sistema global, asegurando así un funcionamiento y una expansión futura sin contratiempos.

1.3 Fundamentos teóricos asociados al tema.

A nivel mundial, la Gestión Tecnológica se ha consolidado como el pilar que sostiene el avance vertiginoso de la era digital. Esta gestión implica no solo la administración de recursos tecnológicos sino también la capacidad de anticipar tendencias, garantizar la seguridad y adaptarse a las cambiantes demandas del mercado. En Cuba, y específicamente en la UCI, esta evolución tecnológica ha encontrado un refugio fértil para su crecimiento y desarrollo.

La gestión tecnológica se refiere al conjunto de actividades que una organización realiza para gestionar su tecnología de la información de manera estratégica. Esto implica la planificación, implementación y supervisión de los recursos tecnológicos de una organización, incluidos los sistemas de información. (Impacto Económico y Social de los Servicios Técnicos de la Dirección de Gestión Tecnológica., 2021)

Uno de los pilares de la gestión tecnológica es la infraestructura que es el cimiento sobre el cual se construye toda la arquitectura tecnológica, desde redes de comunicación hasta centros de datos. Una infraestructura sólida garantiza la continuidad de operaciones, la eficiencia en la transmisión de datos y la disponibilidad de servicios. (Impacto Económico y Social de los Servicios Técnicos de la Dirección de Gestión Tecnológica., 2021).

Por otra parte, la gestión tecnológica está estrechamente relacionada con los Sistemas de Gestión de Información (SGI) ya que ambos se ocupan de la utilización efectiva y eficiente de la tecnología de la información en una organización.

Los Sistemas de Gestión de Información (SGI) son una parte vital en el funcionamiento de las organizaciones en la actualidad. Estos sistemas permiten recopilar, almacenar, organizar y administrar la información de una manera eficiente y segura. Con el avance de la tecnología, los SGI han experimentado grandes cambios en los últimos años.

Uno de los principales avances en los SGI es la adopción de tecnologías basadas en la nube. Anteriormente, los sistemas de gestión de información se basaban en servidores locales, lo que implicaba altos costos de mantenimiento y limitaciones en el acceso a la información. Ahora, con la nube, las organizaciones pueden acceder a sus sistemas de información desde cualquier lugar y en cualquier momento, lo que facilita la colaboración y el intercambio de datos entre diferentes departamentos.

Otro avance importante es el uso de la inteligencia artificial (IA) en los SGI. La IA permite a los sistemas de gestión de información analizar grandes cantidades de datos de forma rápida y precisa. Esto es especialmente útil en entornos donde se generan grandes volúmenes de información, como empresas de comercio electrónico o instituciones financieras. La IA también puede ayudar en la detección de patrones y tendencias, lo que facilita la toma de decisiones basada en datos.

La seguridad de la información es otro aspecto fundamental en los SGI. A medida que aumenta la cantidad de información almacenada y compartida, también aumentan los riesgos de seguridad. Por eso, los sistemas de gestión de información actuales están equipados con medidas de seguridad avanzadas, como la encriptación de datos, el control de acceso y la detección de intrusiones. Además, se están desarrollando técnicas de seguridad basadas en el aprendizaje automático para detectar y prevenir amenazas cibernéticas de manera proactiva. (Ito, 2018)

La movilidad es otro aspecto destacado en los SGI modernos. Con el creciente uso de dispositivos móviles, los sistemas de gestión de información ahora están diseñados para ser compatibles con diferentes plataformas, lo que permite acceder a la información desde smartphones y tabletas. Esto es especialmente útil para los empleados que necesitan acceder a la información mientras se encuentran fuera de la oficina.

Además, los SGI actualmente son más intuitivos y fáciles de usar. Con interfaces de usuario amigables y funcionalidades simplificadas, los usuarios pueden acceder y gestionar la información de manera más rápida y eficiente.

Dada la evolución de los SGI la gestión de la Configuración de los mismos pasó a jugar un papel más relevante para garantizar que dichos sistemas sean estables, consistentes y confiables.

La Gestión de la Configuración de Sistemas de Gestión de Información (SGI) se refiere al proceso de supervisar y controlar los elementos de configuración de los sistemas de gestión de información empresarial.

En el mundo de la tecnología, la gestión de la configuración es un proceso de gestión de Tecnología de la información (TI) que supervisa los elementos de configuración individuales de un sistema de TI. Los sistemas de TI se componen de activos informáticos que varían en granularidad. Un activo informático puede representar una parte de un software, un servidor o un clúster de servidores. Lo que sigue se centra en la gestión de la configuración tal y como se aplica directamente a los activos de software de TI y a la Integración Continua (CI) y la Distribución Continua (CD) de activos de software. (BUCHANAN)

En la bibliografía especializada sobre la Gestión de la Configuración de Sistemas de Gestión de Información, el autor del presente trabajo identificó varias tendencias y enfoques que son cada vez más relevantes en el campo.

Uno de los enfoques emergentes es la adopción de la gestión de la configuración basada en la nube. Esto implica utilizar servicios en la nube para almacenar y gestionar la configuración de los sistemas de gestión de información. La gestión basada en la nube permite a las organizaciones acceder a sus datos de configuración desde cualquier lugar y en cualquier momento, lo que facilita la colaboración y mejora la eficiencia.

Además, se ha observado un aumento en la automatización de la gestión de la configuración. Esto implica el uso de herramientas y técnicas automatizadas para realizar tareas como la monitorización de cambios en la configuración, la detección de desviaciones y la aplicación automática de cambios de configuración. La automatización mejora la precisión y la rapidez de los procesos de gestión de la configuración, lo que a su vez reduce los errores humanos y los tiempos de inactividad no planificados.

Otra tendencia importante es la integración de la gestión de la configuración con otras disciplinas de gestión, como la gestión del cambio y la gestión de incidentes. La integración de estas disciplinas permite una mayor visibilidad y coordinación entre los diferentes procesos de gestión, lo que facilita la detección y resolución de problemas en tiempo real.

En cuanto a las mejores prácticas, se ha observado un enfoque en la estandarización y documentación de los procesos de gestión de la configuración. Esto implica definir y documentar claramente los roles, responsabilidades y acciones requeridas en cada etapa del ciclo de vida de la configuración. Además, se han establecido estándares y

marcos de referencia, como ITIL (Information Technology Infrastructure Library), para guiar y mejorar los procesos de gestión de la configuración.

En resumen, la Gestión de la Configuración de Sistemas de Gestión de Información está evolucionando constantemente para adaptarse a las necesidades cambiantes de las organizaciones. Las tendencias actuales incluyen la gestión basada en la nube, la automatización, la integración con otras disciplinas de gestión y la estandarización de los procesos. Estas tendencias y mejores prácticas ayudan a las organizaciones a gestionar eficazmente la configuración de sus sistemas de gestión de información, garantizando así su estabilidad y confiabilidad.

1.4 Análisis de soluciones similares existentes

Con el objetivo de identificar las principales características que debe poseer un módulo de configuración en sistemas de gestión, se realiza un estudio de diferentes sistemas informáticos que hacen uso de él. Actualmente es imprescindible para la mayoría de las soluciones informáticas contar con un producto configurable y adaptable. La personalización es una excelente manera de mejorar las experiencias de los usuarios, ya que les permite decidir específicamente qué desean ver, lo que significa que solo mostrará el contenido que sus usuarios desean. (ZAPROO, 2022). Es por esto que se realiza un análisis de la configuración de sistemas que gestionen recursos tanto a nivel internacional como nacional.

1.4.1 Sistema Integrado de Gestión Administrativa del Ministerio de Economía y Finanzas de Perú SIGA – MEF

El Sistema integrado de gestión administrativa es una aplicación informática desarrollada por la Oficina General de Tecnología de la Información (OGTI) del Ministerio de Economía y Finanzas de Perú (Farro, 2016). Este proyecto se originó en el año 2000 como una necesidad, con la finalidad de lograr la integración de todos los sistemas administrativos en uno solo (2019). Entre las herramientas que presenta contiene una sección “Patrimonio” donde se gestionan los bienes estatales de muebles e inmuebles. Otros apartados del sistema son las herramientas para la gestión de la “Logística”, “Presupuesto”, “Bienes corrientes” y “Tesorería”. La importancia de este sistema se basa en la mejora de la eficiencia de la Gestión Pública de los Procesos de Abastecimiento y Control Patrimonial.

El módulo de configuración de este sistema comprende las funcionalidades de configuración de Logística, Sedes, Feriados, Centros de Costo, Personal, Proveedores, Configuración de la Programación y Carga de Datos del Siga (Ministerio de Economía y

Finanzas, 2021). Entre las funcionalidades se realizan configuraciones para consultar o registrar información ya sea para bienes, servicios e inventarios, además de actualización de lotes y precios.

Por último, es importante destacar que es una aplicación de escritorio que no presenta posibilidades de modificar su estructura o contenido (Farro, 2016). Por tanto, no es posible disminuir la carga de la aplicación retirando alguna sección en caso de no ser necesaria o utilizada. Entre las principales deficiencias actuales del sistema se encuentra la obsolescencia de las tecnologías con las que ha sido desarrollado, ya que no ha contado con un seguimiento, mantenimiento y actualización. Teniendo esto en cuenta se hace necesario el estudio de otra herramienta que permita establecer comparaciones.

1.4.2 Sistema de Gestión Administrativa Escolar de Argentina SIGAEWEB

El sistema de Gestión Administrativa Escolar (SIGAEWEB), es un producto informático integrado, provisto por el Ministerio de Educación de la Provincia de Santa Fe, Argentina (SIGAE WEB, 2018). Es una aplicación web disponible en internet diseñada para empleados en gestión administrativa en colegios.

La integración de la escuela y el Ministerio de Educación en un único sistema facilita la reducción sustancialmente de los tiempos requeridos para la obtención y procesamiento de información (SIGAEWEB, 2013). Las posibilidades que brinda son la gestión de alumnos, carreras, secciones, plazas, inasistencias, calificaciones, servicios alimentarios e infraestructura.

Los beneficios de esta herramienta van desde la disminución de las tareas administrativas, reducción de los tiempos de respuesta en los trámites iniciados por los directores de escuelas y los docentes, así como la disponibilidad online de información de alumnos, secciones y docentes frente alumnos, indicadores pedagógicos y comensales (Santa Fe Provincia, 2022). Cuenta con una sección de configuración en su pantalla principal que brinda posibilidades de cambio de contraseña, gestión de datos de un establecimiento, el registro de días no hábiles y un apartado de mensajes donde se recibe información sobre los cambios que sufre la plataforma.

Algunas de las principales desventajas que presenta este sistema es la dependencia total de conexión a internet para su uso, en caso de que la conexión a Internet sea deficiente o inestable, el acceso a la herramienta podría verse afectado. Actualmente se ha visto afectado su funcionamiento por la interrupción de actualizaciones, lo que ha provocado que presente vulnerabilidades de seguridad. Por último, no es capaz de

adaptar su comportamiento y contenido a diferentes centros educativos con características particulares. A partir de los referentes internacionales que se han estudiado, se requiere el análisis de herramientas que a nivel nacional realicen estas funciones.

1.4.3 Distra

Distra es un sistema de gestión empresarial, que permite integrar y optimizar los procesos empresariales, manejar con eficacia la información y disminuir los costos de las operaciones (XETID, 2023). Este producto lleva a digital todas las áreas de gestión: la logística, el inventario, lo relacionado con la gestión de proyectos, recursos humanos, planificación financiera, y de actividades (MUÑOA, 2022). Es un sistema cubano desarrollado en la XETID, empresa de referencia destinada al desarrollo de tecnologías de la información para la defensa.

Según Antonio Jesús Matos Reyes (2022) las prestaciones de este sistema permiten que desde los hogares los trabajadores puedan acceder al software RP-DISTRA, disponible en la red de redes. El auge de su utilización fue durante el año 2020, donde la COVID-19 afectó la asistencia a los centros de trabajo por lo que se explotó en el país el teletrabajo.

A pesar de ser un producto adaptable a las instituciones que lo requieran, es un sistema que no cuenta con un módulo de configuración que permita modificarlo en su totalidad a las características de cada cliente. Para esto, los especialistas de la XETID deben realizar un estudio previo para analizar las necesidades y proponer, según estas, la distribución más adecuada y la adaptación del software lo que puede requerir un atraso para que el producto final llegue a manos del cliente. Sin embargo, no es posible acceder a este sistema por tener un carácter propietario, por lo que no se puede comprobar de forma empírica cómo se gestionan ciertas características deseadas. Tomando Distra como referente nacional, es posible llegar a ciertas conclusiones sobre la actualidad de estas plataformas en el mercado.

1.4.4 Análisis del estudio realizado a soluciones existentes

Con el propósito de establecer comparaciones entre sistemas homólogos, se presentan a continuación ciertos parámetros que, si bien no pretenden ser definiciones absolutas, son aspectos considerados relevantes por el autor de este trabajo. Estos parámetros se han seleccionado con base en las necesidades identificadas para el sistema en cuestión:

1. Adaptabilidad: Refiere a la capacidad inherente de una aplicación para ajustarse y funcionar en distintos entornos sin requerir modificaciones en su estructura principal. Una aplicación con alta adaptabilidad es aquella que puede integrarse o cambiar según las demandas del entorno sin la necesidad de una reestructuración completa.
2. Aplicación Web: Denota la capacidad de un sistema para ser accesible a través de Internet. Esto implica que los usuarios puedan interactuar con la plataforma desde cualquier lugar y en cualquier momento, siempre que cuenten con una conexión a la web.
3. Control y Gestión de Configuraciones: Este parámetro se refiere a la habilidad del sistema para administrar, organizar y controlar diferentes tipos de configuraciones, ya sean datos, usuarios, recursos u otros componentes. Un sistema eficiente en este aspecto permite una gestión intuitiva y organizada de sus recursos.
4. Tecnología Moderna: Hace énfasis en la implementación de tecnologías actuales y vanguardistas en el desarrollo del sistema. Utilizar tecnologías modernas puede ofrecer ventajas en términos de rendimiento, seguridad y funcionalidad.
5. Seguimiento: Este criterio alude a la importancia de realizar revisiones y mantenimientos periódicos a un sistema por parte de un equipo de desarrollo. Un buen seguimiento asegura que el sistema se mantenga actualizado, corrigiendo posibles fallos y adaptándose a nuevas necesidades.

Teniendo esto en cuenta, se presenta una tabla resumen donde se recogen los datos más significativos de cada una de las soluciones homólogas estudiadas:

Tabla 1 Características generales de las soluciones estudiadas.

SOLUCIONES / **CRITERIOS**	**SIGA-MEF**	**SIGAEWEB**	**DISTRA**
Adaptabilidad	NO	NO	NO
Aplicación Web	NO	SÍ	SÍ
Control y gestión de configuraciones	SÍ	SÍ	SÍ

Tecnología moderna	NO	SÍ	SÍ
Seguimiento	NO	NO	SÍ

Cada uno de los sistemas que aquí se han expuesto tienen como semejanza que carecen de adaptabilidad ya que no permiten adaptarse a diferentes entornos, no sin antes haber realizado su modificación. Por estos motivos no constituyen una solución factible para la DGT.

Los sistemas analizados poseen características semejantes a las necesidades planteadas y se puede afirmar que a pesar de no constituir soluciones factibles, el estudio de estas plataformas permitió entender su funcionamiento haciendo énfasis en como realizan el control y la gestión de configuraciones. El control de usuarios y roles fue otra de las características que se tuvo en cuenta a la hora del desarrollo de la aplicación. Se profundizó, además, buscando la eficiencia del módulo lo que llevó a la conclusión de que era necesario asumir una metodología de desarrollo de software que guiara este proceso.

1.5. Metodología de Desarrollo de Software.

La metodología de investigación es una disciplina de conocimiento encargada de elaborar, definir y sistematizar el conjunto de técnicas, métodos y procedimientos que se deben seguir durante el desarrollo de un proceso de investigación para la producción de conocimiento. (Gitnux, 2023)

Una metodología está compuesta por: (Figueroa, 2012)

- Qué tareas se llevan a cabo en cada etapa: actividades elementales en que dividen los procesos.
- Procedimiento: definición de la forma de ejecutar la tarea.
- Técnica: herramienta utilizada para aplicar un procedimiento.
- Herramienta: para realizar una técnica, se puede apoyar en las herramientas software que automatizan su aplicación.
- Producto: resultado de cada etapa.

Se definió como metodología a utilizar en la presente investigación Proceso Unificado Ágil Variación UCI (AUP-UCI), teniendo en cuenta que es la metodología adaptada al ciclo de vida de los proyectos productivos de la universidad, es ampliamente usada en el área y es extremadamente flexible al proceso de desarrollo de software, además fue un requisito definido por el cliente. AUP-UCI constituye una variante de AUP (Proceso

Unificado Ágil, por sus siglas en inglés). Surge con el objetivo de ser una metodología que se adapte al ciclo de vida definido por la actividad productiva en la universidad.

1.5.1 Metodología AUP-UCI

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Se decidió hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. Una metodología de desarrollo de software tiene entre sus objetivos aumentar la calidad del software que se produce, de ahí la importancia de aplicar buenas prácticas, el cual constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora. Estas prácticas se centran en el desarrollo de productos y servicios de calidad (Rodríguez, (2014).)

1.5.2 Descripción de las fases de AUP en su variación UCI

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, a la que llamaremos Ejecución y se agrega la fase de Cierre. Para una mejor comprensión se muestra la tabla con la descripción de la fase de AUP-UCI.

Tabla 2 Descripción de las fases AUP-UCI. Fuente Elaboración propia.

Fases AUP	**Fases Variación AUP-UCI**	**Objetivos de las fases (Variación AUP-UCI)**
Inicio	Inicio	Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.

Elaboración	Ejecución	En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
Construcción		
Transición		
	Cierre	En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

1.5.3 Escenarios en cuanto a Requisitos

A partir de que el Modelado de negocio propone tres variantes a utilizar en los proyectos como Casos de Uso del Negocio(CUN), Descripción de Requisitos de Negocio(DPN), o Modelo Conceptual(MC) y existen tres formas de encapsular los requisitos como Casos de Uso del Sistema(CUS), Historias de Usuarios(HU) y Descripción de Requisitos por Proceso(DRP), surgen cuatro escenarios para modelar el sistema en los proyectos, manteniendo en dos de ellos el MC, quedando de la siguiente forma:

Escenario No1: Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.

Figura 1 Escenario 1 AUP-UCI

Escenario No2: Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.

Figura 2 Escenario 2 AUP-UCI

Escenario No3: Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.

Figura 3 Escenario 3 AUP-UCI

Escenario No4: Proyectos que no modelen negocio solo pueden modelar el sistema con HU.

Figura 4 Escenario 4 AUP-UCI

Todas las disciplinas antes definidas (desde Modelado de negocio hasta Pruebas de Aceptación) se desarrollan en la Fase de Ejecución, de ahí que en la misma se realicen Iteraciones y se obtengan resultados incrementales. En una iteración se repite el flujo de trabajo de las disciplinas, Requisitos, Análisis y diseño, Implementación y Pruebas internas. De esta forma se brinda un resultado más completo para un producto final de manera creciente. Para llegar a lograr esto, cada requisito debe tener un completo desarrollo en una única iteración (Rodríguez, 2015)

Luego de analizar los escenarios propuestos por la variante AUP-UCI para la actividad productiva en la UCI, el autor de la presente investigación seleccionó el escenario número **cuatro**, ya que el negocio a informatizar está bien definido, el equipo de desarrollo está conformado por una sola persona, el proyecto a desarrollar no es extenso, y el cliente acompaña al equipo de desarrollo todo el tiempo.

Por tanto, esta metodología guiará al desarrollo de la aplicación aplicando los valores y principios ágiles al proceso unificado iterativo e incremental para permitir entregas rápidas de software de alta calidad, se adapta a las condiciones existentes y además está avalada por el departamento de Calidad-UCI. Para garantizar el éxito en el desarrollo de software, aparte de implementar una metodología apropiada, es crucial realizar la selección adecuada de los lenguajes de programación que permitan construir el sistema de forma eficiente.

1.6. Lenguajes de programación.

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis;

que se pone a disposición del programador para que pueda comunicarse con los dispositivos hardware y software existentes. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo. (Gutierrez, 2020)

1.6.1. Python.

Para el desarrollo del nuevo módulo que se integrará al sistema existente escrito en Python, se ha optado por utilizar Python como lenguaje de programación debido a las siguientes razones técnicas: (Castillo, Celia Clemente, 2008) (infostudio, 2011)

- Consistencia tecnológica: Al estar desarrollado el sistema Backend principal en Python, construir el nuevo módulo en el mismo lenguaje permite mantener una estructura tecnológica consistente y evita complejidades innecesarias en la integración de componentes con distintas tecnologías.
- Curva de aprendizaje: El equipo de desarrollo que trabaja en el sistema Backend está familiarizado con Python y su ecosistema, por lo que no es necesario incurrir en la curva de aprendizaje de adoptar un nuevo lenguaje y framework.
- Madurez de la plataforma: Python es un lenguaje ampliamente utilizado y testeado para aplicaciones web Backend, con un ecosistema rico en frameworks y librerías que facilitan y aceleran el desarrollo. Su madurez reduce la probabilidad de enfrentar errores o limitaciones del lenguaje.
- Rendimiento probado: La infraestructura y configuraciones actuales del sistema Backend están optimizadas para ejecutar aplicaciones de manera eficiente y escalable. Utilizar Python para el nuevo módulo aprovecha estas optimizaciones.
- Seguridad: Python permite construir aplicaciones seguras, especialmente al utilizar frameworks que proveen funciones y buenas prácticas probadas para evitar vulnerabilidades comunes.

1.6.2 HTML

HTML es el lenguaje con el que se escriben las páginas web. Es un lenguaje de hipertexto, es decir, un lenguaje que permite escribir texto de forma estructurada, y que está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento del documento. (W3Schools) ((W3C), 2014) (Carnes, 2021)

Las razones por las cuales se utilizará HTML en el proyecto son:

- HTML es el lenguaje base para crear el contenido y la estructura de las páginas web. Es un estándar necesario en la web.

- Se separa el contenido de la presentación. El HTML se enfoca en la información, mientras que con CSS se define la parte visual.
- Es soportado por todos los navegadores web permitiendo crear contenido accesible desde cualquier dispositivo.
- Permite incorporar contenidos multimedia como imágenes, videos y audio de forma sencilla.
- Existe gran compatibilidad con otros lenguajes web como JavaScript, CSS, XML, etc.
- Hay muchas librerías, frameworks y herramientas que potencian y facilitan el desarrollo en HTML.

1.6.3 CSS ("Hojas de Estilo en Cascada"). Tailwind CSS

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML (Lenguaje de marcado de hipertexto extensible, eXtensible HyperText Markup Language). Además, se estará utilizando Tailwind CSS que no es más que un framework de desarrollo que a diferencia de otros frameworks CSS que se centran en proporcionar estilos prediseñados para elementos específicos, Tailwind CSS proporciona clases de utilidad que pueden ser aplicadas directamente en el código HTML para estilizar elementos y diseñar los sitios web de manera más eficiente. Las ventajas de utilizar CSS con Tailwind CSS en el proyecto son: (MDN) (CM, 2009) (tailwindcss, 2020)

- Permite separar el contenido HTML de la presentación visual, haciendo el código más limpio, escalable y mantenible.

- Tailwind CSS acelera el proceso de desarrollo web al eliminar gran parte del trabajo repetitivo de escribir CSS desde cero o buscar clases específicas en un archivo CSS.

- Se pueden crear diseños y temas consistentes fácilmente al definir estilos comunes en archivos CSS externos.

- Facilita aplicar diseño responsivo en distintos dispositivos mediante media queries y layouts flexibles.

- Aunque Tailwind CSS se centra en clases de utilidad, aún puedes crear componentes reutilizables mediante la combinación de clases o la creación de clases personalizadas. Esto facilita la construcción de interfaces coherentes en todo tu sitio web.

- Mejora el tiempo de carga de páginas web al poder hacer uso de los archivos CSS externos.
- Tailwind CSS se puede combinar fácilmente con JavaScript y frameworks de JavaScript

1.6.4 JavaScript.

Para el desarrollo del módulo del lado del Frontend, se utilizó JavaScript con el fin de mantener una estructura tecnológica consistente y evitar complejidades innecesarias en la integración de componentes con distintas tecnologías. (J, 2009.) (Hernández Claro R, 2010.) (Salvaggio, 2019)

Es un lenguaje interpretado, por lo que no es necesario compilar los programas para ejecutarlos. JavaScript nació con la necesidad de permitir a los autores de sitio web crear páginas que permitan intercambiar con los usuarios, ya que se necesitaba crear webs de mayor complejidad. Dentro de las facilidades que brinda al proyecto están:

- Permite crear interfaces de usuario interactivas ya que puede modificar dinámicamente el HTML y CSS.
- Se pueden construir aplicaciones web complejas del lado del cliente como aplicaciones de una sola página (SPA).
- Los efectos y animaciones dinámicas son posibles sin necesidad de recargar la página.
- Da acceso al DOM para manipular y modificar elementos HTML y CSS.
- Tiene una sintaxis familiar a otros lenguajes de programación como Java o C++.
- Cuenta con un ecosistema muy amplio de librerías y framework para tareas específicas.

1.7. Marco de trabajo (Framework).

En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente, con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

1.7.1 Django y Django REST

Django es un framework de desarrollo web completo que facilita la creación de aplicaciones web, y Django REST framework se utiliza para extender Django y crear APIs RESTful de manera eficiente. Juntos, permiten a los desarrolladores construir aplicaciones web con interfaces de usuario tradicionales y proporcionar APIs para interactuar con datos de manera programática. Esto es especialmente útil en aplicaciones web que requieren servicios web para aplicaciones móviles, clientes web, y más. (django, 2023) (REST, 2011)

- Desarrollo Completo: Django ofrece una solución completa para el desarrollo web, incluyendo la gestión de bases de datos, autenticación, seguridad y generación de vistas web. Django REST framework se integra perfectamente para manejar la capa de servicios web y APIs.
- Eficiencia: Al utilizar Django REST framework junto con Django, permite crear rápidamente tanto una interfaz de usuario tradicional como una API web en un solo proyecto, lo que ahorra tiempo y recursos de desarrollo.
- Consistencia: Ambos frameworks comparten la misma filosofía y estilo de desarrollo, lo que facilita mantener una coherencia en la estructura y el flujo de datos entre la interfaz de usuario y la API.
- Escalabilidad: Django es conocido por su capacidad para manejar aplicaciones web escalables y de alto tráfico. Utilizar Django REST framework permite escalar tanto la interfaz de usuario como la API de manera eficiente.
- Documentación Automática: La documentación automática de Django REST framework es especialmente útil cuando se está construyendo una API.
- Seguridad Integrada: Ambos frameworks tienen medidas de seguridad incorporadas, lo que ayuda a garantizar que tanto la interfaz de usuario como la API sean seguras contra amenazas comunes.

1.7.2 Next.js

Next.js es un framework popular de React para construir aplicaciones web y tiene varias ventajas que lo convierten en una elección sólida para consolidar el desarrollo. Aquí algunas razones: (NEXT.js, 2023) (Thelin, 2021)

- Next.js está diseñado para ofrecer un rendimiento óptimo. Implementa la representación del lado del servidor (SSR) de forma predeterminada, lo que mejora la velocidad de carga de la página y ayuda con el SEO al proporcionar contenido indexable por los motores de búsqueda.

- Simplifica el desarrollo de aplicaciones React al proporcionar una estructura de proyecto predefinida y herramientas como el enrutamiento integrado, la recarga en caliente y la optimización automática de imágenes. Esto acelera el desarrollo y permite a los desarrolladores concentrarse en la lógica de la aplicación en lugar de la configuración.
- Next.js facilita la creación de rutas dinámicas, lo que es especialmente útil si se está construyendo una aplicación con muchas páginas o rutas diferentes.
- Tiene una comunidad activa y una gran cantidad de recursos disponibles, lo que facilita encontrar ayuda, tutoriales y bibliotecas adicionales.
- Es compatible con múltiples plataformas de alojamiento, lo que facilita el despliegue.
- Aunque Next.js tiene una estructura predefinida, también es lo suficientemente flexible como para adaptarse a necesidades específicas.

1.8. Sistema gestor de Base de Datos.

Una Base de Datos (BD) es una colección de datos pertenecientes a un mismo entorno, organizados para su búsqueda y recuperación. Existen programas computarizados denominados Sistemas Gestores de Bases de Datos (SGBD) que permiten almacenar y posteriormente recuperar y actualizar estos datos de forma rápida y estructurada.

Un SGBD es un sistema computarizado cuya finalidad general es almacenar información y permitir a los usuarios recuperar y actualizar la misma sobre la base de peticiones. La información en cuestión puede ser cualquier elemento que sea de importancia para el individuo u organización; en otras palabras, todo lo que sea necesario para auxiliarle en el proceso general de su administración.

1.8.1 PostgreSQL.

A continuación, se hace un estudio a los Sistema gestor de Base de datos PosgreSQL y MySQL: (Bhatia, y otros, 2023)

- Estándares: PostgreSQL cumple mejor los estándares SQL y ACID que MySQL.
- Rendimiento: PostgreSQL tiene un mejor desempeño en cargas de trabajo intensas y consultas complejas.
- Tipos de datos: PostgreSQL soporta una mayor variedad de tipos de datos avanzados como arrays, geoespacial, JSON, etc.

- Concurrencia: PostgreSQL maneja mejor el bloqueo, control de concurrencia y escritura en disco.
- Extensibilidad: PostgreSQL permite ampliar fácilmente su funcionalidad con extensiones, usuarios y funciones.
- Estabilidad: PostgreSQL tiene una mayor estabilidad y menos chances de corromper datos ante fallos.
- Comunidad: PostgreSQL tiene una comunidad más grande y activa que contribuye constantemente.

En conclusión, si bien MySQL es popular por su facilidad de uso, PostgreSQL es superior en rendimiento, estabilidad, estándares y extensibilidad. Es recomendable para aplicaciones empresariales que requieren alta concurrencia, y consultas analíticas complejas.

1.9 Otras Herramientas

Se emplearon diversas herramientas tecnológicas para facilitar el desarrollo y modelado de la solución propuesta. Entre ellas:

- Editor de código para la implementación ágil y efectiva.
- Software de modelado UML para el diseño gráfico y conceptual.

Las aplicaciones usadas proveyeron las capacidades necesarias para un proceso de construcción completo, desde el modelado inicial hasta la codificación final de la solución. La selección apropiada de tecnologías de soporte fue clave para optimizar el desarrollo. A continuación, se describen las herramientas utilizadas:

1.9.1. Lenguaje de Modelado.

Unified Modeling Language (UML), Lenguaje Unificado de Modelado traducido al español es el lenguaje más utilizado para el diseño orientado a objetos.

Permite a un diseñador describir el proyecto con una rica variedad de esquemas. Por ejemplo, los diagramas de clases UML pueden mostrar los elementos importantes o relevantes de las clases, y las relaciones entre ellos, de forma concisa e intuitiva. Otros tipos de diagramas UML pueden ilustrar cómo colaboran las clases entre sí y cómo los usuarios interactúan con objetos de clase.

UML ofrece soporte para clases, clases abstractas, relaciones, comportamiento por interacción y empaquetamiento. Estos elementos se pueden representar mediante nueve tipos de diagramas, que son: de clases, de objetos, de casos de uso, de

secuencia, de colaboración, de estados, de actividades, de componentes y de desarrollo. (MARTIN FOWLER, 2003.)

1.9.2 Herramienta para el modelado. Visual Paradigm

Las Herramientas CASE (Ingeniería de Software Asistida por Ordenador, Computer Aided Software Engineering,) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software, en tareas como: el proceso de realizar un diseño del proyecto, cálculo de costes, implementación automática de parte del código con el diseño dado, compilación automática y documentación o detección de errores. (MÉNDEZ, 2013)

A continuación, se hace un estudio a las herramientas de modelado Visual Paradigm y UML Designer: (Obeo., (s.f.)) (Lianet Cabrera González, 2012.)

- Alcance de modelado: Visual Paradigm permite modelar una amplia variedad de diagramas UML, así como modelado de procesos de negocio con BPMN y modelado de bases de datos. UML Designer se enfoca específicamente en diagramas UML.
- Integraciones: Visual Paradigm permite integrarse con otras herramientas de desarrollo como IDEs y sistemas de gestión de proyectos. UML Designer tiene integración limitada con algunas IDEs.
- Generación de código: Visual Paradigm puede generar código fuente a partir de los diagramas UML para varios lenguajes. UML Designer no genera código.
- Simulación: Visual Paradigm puede simular diagramas de máquina de estados y procesos de negocio. UML Designer no tiene capacidades de simulación.
- Colaboración: Visual Paradigm facilita el modelado colaborativo en tiempo real. UML Designer no tiene esta capacidad.
- Licenciamiento: Visual Paradigm tiene versiones comunitarias gratuitas pero las profesionales son pagas. UML Designer es completamente gratuito.

En conclusión, a pesar de que UML Designer es una buena herramienta de código abierto para modelado UML básico, Visual Paradigm presenta funcionalidades más avanzadas como generación de código, simulación y trabajo colaborativo, por lo cual es la herramienta que se utilizará en el proyecto. La versión comunitaria gratuita de Visual Paradigm puede ser suficiente para muchos casos de uso.

1.9.3 Entorno de Desarrollo Integrado. Visual Studio Code (VS Code)

Un Entorno Integrado de Desarrollo en inglés Integrated Development Enviroment (IDE) es un programa compuesto por un conjunto de herramientas a disposición de los desarrolladores que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien poder utilizarse para varios. Se estará utilizando el VS Code que es un entorno de desarrollo integrado altamente popular y ampliamente utilizado para escribir, editar y depurar código en una variedad de lenguajes de programación. Se caracteriza por ser gratuito, de código abierto y altamente personalizable. (CODE, 2023) (CODE, 2023) (Arthorne, 2007)

Entre sus principales características están:

- VS Code es completamente gratuito y de código abierto.
- Está disponible para Windows, macOS y Linux, lo que lo hace accesible para una amplia variedad de usuarios en diferentes sistemas operativos.
- Es compatible con una amplia gama de lenguajes de programación, lo que lo hace versátil y adecuado para desarrolladores que trabajan en diferentes tecnologías.
- Una de las características más destacadas de VS Code es su sistema de extensiones. Puedes instalar extensiones específicas para tu flujo de trabajo de desarrollo, lo que te permite personalizar y ampliar las capacidades del IDE según tus necesidades.
- Ofrece una sólida capacidad de depuración integrada que permite realizar un seguimiento y solucionar problemas en el código de manera eficiente.
- VS Code está diseñado para ser rápido y eficiente, incluso cuando se trabaja en proyectos grandes.

Conclusiones Parciales

En el capítulo analizado, se llevaron a cabo evaluaciones de sistemas informáticos relevantes, lo que contribuyó significativamente a una comprensión más profunda del problema bajo estudio. A pesar de que las plataformas similares revisadas no satisfacían por completo las necesidades del módulo en cuestión, su análisis fue fundamental para

recopilar tendencias e información sobre los sistemas de Gestión de Configuración. Se reveló que, aunque existen sistemas que pueden cumplir con algunas de las demandas planteadas por el problema de investigación, no resultaron completamente aplicables debido a la falta de cumplimiento con los requisitos necesarios.

La selección cuidadosa de herramientas, junto con la aplicación de la metodología AUP en su variante UCI, así como la elección de un lenguaje de programación y un lenguaje de modelado adecuados, contribuyeron de manera significativa a la definición precisa de la estructura y el diseño del proyecto. Esto proporcionó información crucial para la ejecución exitosa del módulo, asegurando el cumplimiento de las mejores prácticas en el desarrollo de software. Además, este enfoque agilizó el desarrollo, ofreciendo una solución completa y eficiente para la implementación del proyecto.

Capítulo 2: Análisis y Diseño

2.1 Introducción

En el presente capítulo se ejemplifican las características del sistema a desarrollar, definiendo el modelo de dominio según el objeto de estudio haciendo uso de la metodología AUP-UCI. Se identificaron los requisitos con los que debe cumplir la herramienta para el despliegue, se diseña la arquitectura empleada, así como los patrones de diseño que son la base para la implementación de la propuesta de solución. De igual forma se presenta el diagrama de despliegue con el fin de describir la distribución física del sistema.

2.2 Propuesta de Solución

Para abordar la problemática identificada en la DGT de la UCI en relación con la gestión de configuraciones del Sistema de Gestión de Información, se desarrolló un módulo de configuración que está diseñado para operar en una interfaz web, facilitando su accesibilidad y adaptabilidad a diferentes dispositivos. Incorpora funcionalidades que permiten la gestión eficiente de los usuarios, roles y nomencladores.

Adicionalmente, el sistema proporciona herramientas para la asignación dinámica de roles y permisos, garantizando así una gestión de seguridad adecuada. Con el fin de garantizar la integridad de los datos se implementó un sistema de validación de entradas y un seguimiento de trazas y notificaciones para auditar las acciones realizadas. Este módulo no solo automatiza los procesos internos de la DGT, sino que también reduce las pérdidas de información, minimiza errores humanos y facilita la recopilación y análisis de estadísticas, fortaleciendo así la misión de la UCI en su avance hacia la transformación digital y la soberanía tecnológica.

2.4. Especificación de requisitos del software

Los requisitos de un sistema son una descripción detallada de las necesidades y expectativas de los usuarios, clientes y otras partes interesadas que el sistema debe cumplir para ser considerado exitoso. En otras palabras, los requisitos son una especificación formal de las funciones, características y restricciones que el sistema debe tener para satisfacer las necesidades del usuario final. (Alegsa, 2023)

2.4.1 Requisitos Funcionales

Según (Pressman, 2015) Un requisito funcional es una declaración de los servicios que el sistema debe proveer, cómo el sistema debe reaccionar a entradas particulares y cómo el sistema debe comportarse en situaciones particulares.

Estos son importantes para el proceso de diseño y desarrollo del software, ya que proporcionan una guía clara sobre lo que el sistema debe hacer y cómo debe hacerlo. Estos requisitos también son esenciales para la validación y verificación del sistema, ya que permiten a los desarrolladores y usuarios confirmar que el sistema cumple con las expectativas y necesidades de los usuarios.

A continuación, se describen los requisitos funcionales del módulo a implementar:

Tabla 3 Requisitos Funcionales del Sistema.

No.	***Nombre***	***Descripción***	***Prioridad***	***Complejidad***
RF 1	Autenticar Usuario	El sistema debe permitir la autenticación de un usuario completando los campos: (Correo y contraseña)	*Alta*	*Alta*
RF 2	Autenticar Usuario usando servicio LDAP de la UCI	El sistema debe permitir la autenticación de un usuario utilizando el servicio de autenticación de la UCI completando los campos: (Correo y contraseña)	*Media*	*Alta*
RF 3	Crear Usuario	El sistema debe permitir crear un usuario completando los siguientes campos: (nombre, apellidos, usuario, rol, correo)	*Alta*	*Alta*
RF4	Eliminar Usuario	El sistema debe permitir eliminar el/los usuarios creados anteriormente.	*Media*	*Alta*
RF 5	Editar Usuario	El sistema debe permitir editar el/los usuarios creados anteriormente con los campos: (nombre, apellidos, usuario, rol, correo)	*Media*	*Alta*

RF 6	Listar Usuarios	El sistema debe permitir listar todos los usuarios creados anteriormente.	*Media*	*Alta*
RF 7	Crear Rol	El sistema debe permitir crear un rol completando los siguientes campos:(nombre, permisos)	*Alta*	*Alta*
RF 8	Eliminar Rol	El sistema debe permitir eliminar el/los roles creados anteriormente.	*Media*	*Alta*
RF 9	Editar Rol	El sistema debe permitir editar el/los roles creados anteriormente con los campos: (nombre, permisos)	*Media*	*Alta*
RF 10	Listar Roles	El sistema debe permitir listar todos los roles creados anteriormente.	*Media*	*Alta*
RF 10	Listar Notificaciones del usuario	El sistema debe permitir listar las notificaciones propias de cada usuario	*Alta*	*Media*
RF 11	Editar una notificación	El sistema debe permitir editar una notificación propia del usuario con el fin de marcarlas como vistas	*Baja*	*Media*
RF 12	Eliminar una notificación	El sistema debe permitir eliminar una notificación propia del usuario	*Baja*	*Media*
RF 13	Enviar Notificaciones	El sistema debe permitir enviar notificaciones a los usuarios.	*Media*	*Alta*
RF 14	Crear Identificador de Nomenclador	El sistema debe permitir crear un identificador de nomenclador completando los siguientes campos:(nombre)	*Alta*	*Baja*

RF 15	Editar Identificador de Nomenclador	El sistema debe permitir editar el/los identificadores de nomencladores creados anteriormente con los campos: (nombre)	*Alta*	*Baja*
RF 16	Eliminar Identificador de Nomenclador	El sistema debe permitir eliminar el/los identificadores de nomencladores creados anteriormente.	*Alta*	*Baja*
RF 17	Listar Identificadores de Nomencladores	El sistema debe permitir listar los identificadores de nomencladores	*Alta*	*Baja*
RF 18	Agregar nuevo dato a un nomenclador	El sistema debe permitir agregar una nueva instancia a un nomenclador determinado.	*Alta*	*Alta*
RF 19	Editar un dato de un nomenclador	El sistema debe permitir editar una instancia ya existente de un nomenclador determinado.	*Media*	*Alta*
RF 20	Eliminar un dato de un nomenclador	El sistema debe permitir eliminar una instancia ya existente de un nomenclador determinado.	*Media*	*Alta*
RF 21	Listar datos de un nomenclador	El sistema debe permitir listar los datos ya existentes de nomenclador determinado	*Media*	*Alta*
RF 22	Crear Ruta	El sistema debe permitir crear una ruta completando los siguientes campos: (ruta y rol)	*Alta*	*Alta*
RF 23	Eliminar Ruta	El sistema debe permitir eliminar la/las rutas creadas anteriormente.	*Media*	*Alta*
RF 24	Editar Ruta	El sistema debe permitir editar la/las rutas creadas	*Media*	*Alta*

		anteriormente con los campos: (ruta y rol)		
RF 25	Listar Ruta	El sistema debe permitir listar todas las rutas creadas anteriormente.	*Media*	*Alta*
RF 26	Listar Historial	El sistema debe permitir listar todos los datos del historial.	*Baja*	*Baja*

2.4.2 Requisitos no funcionales

Según (Pressman, 2015) Un requisito no funcional es una restricción en los servicios o funciones ofrecidos por el sistema. Incluyen requisitos de tiempo, requisitos de procesamiento, requisitos de seguridad, disponibilidad, mantenibilidad, confiabilidad, y otras. Los requisitos no funcionales imponen restricciones en el diseño o la implementación del sistema.

Estos requisitos son importantes para el proceso de diseño y desarrollo del software, ya que ayudan a los desarrolladores a diseñar un sistema que cumpla con los estándares y expectativas del usuario final en términos de calidad, eficiencia, seguridad y usabilidad.

El sistema propuesto cuenta con los siguientes requisitos no funcionales:

RNF1 Requisitos de Usabilidad

RNF 1.1 El sistema deberá permitir encontrar cualquier función en no más de 2 clics y en menos de 5 segundos.

RNF 1.2 Los usuarios nuevos deberán poder usar el 80% de funciones sin ayuda luego de 30 minutos de uso.

RNF 1.3 El 90% de las consultas y búsquedas deberán obtener respuesta en menos de 3 segundos. Para el resto el tiempo máximo será de 5 segundos.

RNF2 Requisitos de Rendimiento

RNF 2.1 El 90% de las operaciones deben responder en menos de 3 segundos. El tiempo máximo de respuesta para cualquier operación no deberá exceder los 5 segundos.

RNF 2.2 El sistema permitirá un mínimo de 50 usuarios concurrentes sin pérdida de rendimiento. El tiempo de respuesta no deberá incrementarse en más de 0.5 segundos con 50 usuarios concurrentes.

RNF 2.3 El sistema será capaz de manejar una carga pico de 5,000 transacciones por segundo sin pérdida de rendimiento.

RNF3 Requisitos de Software

RNF 3.1 El sistema deberá funcionar correctamente en las últimas versiones de los navegadores Mozilla Firefox, Google Chrome y Microsoft Edge.

RNF 3.2 El sistema deberá funcionar correctamente y ser totalmente operativo en dispositivos móviles, incluyendo smartphones y tablets, con sistemas operativos iOS y Android.

RNF 3.3 El diseño del sistema debe adaptarse automáticamente a diferentes resoluciones de pantalla, garantizando una experiencia de usuario coherente y funcional en monitores de distintos tamaños y proporciones.

RNF4 Requisitos de Mantenibilidad

RNF 4.1 La documentación del sistema debe actualizarse cada vez que se realice un cambio en las funcionalidades.

RNF 4.2 El código fuente deberá tener una cobertura de pruebas unitarias de al menos 80%.

RNF5 Requisitos de Seguridad

RNF 5.1 Las contraseñas de los usuarios deben tener un mínimo de 8 caracteres.

RNF 5.2 El código fuente deberá analizarse con herramientas de análisis estático para detectar vulnerabilidades

RNF 5.3 Las sesiones de usuario deben caducar luego de 24 horas de inactividad y en caso de continuar en el sistema por más de 30 minutos deberá refrescar la sesión.

2.5 Historia de usuario

La metodología de desarrollo seleccionada define cuatro escenarios para llevar a cabo la modelación del sistema en los proyectos. De acuerdo a las características del proyecto de desarrollo a quien tributa la propuesta de solución, se selecciona el escenario número cuatro. En este escenario, el proyecto evaluó previamente el negocio a informatizar y como resultado obtuvo un negocio muy bien definido. Además, el cliente está siempre acompañando al equipo de desarrollo para acordar los detalles de los requisitos y así poder implementarlos y probarlos. El artefacto que se genera a través de este escenario es la historia de usuario (Villar, 2023)

Las Historias de Usuario (HU) son descripciones de funcionalidades del sistema desde la perspectiva del usuario final. Cada historia incluye una explicación de las expectativas del usuario y su importancia, junto con criterios de aceptación para determinar la completitud de la funcionalidad. Estas historias son herramientas clave para centrar el desarrollo de software en las necesidades del usuario y fomentar la colaboración entre desarrolladores y usuarios. Además, son flexibles y se pueden actualizar y refinar a medida que avanza el desarrollo y se comprenden mejor las necesidades del usuario.

Tabla 4: Historia de Usuario<Gestionar Usuarios>.

Historia de Usuario	
Número: 1	**Nombre de HU:** Gestionar Usuarios
Programador: Aarom Cárdenas Hernández	**Iteración Asignada:** 1era
Prioridad: Alta	**Tiempo Estimado:** 5 días
Riesgo en Desarrollo: Alto	**Tiempo Real:** 3 días
Descripción:	
1. **Objetivo:** • Ofrece al usuario una vista que permite agregar, eliminar, editar y listar los usuarios con su nombre, apellidos, usuario, rol y correo. 2. **Flujo de la acción a realizar:** • El sistema debe permitir crear, eliminar, editar y listar el/los usuarios con su nombre, apellidos, usuario, rol, contraseña y correo. **3. Flujo alterno** • El sistema debe permitir ir “Atrás” o “Cancelar” en caso de que la opción seleccionada no sea la deseada.	
Prototipo de Interfaz:	

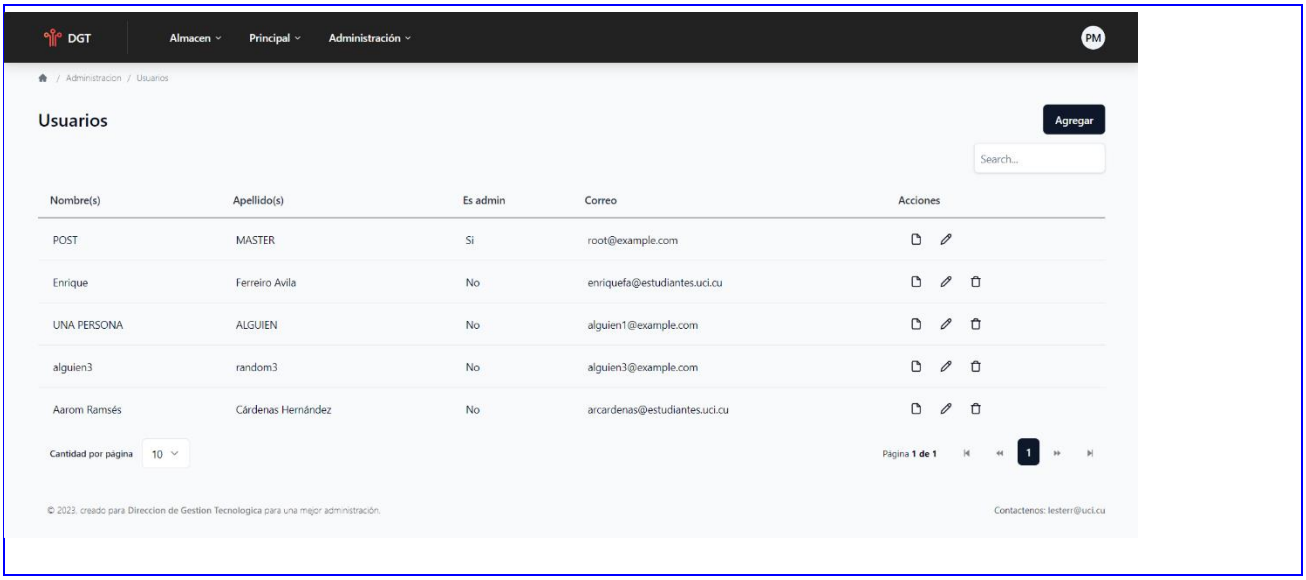
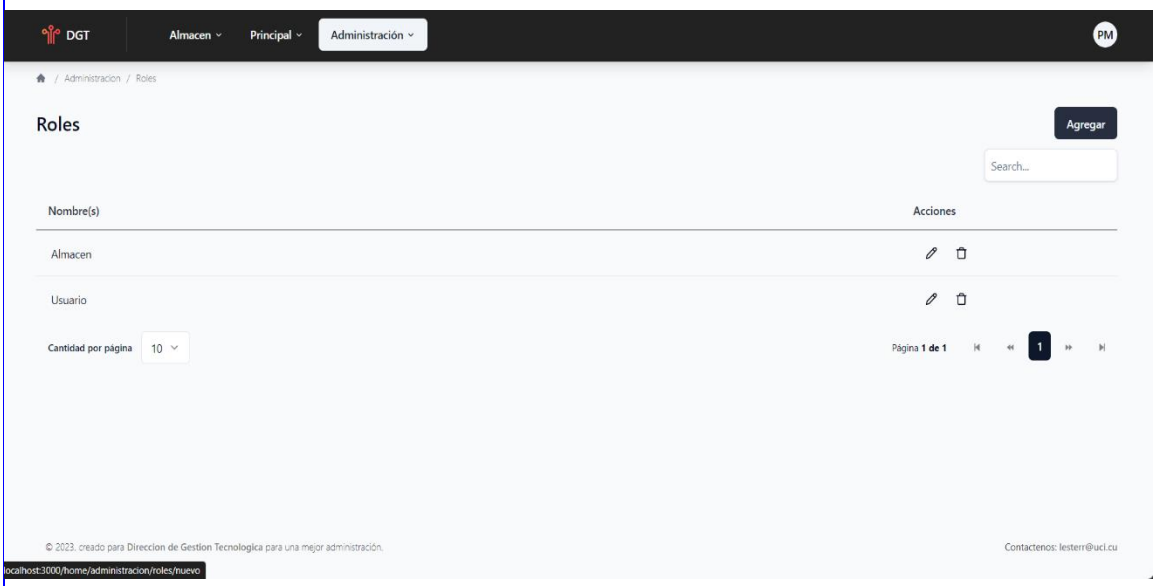


Tabla 5 Historia de Usuario<Gestionar Roles>

Historia de Usuario	
Número: 2	**Nombre del requisito:** Gestionar Roles
Programador: Aarom Cárdenas Hernández	**Iteración Asignada:** 1era
Prioridad: Alta	**Tiempo Estimado:** 7 días
Riesgo en Desarrollo: Alto	**Tiempo Real:** 4 días
Descripción:	

1. **Objetivo:**
   - Ofrece al usuario una vista que permite agregar, eliminar, editar y listar los roles con su nombre y permisos.
2. **Flujo de la acción a realizar:**
   - El sistema debe permitir agregar, eliminar, editar y listar el/los roles con su nombre y permisos asociados.
3. **Flujo alterno**
   - El sistema debe permitir ir “Atrás” o “Cancelar” en caso de que la opción seleccionada no sea la deseada.

Prototipo de Interfaz:



2.6 Diseño

La esencia del diseño de software es la toma de decisiones sobre la organización lógica del software. Algunas veces, se representa la organización lógica como un modelo en un lenguaje definido de modelado tal como UML y otras veces simplemente utiliza notaciones informales y esbozos para representar el diseño. El propósito concreto del diseño es especificar la forma y comportamiento del sistema desde una vista menos abstracta que el análisis, es decir, se centra en una implementación concreta que debe tomar en cuenta los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia.

2.6.1 Arquitectura de Software

Según Sommerville (2011) en su libro “Ingeniería de Software”, la arquitectura de software es la estructura o estructuras del sistema, que comprenden elementos de software, las propiedades visibles externamente de esos elementos y las relaciones entre ellos. La arquitectura es el esqueleto o base de una aplicación, en esta se analiza el sistema desde varios puntos de vista, brindando una clara perspectiva de la misma necesaria para controlar el desarrollo.

Es el conjunto de técnicas metodológicas desarrolladas con el fin de facilitar la programación. Hace referencia a un grupo de abstracciones y patrones que brindan un esquema de referencia útil para guiarse en el desarrollo de software dentro de un sistema informático. Establece todos los fundamentos para que los analistas,

diseñadores y programadores trabajen en una línea común que permita alcanzar los objetivos y necesidades del sistema.

Teniendo en cuenta los problemas antes descritos que posee actualmente la DGT se propone para el desarrollo una arquitectura que permita construir sistemas escalables y flexibles, con separación de preocupaciones entre interface de usuario y lógica de negocio. La arquitectura que se decidió utilizar para la elaboración del proyecto es la arquitectura Cliente Servidor por las siguientes razones:

Modelo Cliente – Servidor: Se define como sistema distribuido a un conjunto o grupo de equipos que son independientes entre sí y que actúan como un único equipo de forma transparente y que tienen como objetivo la descentralización del procesamiento o el almacenamiento de información.

En todo sistema distribuido se establecen una o varias comunicaciones siguiendo un protocolo prefijado mediante un esquema cliente-servidor. Los sistemas cliente-servidor están construidos de tal modo que la base de datos puede residir en un equipo central llamado servidor y ser compartida entre varios usuarios. Los usuarios tienen acceso al servidor a través de una aplicación de cliente o de servidor. (VALLE, 2018)

El tener los datos almacenados y administrados en una ubicación central ofrece varias ventajas: (Arismendiz, 2017)

- ✓ Todos los elementos de datos están almacenados en una ubicación central en donde todos los usuarios pueden trabajar con ellos.
- ✓ No se almacenan copias separadas del elemento en cada cliente, lo que elimina los problemas de hacer que todos los usuarios trabajen con la misma información.
- ✓ Las reglas de la organización y las reglas de seguridad se pueden definir una sola vez en el servidor para todos los usuarios.
- ✓ Los servidores de base de datos relacionales optimizan el tráfico de la red al devolver solo los datos que la aplicación necesita.
- ✓ Los gastos en hardware se pueden minimizar. Como los datos no están almacenados en los clientes, estos no tienen que dedicar espacio de disco a almacenarlos. Los clientes tampoco necesitan la capacidad de proceso para administrar los datos localmente y el servidor no tiene que dedicar capacidad de proceso para presentar los datos.

Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes

- ✓ El servidor presenta a todos sus clientes una interfaz única y bien definida.
- ✓ El cliente no necesita conocer la lógica del servidor, solo su interfaz externa.

- ✓ El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- ✓ Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Con el objetivo de guiar el proceso de desarrollo se seleccionó la arquitectura cliente-servidor debido a sus beneficios específicos para nuestro sistema, debido a su capacidad para manejar cargas de trabajo pesadas, su capacidad para dividir la aplicación en dos partes separadas y escalables, y su capacidad para mejorar la mantenibilidad y la seguridad de la aplicación.

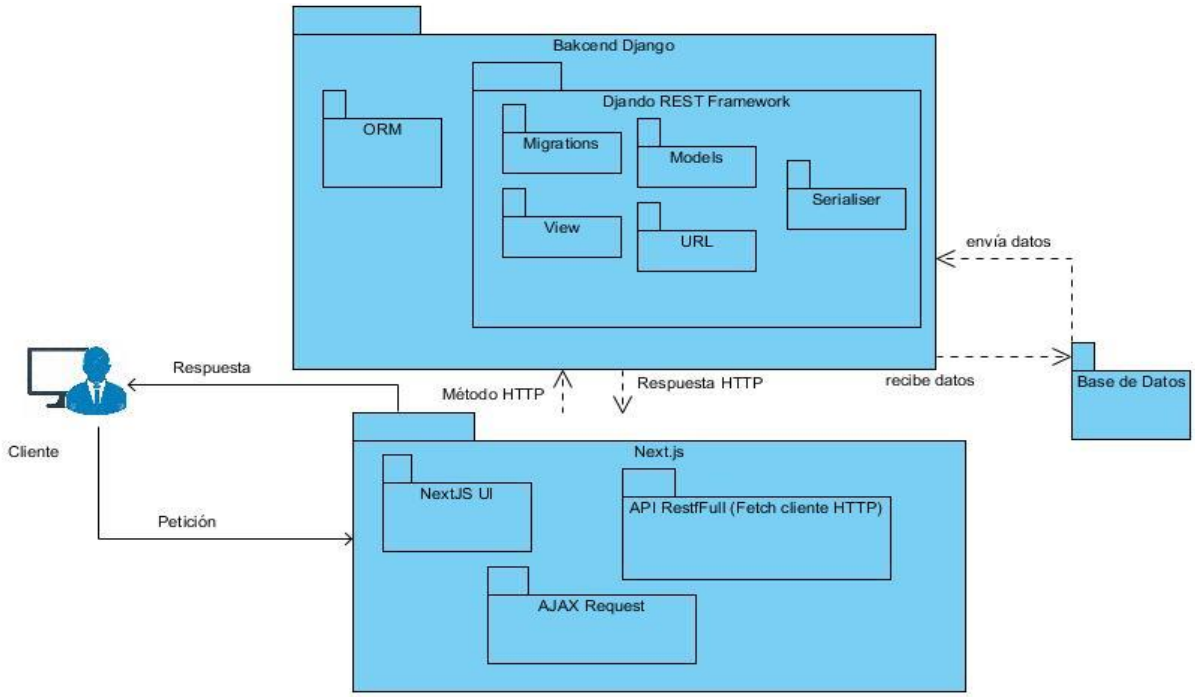


Figura 5 Diseño de la Arquitectura.

2.7 Patrones de Diseño

Los patrones de diseño son soluciones comunes y repetitivas a problemas de diseño de software que han sido identificadas y documentadas por expertos en la materia. Estos patrones proporcionan soluciones probadas y eficaces para problemas que se presentan con frecuencia en el desarrollo de software.

Estos pueden ser considerados como un conjunto de mejores prácticas y principios que se han desarrollado a lo largo del tiempo para resolver problemas específicos en el diseño de software. Estos patrones pueden ser aplicados en diferentes situaciones para mejorar la calidad del software y reducir el tiempo y costo de desarrollo.

2.7.1 Patrones Grasp

Los patrones GRASP (General Responsibility Assignment Software Patterns) son un conjunto de patrones de diseño orientados a objetos que se utilizan para asignar responsabilidades y tareas a las clases y objetos en un sistema de software. Los patrones GRASP se centran en la asignación de responsabilidades como un medio para lograr una buena arquitectura de software. (Improving Software Quality by Using GRASP and GoF Design Patterns., 2013)

- ✓ **Creador**: se utiliza para asignar la responsabilidad de crear objetos a una clase específica. En el desarrollo de este sistema se utiliza en los **serializadores**.

```python
@extend_schema_serializer(component_name="SerializadorDeUsuarioLecturaV1")
class SerializadorDeUsuarioLectura(SerializadorDeUsuarioLecturaBase):
    Jesús Enrique
    class Meta:
        model = SerializadorDeUsuarioLecturaBase.Meta.model
        fields = SerializadorDeUsuarioLecturaBase.Meta.fields + ["last_name"]
```

Figura 6 Patrón Creador.

- ✓ **Controlador**: es empleado para asignar la responsabilidad de controlar la lógica del sistema a una clase específica. Se utiliza en el desarrollo en las vistas.

```python
class VistasDeUsuarios(
    usecases.CreateUsuario,
    usecases.DetailUsuario,
    usecases.UpdateUsuario,
    usecases.DeleteUsuario,
    usecases.ListUsuario,
    ByOperationSerializer,
    ByVersionSerializer,
    viewsets.GenericViewSet,
):
    """Lee y actualiza los usuarios."""

    queryset = UsuarioManager().all()
    permission_classes = [permissions.IsAuthenticated, permissions.IsAdminUser]
    serializer_class = {
        "v1": {
            "read": serializers.v1.SerializadorDeUsuarioLecturaConPerfil,
            "write": serializers.v1.SerializadorDeUsuarioEscritura,
        },
        "v2": {
            "read": serializers.v2.SerializadorDeUsuarioLecturaConPerfil,
            "write": serializers.v2.SerializadorDeUsuarioEscritura,
        },
    }
```

Figura 7 Patrón Controlado.

- ✓ Experto en Información: se utiliza para asignar la responsabilidad de manejar la información a la clase que tiene la información necesaria para realizar la tarea. Para el desarrollo del módulo de configuración se utiliza en los modelos de datos.

```
class Usuario(auth.AbstractUser):
    """Representa un usuario."""
    email = models.EmailField( *args: _("email address"), unique=True)

    Jesús Enrique
    class Meta(auth.AbstractUser.Meta):
        verbose_name = "Usuario"
        verbose_name_plural = "Usuarios"
```

Figura 8 Patrón Experto.

Los patrones GOF son soluciones comunes y probadas a problemas de diseño de software que se presentan con frecuencia en el desarrollo de aplicaciones orientadas a objetos. El patrón utilizado es un patrón estructural, estos facilitan soluciones y estándares eficientes con respecto a las composiciones de clase y las estructuras de objetos. Específicamente el:

Decorator: Añade dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender la funcionalidad, restringe la alteración de la estructura del objeto mientras se le agrega una nueva funcionalidad.

El decorador utilizado fue:

@transaction.atomic: se utiliza para asegurar que un bloque de operaciones de base de datos se ejecute como una transacción atómica, garantiza que todas las operaciones dentro del bloque se traten como una sola.

```
@transaction.atomic
def update(self, instance, validated_data):
```

Figura 9 Patrón GOF, Decorador

2.8 Diagrama de clases del diseño

Los diagramas de clases de diseño (DCD) exponen un conjunto de interfaces, colaboraciones y sus relaciones. Se utilizan para modelar la vista de diseño estática de un sistema y forman parte de las realizaciones de las historias de usuarios. Además, son de gran importancia, pues permiten visualizar, especificar y documentar modelos estructurales. Con la definición de los principales aspectos a tener en cuenta para la

realización del modelo de diseño, se establece una estructura de paquetes dividida en fragmentos manejables para su posterior implementación.

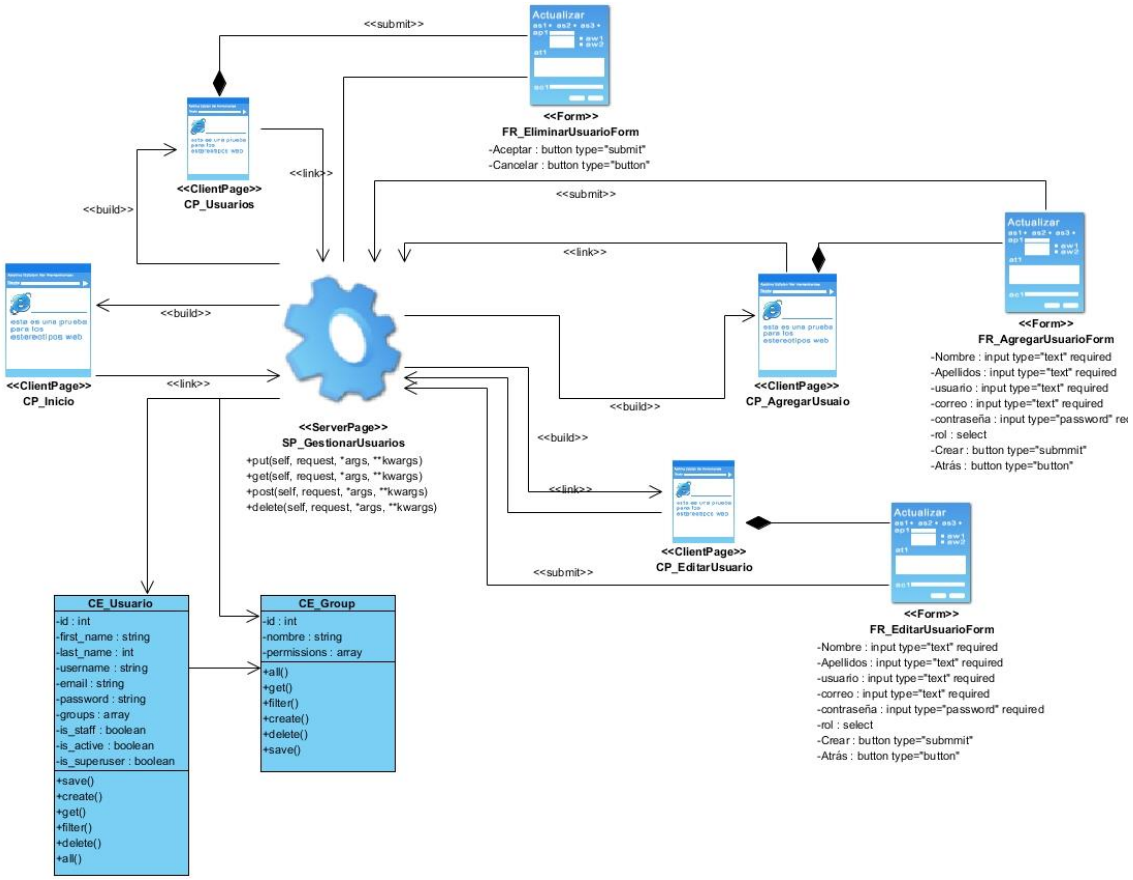


Figura 10 Diagrama de clases<Gestionar Usuarios>.

2.9 Diseño de la Base de Datos

El diseño de una base de datos consiste en definir la estructura de los datos que debe tener la base de datos de un sistema de información determinado. En el caso relacional, esta estructura será un conjunto de esquemas de relación con sus atributos, dominios de atributos, claves primarias, claves foráneas, entre otros.

El modelo entidad-relación (MER) como se muestra en la ilustración, es un modelo de datos que permite representar cualquier abstracción, percepción y conocimiento en un sistema de información formado por un conjunto de objetos denominados entidades y relaciones

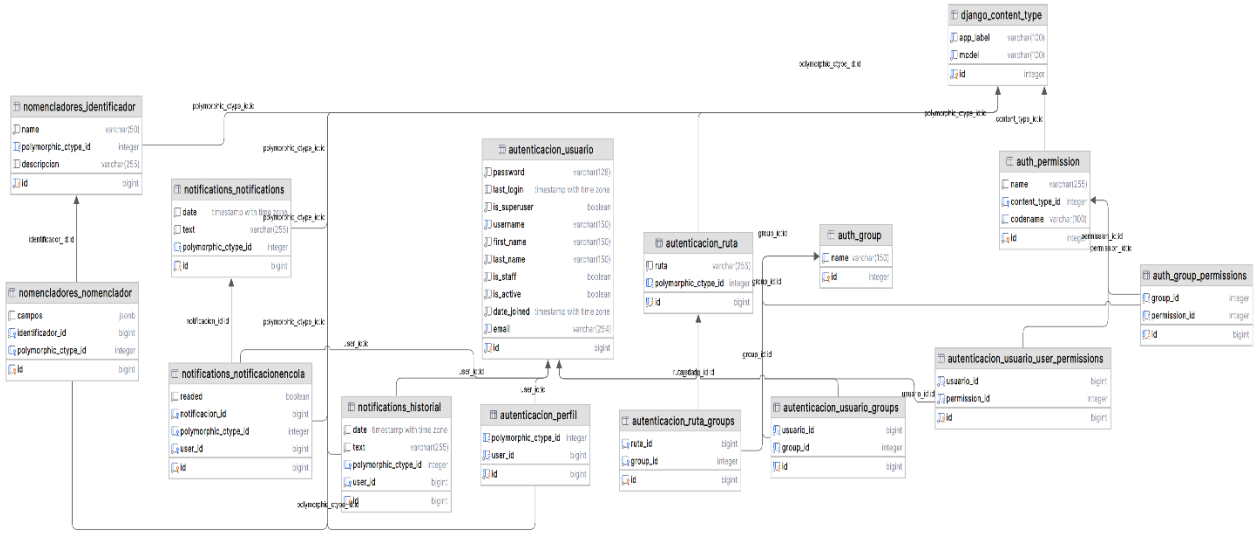


Figura 11 Diseño de la Base de Datos. Fuente: DataGrip.

2.9.1Descripción de las tablas de la base de datos

En esta sección se presenta una breve descripción de cada uno de los atributos de la tabla “Usuarios”.

Tabla 6 Descripción de la tabla Usuarios

Descripción: En esta tabla se agrupa la información correspondiente a los datos comunes de todos los Usuarios		
Atributo	**Tipo**	**Descripción**
Id	integer	Etiqueta única que identifica el objeto.
first_name	Varchar	Guarda la nombre del usuario.
last_name	Varchar	Guarda los apellidos del usuario.
Email	Varchar	Guarda el correo del usuario.
Password	Varchar	Guarda la contraseña del usuario.
is_staff	Boolean	Permite al usuario acceder al sitio administrativo de Django.
is_superuser	Boolean	Asigna permisos al usuario sin necesidad de otorgarlo.
is_active	Boolean	Guarda si el usuario está activo.

Conclusiones Parciales

La especificación de requisitos para el módulo a implementar se basó en 26 requisitos funcionales y 14 no funcionales, lo que proporcionó una comprensión clara de los elementos que debían ser incorporados. Además, los artefactos generados durante la etapa de análisis y diseño utilizando la metodología AUP en su variante UCI, establecieron las bases esenciales para la implementación del módulo, definiendo detalladamente las relaciones entre las entidades del sistema.

La investigación y definición de la arquitectura de software y los patrones de diseño contribuyeron a la implementación, que se centró en el diseño previamente establecido. Asimismo, el diseño de la base de datos definió la estructura del sistema, incluyendo esquemas de relación, atributos, dominios de atributos, claves primarias, claves foráneas, entre otros elementos.

Capítulo 3: Implementación y Pruebas

Establecido el alcance del módulo, definida la arquitectura y concretados los patrones a utilizar, quedan creadas las condiciones para iniciar el proceso de implementación y pruebas al módulo. Este capítulo está enmarcado en la fase de implementación del sistema; además, se muestra el diagrama de componentes, así como los estándares de codificación que se deben seguir para generar el código fuente. Se describen y realizan las pruebas de software al módulo implementado haciendo uso de los diseños de casos de prueba.

3.1 Diagrama de Componentes

Los diagramas de componentes se utilizan para modelar la vista estática de un sistema. Muestran la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema

A continuación, la ilustración muestra el diagrama de componentes del CU gestionar Usuarios:

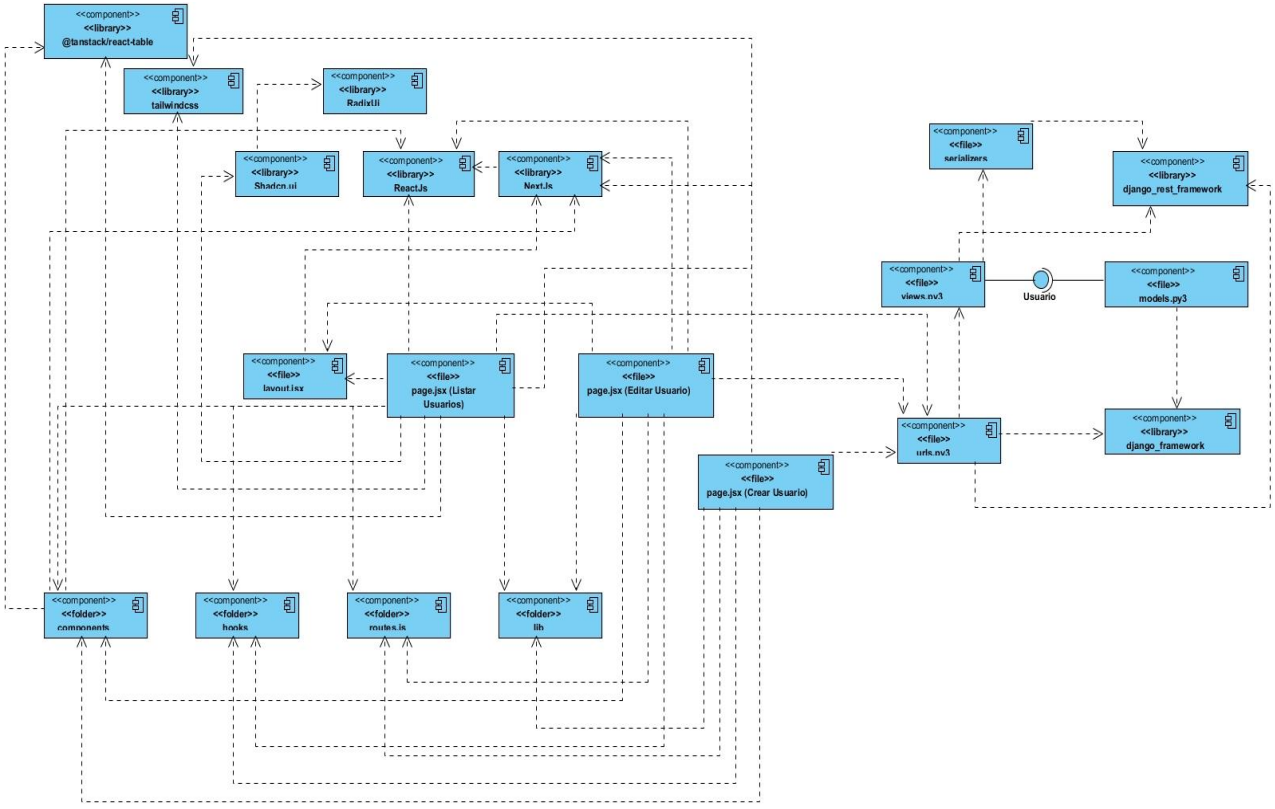


Figura 12 Diagrama de Componentes <Gestionar Usuarios>.

3.2. Diagrama de Despliegue.

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos.

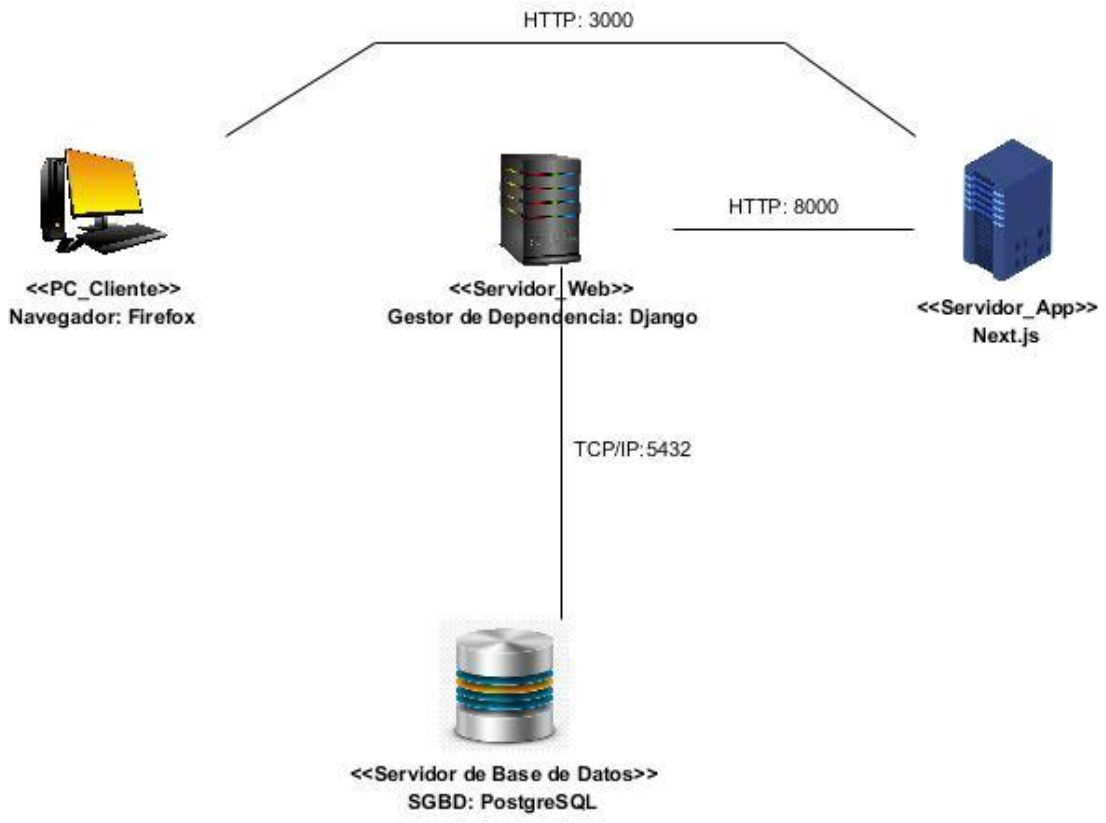


Figura 13 Diagrama de Despliegue

3.3. Implementación

La etapa de implementación del desarrollo de software es el proceso de convertir una especificación del sistema en un sistema ejecutable. Los procesos de diseño y programación de software se encuentran implicados en esta etapa; si se utiliza un enfoque evolutivo de desarrollo, también puede incluir un refinamiento de la especificación del software.

3.3.1. Estándares de codificación

Los estándares de código, son parte de las llamadas buenas prácticas. Estas son un conjunto no formal de reglas, que han ido surgiendo en las distintas comunidades de desarrolladores con el paso del tiempo y las cuales, bien aplicadas pueden incrementar notablemente, la calidad del código. (Arthur 2020). El estándar de codificación para el desarrollo de la herramienta definido luego de un estudio realizado previamente es el CamelCase. Se decidió el uso de este estándar debido a que reduce el esfuerzo

necesario para leer y entender el código fuete, además de que mejora la apariencia de este al no permitir abreviaturas. (Keefe 2022)

CamelCase es un estilo de escritura que se aplica a frases o palabras compuestas. El nombre se debe a que las mayúsculas a lo largo de una palabra se asemejan a las jorobas de un camello cuando la primera letra de cada una de las palabras es mayúscula. (Keefe 2022).

Existen dos tipos de CamelCase el UpperCamelCase, cuando la primera letra de cada una de las palabras es mayúscula y el lowerCamelCase, igual que la anterior con la excepción de que la primera letra es minúscula. A continuación, algunas pautas utilizadas del estándar anteriormente mencionado.

- Los nombres de las clases serán en mayúscula, en caso de ser un nombre compuesto las siguientes palabras se escribirán de igual forma.
- Los nombres de los métodos serán con mayúscula, en caso de ser un nombre compuesto, la primera palabra será en minúscula y la siguiente en mayúscula.
- Los identificadores para las variables y los parámetros serán en minúsculas.
- Los nombres de variables o funciones deben ser lo suficientemente descriptivos no más de 15 caracteres.
- De haber comentarios serán en idioma español para una mejor comprensión.

Tabla 7 Estándares de Codificación.

Descripción	Ejemplo
CamelCase	
UpperCamelCase, cuando la primera letra de cada una de las palabras es mayúscula	export default function UsuarioForm ({ data = undefined }) {
lowerCamelCase, la primera letra es minúscula.	const [selected, setSelected] = useState (data ? data.modelo.nombre : ” ”);

3.4. Pruebas de software

Las pruebas de software son un conjunto de actividades que implican ejecutar una implementación del software con datos de prueba. Se examinan las salidas del

software y su entorno operacional para comprobar que funciona tal y como se requiere. (CB, 2004) Tienen como objetivo descubrir y corregir el máximo de errores posible antes de su entrega al cliente, asegurando así el correcto cumplimiento de las funcionalidades del producto.

Siguiendo los pasos de la metodología que guía el proceso de desarrollo de software en la presente investigación, así como las características del módulo y el sistema a quien tributa el mismo, las pruebas que se utilizaron se llevan a cabo en tres niveles: Pruebas Unitarias y Pruebas de Aceptación y Pruebas de Integración. A continuación, se describen estas disciplinas a través de los principales elementos y los resultados obtenidos en cada una de estas. (RS, 2007)

3.5 Niveles de Pruebas

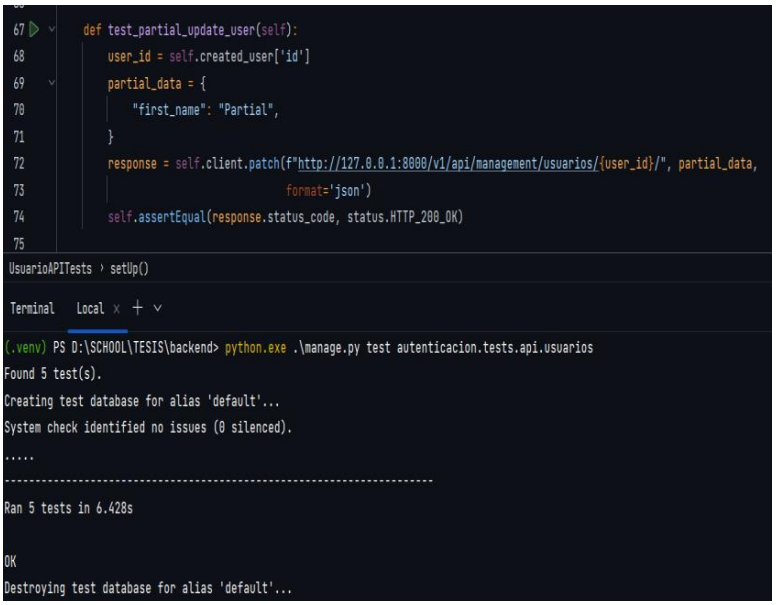
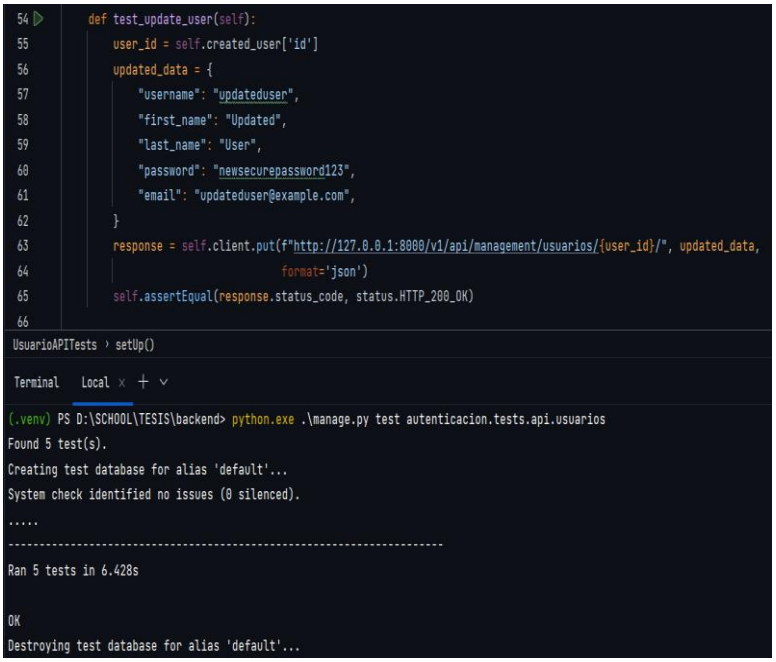
Tabla 8 Niveles de Prueba

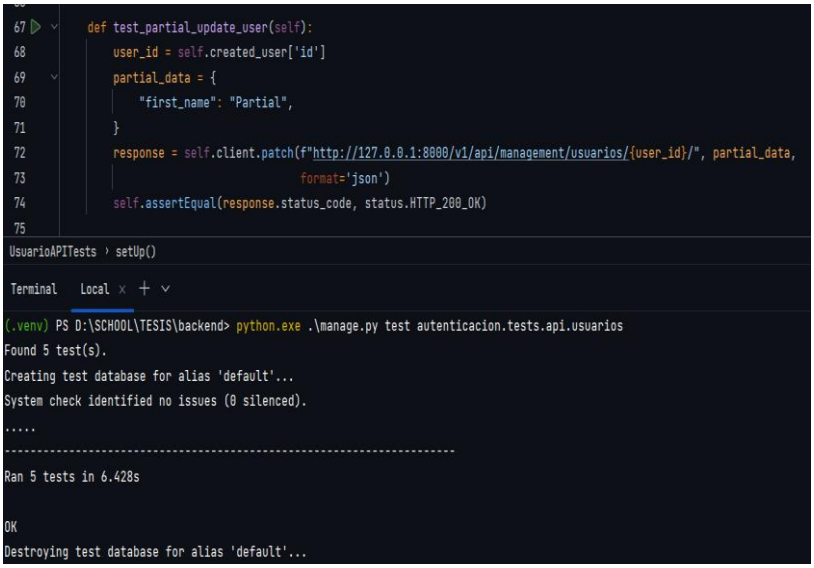
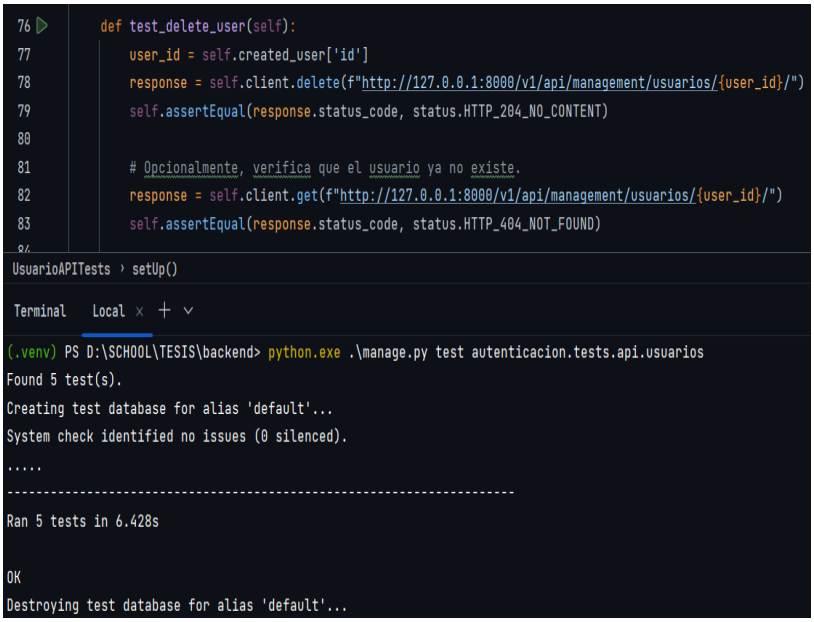
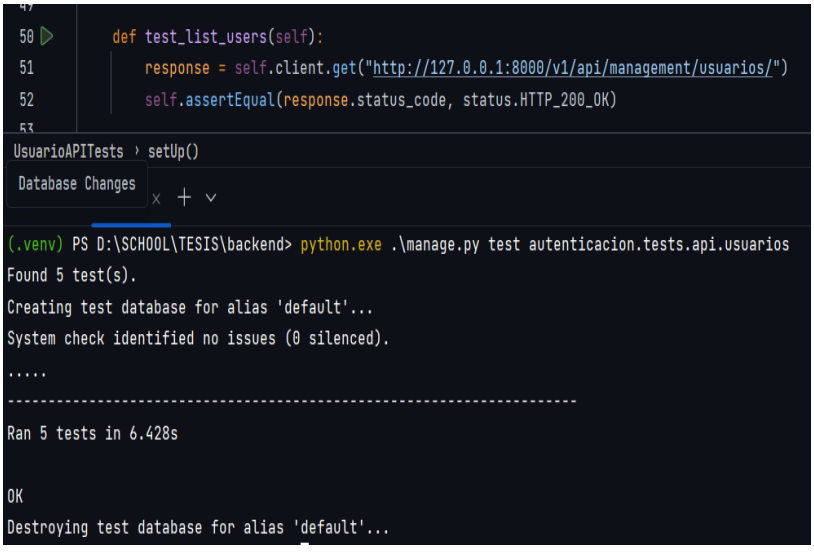
Etapa	Nivel de Prueba	Objetivo
Construcción	Prueba de Sistema, Unitarias	Las pruebas unitarias consisten en comprobar que un método concreto del código de producción funciona correctamente.
Despliegue	Prueba de Integración	Descubrir que el software que se ha creado es lo que el cliente y los usuarios esperaban.
Despliegue	Prueba de Aceptación	Esta prueba tiene como fin validar si el software satisface las necesidades del usuario final.

3.5.1. Pruebas Unitarias

Las pruebas unitarias se enfocan en examinar minuciosamente partes individuales del software, como funciones o métodos, para verificar que funcionen de manera correcta y aislada. El propósito principal es identificar y corregir posibles errores en las partes más pequeñas antes de que se conviertan en problemas más grandes en el sistema en su conjunto.

Tabla 9 Pruebas Unitarias.

Prueba	Resultado dado por Unittest	Resultado
Crear Usuario en base de datos exitosamente con sus datos validados		Se crea un usuario exitosamente
Editar Usuario en base de datos previamente creado con sus datos validados		Se edita un Usuario exitosamente

Editar Usuario parcialmente en base de datos previamente creado con sus datos validados		Se edita parcialmente un Usuario exitosamente.
Eliminar Usuario en base de datos previamente creado con sus datos validados		Se elimina un Usuario exitosamente.
Listar Usuarios en base de datos previamente creados con sus datos validados		Se listan los Usuario exitosamente

3.5.2. Pruebas de Aceptación

Las pruebas de aceptación se centran en observar cómo el sistema completo se comporta desde el punto de vista del usuario final. Estas pruebas se basan en escenarios y flujos de trabajo completos, y su objetivo principal es asegurarse de que el software cumple con los requisitos y las expectativas del cliente.

Tabla 10 Prueba de aceptación <Gestionar Usuarios>

Caso de Prueba de Aceptación
Número de Historia de Usuario: 1
Nombre: Gestionar Usuarios
Descripción: Ofrece al usuario una vista que permite agregar, eliminar, editar y listar los usuarios con su nombre, apellidos, correo y contraseña.
Condiciones de Ejecución: • El usuario ha iniciado sesión en el sistema. • El usuario tiene permisos para agregar, eliminar, editar y listar Usuarios.
Entrada/Pasos de Ejecución: • **Paso 1:** Acceder a la pantalla del listado de usuarios 1. Agregar un nuevo usuario. 2. En la pantalla de listado de usuarios, hacer clic en el botón "Agregar". 3. Completando los siguientes campos: nombre, apellidos, correo y contraseña. 4. Hacer clic en el botón "Crear" para confirmar la adición del Usuario y verificar la adición del mismo. • **Paso 2:** Editar un usuario existente. 1. En la vista de gestión de Usuarios, identificar uno de la lista que se desea editar. 2. Hacer clic en el botón "Editar" junto al Usuario seleccionado. 3. Modificar los detalles del Usuario, como nombre, apellidos, correo y contraseña. 4. Hacer clic en el botón "Editar" para confirmar los cambios. • **Paso 3:** Eliminar un Usuario existente. 1. En la vista de gestión de Usuarios, identificar uno de la lista que se desea eliminar.

2. Hacer clic en el botón "Eliminar" junto al Usuario seleccionado. 3. Confirmar la eliminación en el cuadro de diálogo de confirmación. • **Paso 4:** Listar los Usuarios existentes. 1. Verificar que la lista de Usuarios se muestra correctamente en la vista de gestión de estos.
Resultado Esperado: El Usuario se agrega, elimina, edita y lista correctamente al sistema y los datos se muestren correctamente en la lista de Usuarios.
Evaluación de la Prueba: Satisfactoria

Tabla 11 Prueba de aceptación <Gestionar Roles>.

Caso de Prueba de Aceptación
Número de Historia de Usuario: 2
Nombre: Gestionar Roles
Descripción: Ofrece al usuario una vista que permite agregar, eliminar, editar y listar los roles con su nombre y permisos.
Condiciones de Ejecución: • El usuario ha iniciado sesión en el sistema. • El usuario tiene permisos para agregar, eliminar, editar y listar Roles.
Entrada/Pasos de Ejecución: • **Paso 1:** Acceder a la pantalla del listado de Roles 1. Agregar un nuevo rol. 2. En la pantalla de listado de roles, hacer clic en el botón "Agregar". 3. Completando los siguientes campos: nombre y permisos. 4. Hacer clic en el botón "Crear" para confirmar la adición del Rol y verificar la adición del mismo. • **Paso 2:** Editar un Rol existente. 1. En la vista de gestión de Roles, identificar uno de la lista que se desea editar. 2. Hacer clic en el botón "Editar" junto al Rol seleccionado. 3. Modificar los detalles del Rol, como nombre y permisos. 4. Hacer clic en el botón "Guardar" para confirmar los cambios.

• **Paso 3:** Eliminar un Rol existente. 1. En la vista de gestión de Roles, identificar uno de la lista que se desea eliminar. 2. Hacer clic en el botón "Eliminar" junto al Rol seleccionado. 3. Confirmar la eliminación en el cuadro de diálogo de confirmación. • **Paso 4:** Listar los Roles existentes. 1. Verificar que la lista de Roles se muestra correctamente en la vista de gestión de estos.
Resultado Esperado: El Rol se agrega, elimina, edita y lista correctamente al sistema y los datos se muestren correctamente en la lista de Roles.
Evaluación de la Prueba: Satisfactoria

Aval de aceptación del cliente

3.5.3. Pautas a seguir para la Integración

La integración de módulos en sistemas informáticos es un proceso esencial que garantiza una interacción armónica entre componentes independientes, conformando así un sistema cohesivo y funcional. En el marco del módulo inicial o módulo de configuración, es fundamental sentar las bases para la futura integración de módulos subsiguientes. Para asegurar una transición fluida y una integración efectiva de los módulos venideros, es primordial adherirse a una serie de directrices específicas. A continuación, se presentan estas pautas estratégicas, ofrecidas como recomendaciones del autor:

- **Definición de Roles:** Es esencial definir y establecer roles claros, acompañados de sus permisos asociados. Estos roles deben corresponder al módulo específico que se esté integrando. La claridad en esta definición garantiza que las funcionalidades se ejecuten con los niveles de acceso adecuados.
- **Establecimiento de Rutas:** Se debe proceder a configurar las rutas específicas para cada módulo. Estas rutas comprenden un conjunto de direcciones internas asociadas a cada funcionalidad del módulo. Es vital comprender que cada ruta se asocia a roles específicos. Si bien un usuario puede tener asignados ciertos roles, si las rutas no están correctamente vinculadas a dichos roles, este usuario, a pesar de tener los permisos, no podrá acceder a las interfaces pertinentes en el Frontend.
- **Creación de Permisos Adicionales:** Aunque el framework utilizado pueda proporcionar un conjunto predefinido de permisos para operaciones a nivel de

servidor, pueden surgir situaciones en las que se requiera la definición de permisos adicionales. Estos permisos personalizados pueden ser necesarios para garantizar niveles de acceso o funcionalidades específicas no cubiertas por el framework.

- **Asignación de Roles a Usuarios:** Cada usuario del sistema debe tener un rol asignado. Estos roles determinan los permisos y niveles de acceso del usuario dentro del sistema. La omisión de esta asignación resultaría en un usuario sin capacidad de interactuar o ejecutar acciones dentro del sistema.

Adherirse a estas pautas es fundamental para garantizar una integración de módulos eficaz, segura y sin contratiempos en el sistema. La precisión y meticulosidad en cada paso contribuyen a la construcción de un sistema robusto y eficiente.

Conclusiones parciales

Los estándares de codificación definidos contribuyeron a obtener un estilo de programación homogéneo de manera tal que los participantes pudieran interpretar de manera eficiente la implementación de la propuesta de solución. Conjuntamente, la realización de las pruebas de software, tanto las unitarias como las de aceptación, permitieron obtener un software que cumple con las especificaciones del cliente y redujeron la posibilidad de errores durante su ejecución. Se identificaron varias no conformidades que se resolvieron de manera satisfactoria, logrando que la aplicación se ajuste a las expectativas del cliente.

Conclusiones Generales

La investigación culminó con la creación exitosa del módulo de Gestión de la Configuración para el Sistema de Gestión de Información de la DGT, el cual brindará un apoyo significativo a los procesos internos de la entidad. Los aspectos más destacados del módulo incluyen:

1. El estudio del estado del arte facilitó la recopilación de información y tendencias sobre sistemas de Gestión de la Configuración en términos generales. Esto permitió una comprensión más profunda del alcance y los objetivos de la investigación.
2. El análisis detallado de herramientas y tecnologías resultó esencial para definir y desarrollar la propuesta de solución para el módulo. Este análisis posibilitó la selección del entorno de desarrollo más adecuado para su implementación.
3. La integración de diversas áreas, incluyendo la programación, la ingeniería de software y las bases de datos, fue crucial para el análisis, diseño e implementación eficaces del módulo de Gestión de la Configuración.
4. La validación de las estrategias de prueba establecidas confirmó el funcionamiento adecuado del módulo. Esta validación tomó en cuenta los requisitos definidos junto con el cliente y demostró que el módulo desarrollado cumplió con el objetivo principal de la investigación.

Recomendaciones

Tomando como base la investigación realizada y el análisis de los resultados obtenidos se recomienda a la Dirección de Gestión Tecnológica (DGT):

1. Incorporar al sistema técnicas de inteligencia artificial (IA) que permita analizar las grandes cantidades de datos que generan las trazas del sistema con el objetivo de detectar patrones y tendencias, lo que facilitará la toma de decisiones basada en datos.
2. Definir una política de respaldo de la información a corto, mediano y largo plazo y automatizar la misma.
3. Incluir a la aplicación técnicas de seguridad basadas en el aprendizaje automático para detectar y prevenir amenazas cibernéticas de manera proactiva así como la encriptación de datos.

Bibliografía

(W3C), World Wide Web Consortium. 2014. HTML 5.2. *https://www.w3.org/TR/html52/.* [Online] 2014.

Alegsa, Leandro. 2023. *https://www.alegsa.com.ar/Dic/requerimiento_de_software.php#gsc.tab=0.* [Online] junio 19, 2023.

Ambler, S. 2013. The Agile Unified Process (AUP). Ambysoft. *https://www.ambysoft.com/unifiedprocess/agileUP.html.* [Online] 2013.

Arismendiz, Cruz Delia Cabada. 2017. Arquitectura Cliente - Servidor . *https://www.academia.edu/35294361/Arquitectura_Cliente_Servidor.* [Online] 2017.

Arthorne, John. 2007. https://wiki.eclipse.org/Eclipse_Documentation#:~:text=Eclipse%20Documentation%20This%20page%20is%20a%20hub%20for,dev.eclipse.org%3A%2Fcvsroot%2Feclipse%20CVS%20repository%3A%20Workbench%20User%20Guide%20-%20org.eclipse.platform.doc.user. [Online] abril 25, 2007.

Bhatia, Sagar and Robert , Johns. 2023. PostgreSQL vs MySQL | ¿Qué RDBMS es mejor? *https://hackr.io/blog/postgresql-vs-mysql.* [Online] 2023.

BUCHANAN, IAN. Gestión de configuración. *Atlassian.* [Online] https://www.atlassian.com/es/microservices/microservices-architecture/configuration-management.

Carnes, Beau. 2021. FreeCodeCamp - An Introduction to HTML:. *https://www.freecodecamp.org/news/html-crash-course/.* [Online] agosto 5, 2021.

Castillo, Celia Clemente. 2008. Python. [Online] julio 27, 2008. http://www.it.uc3m.es/spickin/docencia/comsoft/presentations/spanish/doc/Python.pdf.

CB, Reynoso. 2004. *Introducción a la Arquitectura de Software.* Argentina: BuenosAires : s.n., 2004.

CM, Pedro. 2009. *https://www.cssblog.es/ventajas-e-inconvenientes-al-usar-css/.* [Online] 2009.

CODE. 2023. https://code.visualstudio.com/docs/editor/debugging. [Online] 2023.

—. 2023. https://code.visualstudio.com/docs/languages/php. [Online] 2023.

django. 2023. *https://docs.djangoproject.com/en/4.2/#top.* [Online] 2023.

Duque, Raúl González. 2019. *Python para todos.* Madrid : Creative Commons , 2019.

Farro, Jose. 2016. *SISTEMA INTEGRADO DE GESTIÓN ADMINISTRATIVA.* Perú : s.n., 2016.

Figueroa, R G Solís, C J Coelho, F. 2012. *Metodologías tradicionales VS. Metodologías .* 2012.

Gitnux. 2023. https://blog.gitnux.com/es/metodologias-de-desarrollo-de-software/. [Online] 2023.

—. 2023. Metodologías de desarrollo de software: características, tipos, ventajas y desventajas. [Online] 2023.

Gutierrez, Margarita Elizabeth Dorado. 2020. *Estudio comparativo de la implementación de una aplicación web o de escritorio para el control del personal de la empresa AMESEPREVE en la ciudad de Babahoyo.* Babahoyo : s.n., 2020.

Hernández Claro R, Greuguas Navarro D. 2010.. *Estándares de Diseño Web. s.l. : Ciencias de la Información,.* 2010.

Impacto Económico y Social de los Servicios Técnicos de la Dirección de Gestión Tecnológica. **Rodríguez , Lester, Mateu Romero, Yamila and Castaño, José A. 2021.** 2021.

Improving Software Quality by Using GRASP and GoF Design Patterns. **Oviedo-García M. A., García-Sánchez F.,Colomo-Palacios. 2013.** 2013, Journal of Universal Computer Science.

infostudio. 2011. Principales Características del Lenguaje Python. [Online] agosto 31, 2011.

Ito, Yoshimi. 2018. *Modular Design for Machine Tools.* 2018.

J, Eguíluz Pérez. 2009.. *Introducción a JavaScript . [En línea] www.librosweb.es.* 2009.

Kruchten, P. 2004. *The Rational Unified Process: An Introduction. Addison-Wesley Professional.* 2004.

Larman, C. 2004. *Agile and Iterative Development: A Manager's Guide. Addison-Wesley Professional.* 2004.

Letelier, Patricio. noviembre 15, 2006.. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).* noviembre 15, 2006.

Lianet Cabrera González, Enrique Roberto Pompa Torres. 2012.. *Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información. .* La Habana : ISSN | RNPS, 2012., 2012.

MARTIN FOWLER, KENDALL SCOTT. 2003.. *UML gota a gota. .* Mexico : Editorial Mexicana, 2003.

MDN. https://developer.mozilla.org/es/docs/Learn/CSS/First_steps/Why_use_CSS. [Online]

MÉNDEZ, A. 2013. *Proceso de desarrollo de software DAC. Universidad de las Ciencias Informáticas: Departamento de Desarrollo de la Dirección de Informatización.* La Lisa : s.n., 2013.

Mendoza, Marvin López. 2020. *https://openwebinars.net/blog/que-es-un-lenguaje-de-programacion/.* [Online] 2020.

Ministerio de Economía y Finanzas. 2021. *Módulo de Configuración.* Perú : s.n., 2021.

Montoro, Arturo Fernández. 2012. *Python al descubierto.* Madrid : RC Libros, 2012.

MUÑOA, DANIELLA PÉREZ. 2022. Agencia Cubana de Noticias. *Crea XETID sistema informático para la gestión empresarial.* [Online] 2022. [Cited: Mayo 19, 2023.] http://www.acn.cu/medio-ambiente/92320-crea-xetid-sistema-informatico-para-la-gestion-empresarial.

NEXT.js. 2023. *https://nextjs.org/docs.* [Online] 2023.

Obeo. (s.f.). UML Designer. Obeo. . *https://www.obeodesigner.com/en/product-overview.* [Online] (s.f.).

Pressman, Roger. 2015. *Ingeniería del Software: Un enfoque práctico.* 2015.

2019. R&C Consulting. *QUÉ ES EL SIGA? (SISTEMA INTEGRADO DE GESTIÓN ADMINISTRATIVA).* [Online] 2019. [Cited: Mayo 19, 2023.] https://rc-consulting.org/blog/2019/06/que-es-el-siga/.

REST. 2011. https://www.django-rest-framework.org/. [Online] 2011.

Reyes, Antonio Jesús Matos. 2022. Radio Rebelde. *Cuba cuenta con un software para la gestión empresarial.* [Online] 2022. [Cited: Mayo 19, 2023.]

https://www.radiorebelde.cu/noticia/cuba-cuenta-con-un-software-para-la-gestion-empresarial-20220324/.

Riquelme, Matias. 2018. https://www.webyempresas.com/las-tic-en-la-administracion/. [Online] 2018.

Rodríguez, T. 2015. *Actualización de los roles de la metodología.* 2015.

Rodríguez, T. (2014).. *Metodología de desarrollo para la Actividad productiva de la UCI. pages 1–16.* (2014).

RS, Pressman. 2007. *Ingeniería de software: Un enfoque práctico. .* Nueva York, EUA : s.n., 2007.

Salvaggio, Alessandra. 2019. https://zoboko.com/book/431yw299/javascript-guia-completa#:~:text=Si%20quieres%20crear%20codigo%20compatible%20con%20la%20mayor,version%20de%20JavaScript%20mas%20utilizada%20en%20la%20actualidad. [Online] 2019.

Santa Fe Provincia. 2022. Santa Fe Provincia. *Educación- Sistema de gestión escolar web.* [Online] 2022. [Cited: Mayo 19, 2023.] http://www.santafe.gov.ar/index.php/educacion/guia/get_tree_by_node?node_id=122273.

SIGAE WEB. 2018. *TUTORIAL DE SIGAE WEB.* Argentina : s.n., 2018.

SIGAEWEB. 2013. *Manual de usuario.* Argentina : s.n., 2013.

Sommerville, Ian. 2011. *Ingeniería de Software.* 2011.

Spiegato. 2023. ¿Qué es un modelo de dominio? *https://spiegato.com/es/que-es-un-modelo-de-dominio.* [Online] 2023.

tailwindcss. 2020. *https://tailwindcss.com/.* [Online] 2020.

Thelin, Ryan. 2021. A continuación.js tutorial con ejemplos: Cree mejores aplicaciones de React con Next. *https://www.educative.io/blog/nextjs-tutorial-examples.* [Online] 2021.

VALLE, JOSE GUILLERMO. 2018. Definición arquitectura cliente servidor. *https://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.* [Online] 2018.

Villar, Miguel Ángel. 2023. Historias de usuario: qué son, cómo escribirlas, plantilla y ejemplos. *https://profile.es/blog/historias-de-usuario/.* [Online] febrero 13, 2023.

W3Schools. - HTML Introduction:. *https://www.w3schools.com/html/html_intro.asp.* [Online]

XETID. 2023. XETID. *Gestiona y controla fácilemente tu empresa.* [Online] 2023. [Cited: Mayo 19, 2023.] https://www.xetid.cu/es/productos/distra.

ZAPROO. 2022. ZAPROO. *Enhancing User Experience: Customization vs Personalization.* [Online] 2022. [Cited: Mayo 19, 2023.] https://www.zaproo.com/articles/user-experience-customization-vs-personalization.

Anexos

Historias de Usuarios

Tabla 12 Historia de Usuario<Gestionar Identificadores>

Historia de Usuario	
Número: 3	**Nombre de HU:** Gestionar Identificadores
Programador: Aarom Cárdenas Hernández	**Iteración Asignada:** 1era
Prioridad: Alta	**Tiempo Estimado:** 3 días
Riesgo en Desarrollo: Alto	**Tiempo Real:** 2 días
Descripción:	

1. **Objetivo:**
   - Ofrece al usuario una vista que permite agregar, eliminar, editar y listar los identificadores con su nombre y descripción.
2. **Flujo de la acción a realizar:**
   - El sistema debe permitir crear, eliminar, editar y listar el/los identificadores con su nombre y descripción.
3. **Flujo alterno**
   - El sistema debe permitir ir “Atrás” o “Cancelar” en caso de que la opción seleccionada no sea la deseada.

Prototipo de Interfaz:

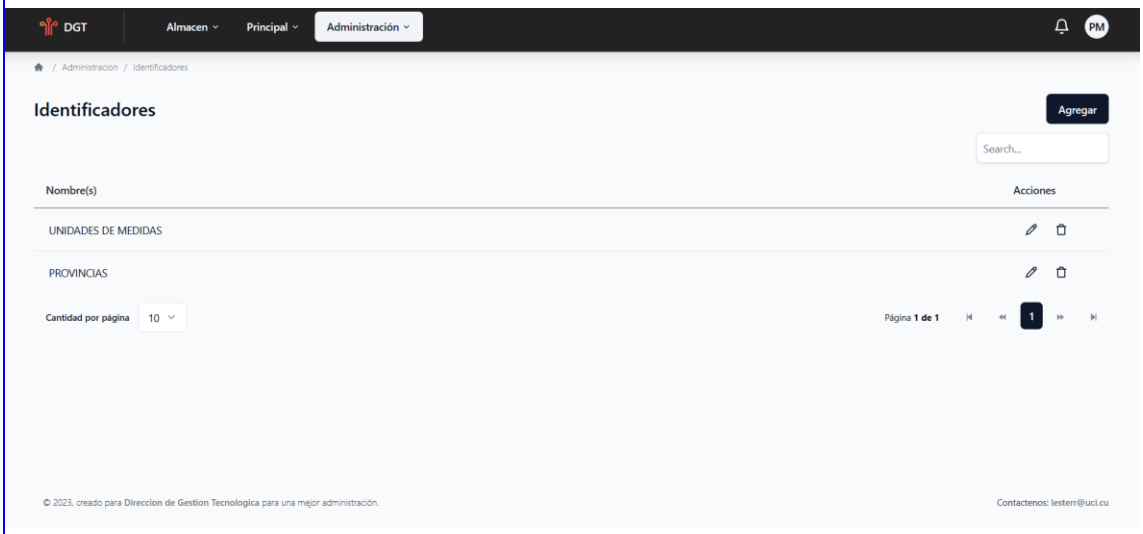


Tabla 13 Historia de Usuario<Gestionar Nomencladores>

Historia de Usuario	
Número: 4	**Nombre de HU:** Gestionar Nomencladores
Programador: Aarom Cárdenas Hernández	**Iteración Asignada:** 1era
Prioridad: Alta	**Tiempo Estimado:** 7 días
Riesgo en Desarrollo: Alto	**Tiempo Real:** 5 días
Descripción:	
1. **Objetivo:** • Ofrece al usuario una vista que permite agregar, eliminar, editar y listar los nomencladores con su nombre, descripción e identificador. 2. **Flujo de la acción a realizar:** • El sistema debe permitir crear, eliminar, editar y listar el/los nomencladores con su nombre, descripción e identificador. **3. Flujo alterno** • El sistema debe permitir ir “Atrás” o “Cancelar” en caso de que la opción seleccionada no sea la deseada.	
Prototipo de Interfaz:	

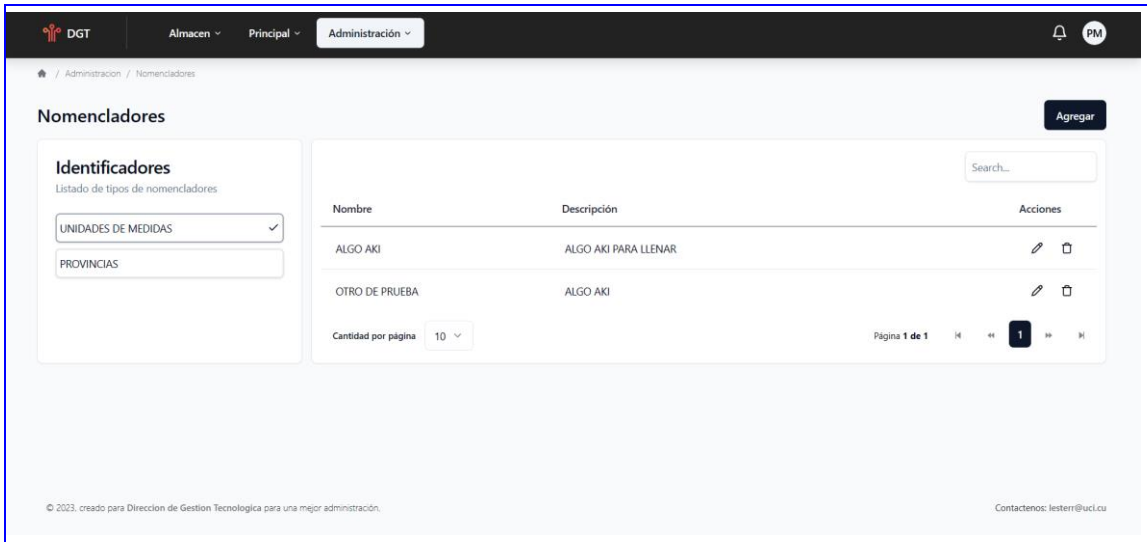

DGT
Almacen
Principal
Administración
PM
/ Administracion / Nomencladores
Nomencladores
Agregar
Identificadores
Listado de tipos de nomencladores
UNIDADES DE MEDIDAS
PROVINCIAS
Search...
Nombre
Descripción
Acciones
ALGO AKI
ALGO AKI PARA LLENAR
OTRO DE PRUEBA
ALGO AKI
Cantidad por página
10
Página 1 de 1
1
© 2023, creado para Direccion de Gestion Tecnologica para una mejor administración.
Contactenos: lesterr@uci.cu

Tabla 14 Historia de Usuario<Gestionar Rutas>

Historia de Usuario	
Número: 5	**Nombre de HU:** Gestionar Rutas
Programador: Aarom Cárdenas Hernández	**Iteración Asignada:** 1era
Prioridad: Alta	**Tiempo Estimado:** 4 días
Riesgo en Desarrollo: Alto	**Tiempo Real:** 3 días

Descripción:

1. **Objetivo:**
   - Ofrece al usuario una vista que permite agregar, eliminar, editar y listar las rutas con su nombre y un listado de roles asociados.
2. **Flujo de la acción a realizar:**
   - El sistema debe permitir crear, eliminar, editar y listar la/las rutas con su nombre y un listado de roles asociados
3. **Flujo alterno**
   - El sistema debe permitir ir “Atrás” o “Cancelar” en caso de que la opción seleccionada no sea la deseada.

Prototipo de Interfaz:

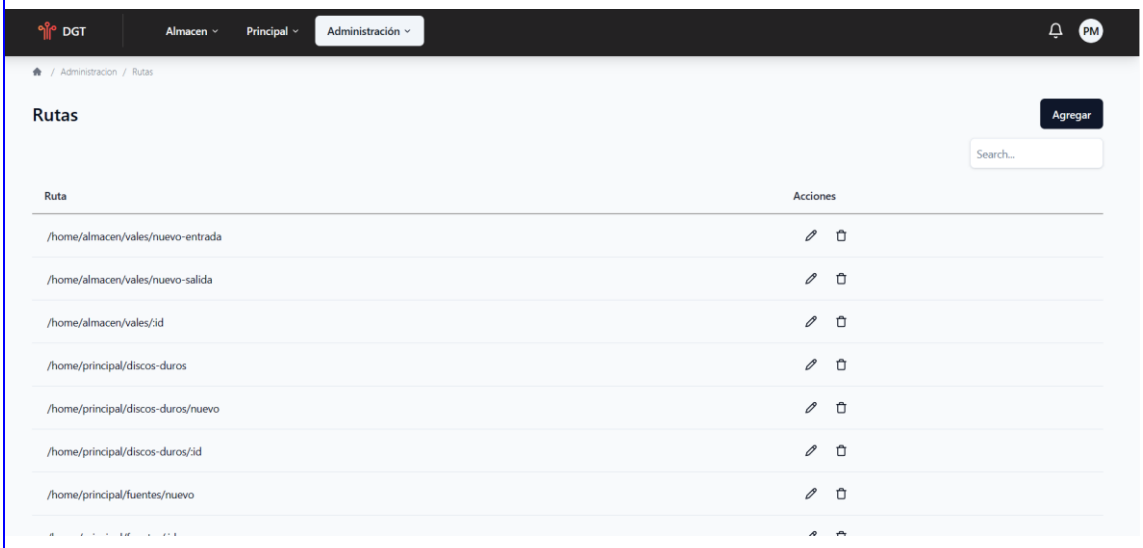
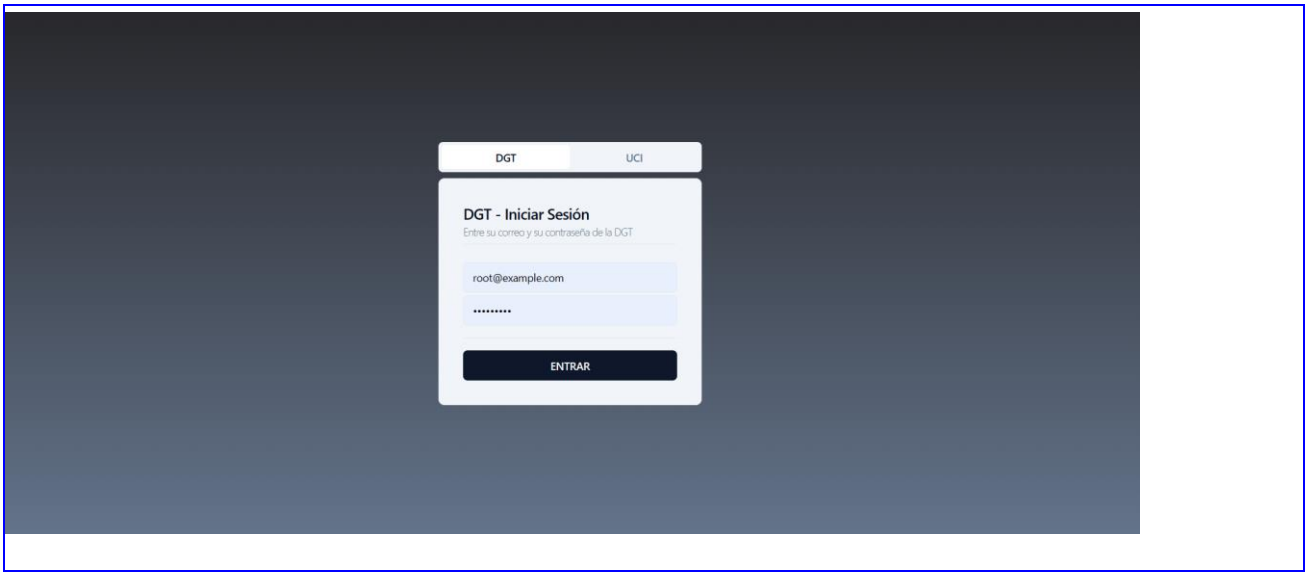


Tabla 15 Historia de Usuario<Autenticar Usuario>

Historia de Usuario	
Número: 5	**Nombre de HU:** Autenticar Usuario
Programador: Aarom Cárdenas Hernández	**Iteración Asignada:** 1era
Prioridad: Alta	**Tiempo Estimado:** 5 días
Riesgo en Desarrollo: Alto	**Tiempo Real:** 4 días
Descripción:	
1. **Objetivo:** • Ofrece al usuario una vista que permite inicial sesión con su correo o usuario y contraseña. 2. **Flujo de la acción a realizar:** • El sistema debe permitir autenticar al usuario con su correo o usuario y contraseña. **3. Flujo alterno** • Cuando se escriba un usuario o una contraseña incorrecta el sistema deba mostrar una pantalla con un error	
Prototipo de Interfaz:	



Diagramas de Clase

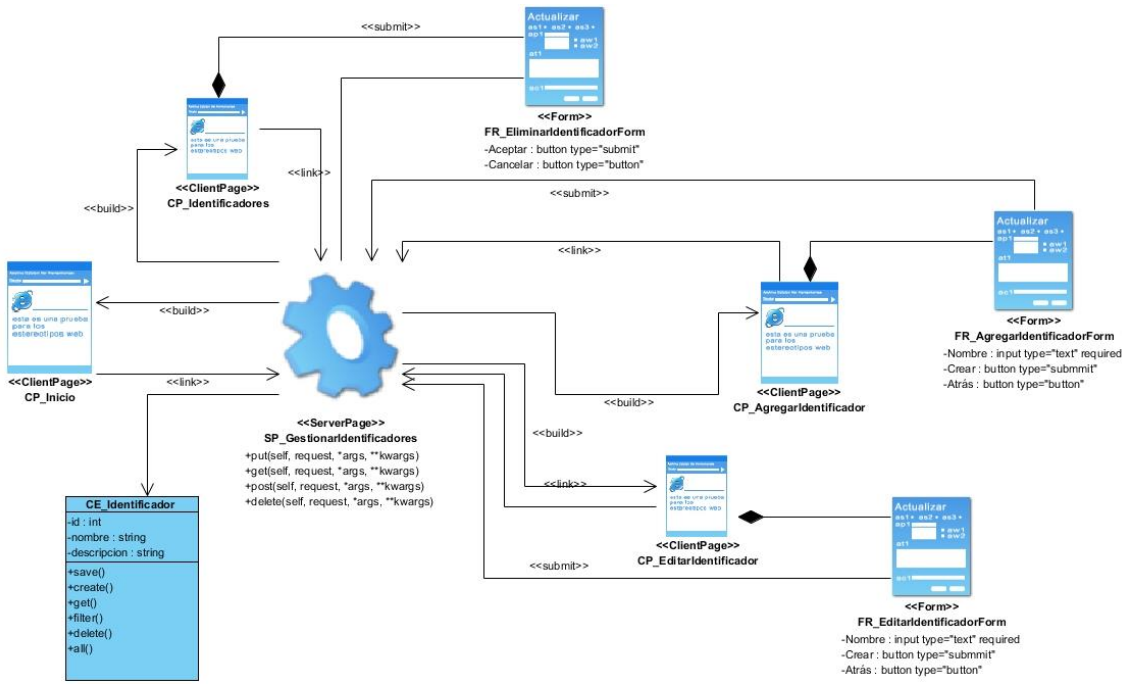


Figura 14 Diagrama de clases<Gestionar Identificadores>.

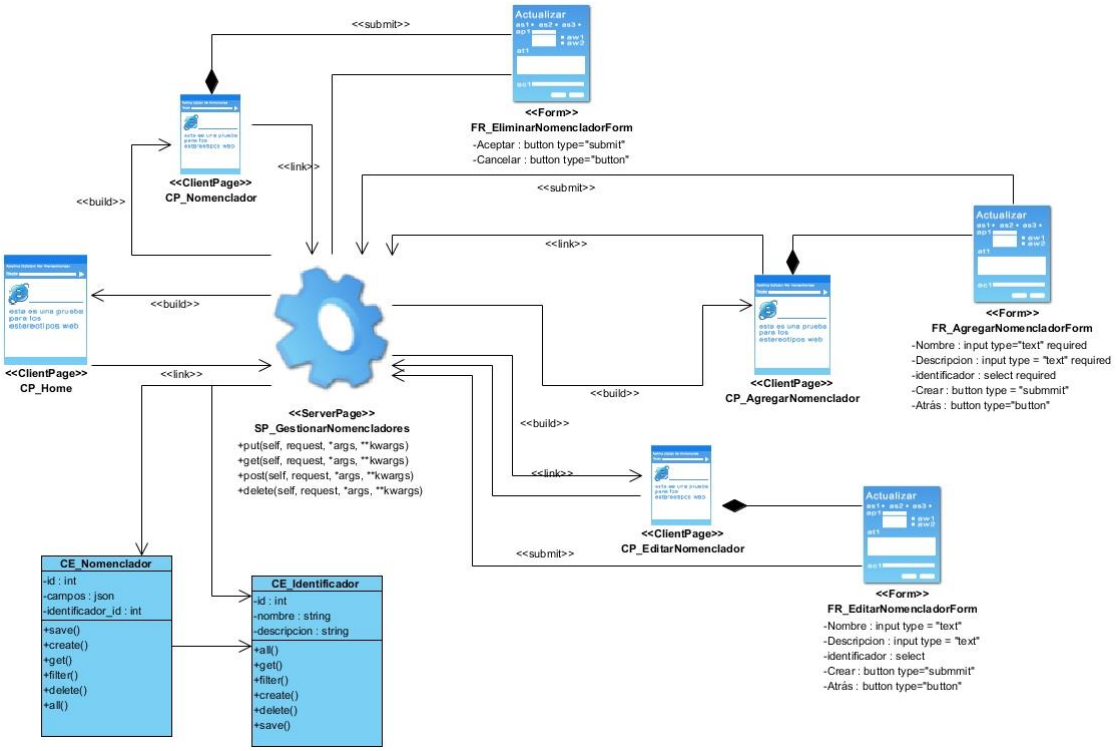


Figura 15 Diagrama de clases<Gestionar Nomencladores>

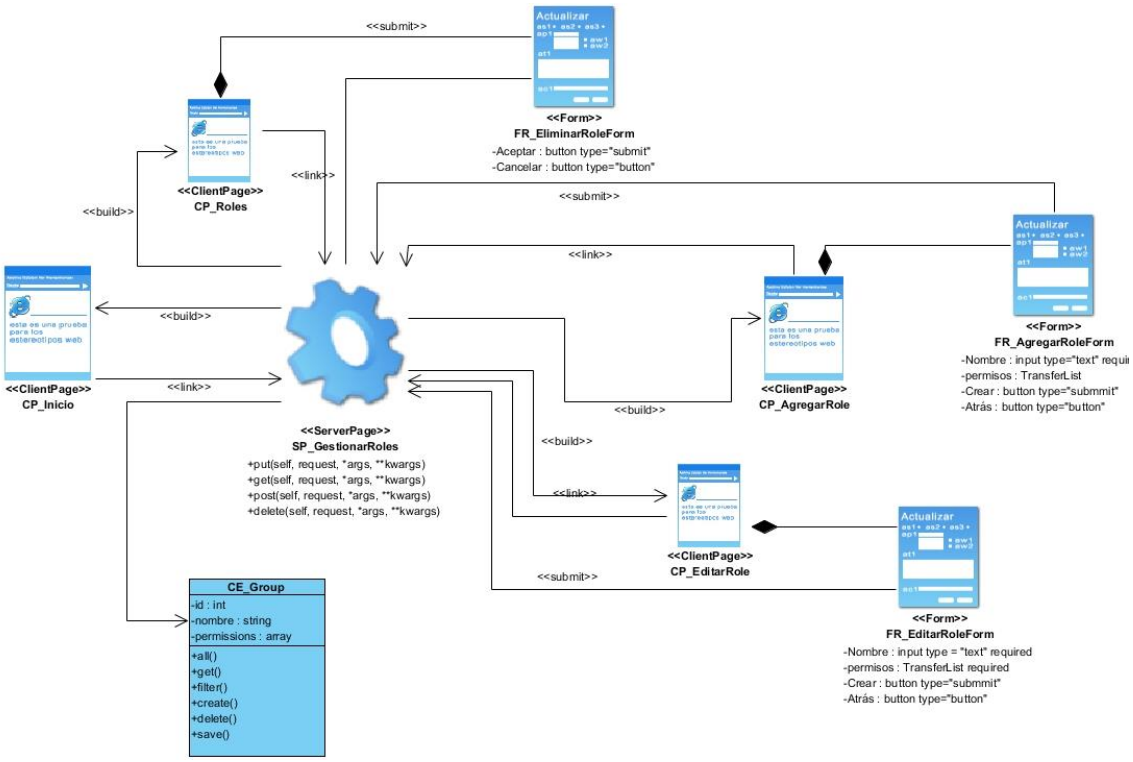


Figura 16 Diagrama de clases<Gestionar Roles>.

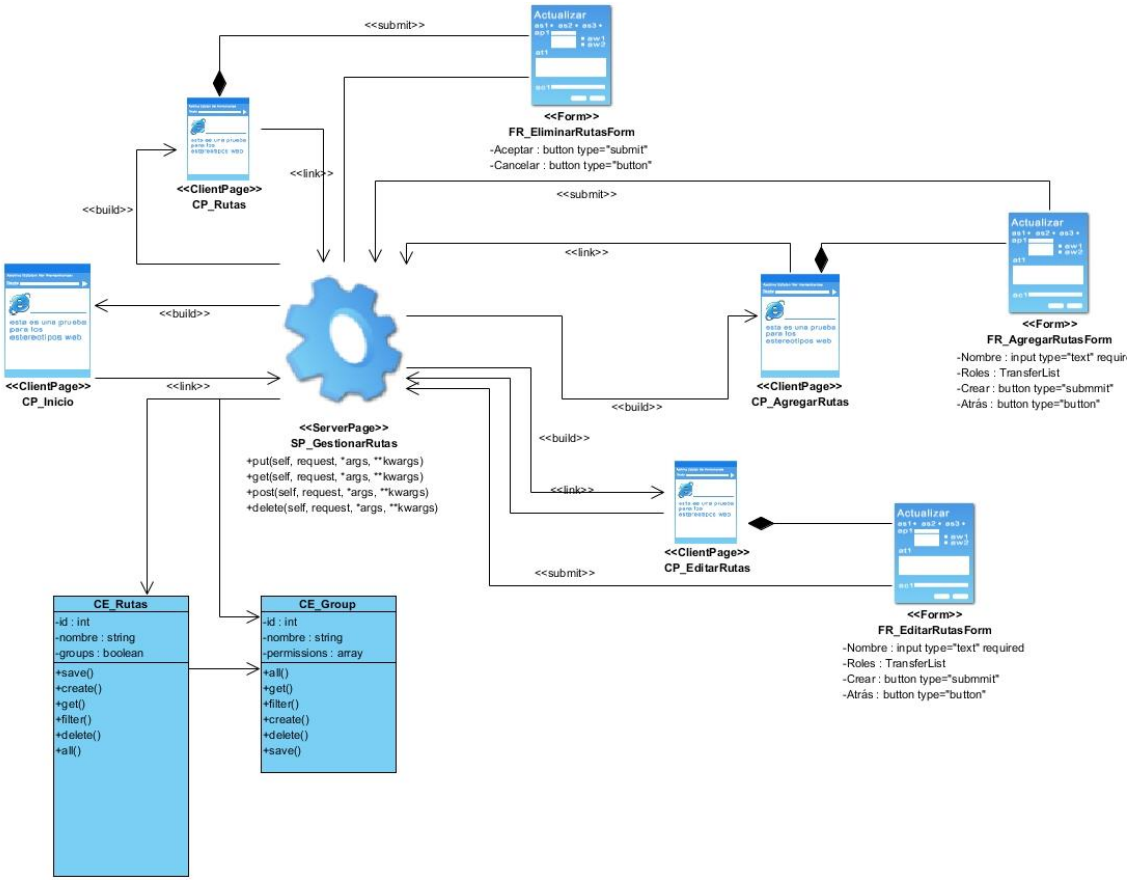


Figura 17 Diagrama de clases<Gestionar Rutas>.

Aval de aceptación del cliente

UCI Universidad de las Ciencias Informáticas | DIRECCIÓN DE GESTIÓN DE TECNOLOGÍA

La Habana, 24 de noviembre de 2023
"Año 65 de la Revolución"

ACTA DE ACEPTACIÓN

Por medio de la presente, la Dirección de Gestión Tecnológica (DGT), en nombre de la Universidad de las Ciencias Informáticas, emite esta acta de aceptación ya que está de acuerdo con el trabajo realizado y pondrá en uso en el menor tiempo posible el módulo desarrollado por el estudiante Aarom Ramsés Cárdenas Hernández.

En la Dirección de Gestión Tecnológica se desarrolla un Sistema de Gestión de Información Tecnológica, concebido de manera modular, teniendo en cuenta las áreas en las que está estructurada la dirección, para informatizar los procesos y procedimientos que se ejecutan en cada una de ellas. Se hace necesario que este sistema cuente con un núcleo que es la base para que exista un correcto funcionamiento en las operaciones que se realizan entre los módulos que lo conforman, lo cual garantiza la integración entre dichos módulos y la gestión de elementos comunes como son: la seguridad y los nomencladores.

El módulo Gestión de la configuración para el Sistema de Gestión de Información de la DGT desarrollado permite la gestión de usuarios, de roles, de notificaciones, de nomencladores, y establece la arquitectura base sobre la cual funciona todo el sistema de Gestión de la DGT.

El módulo en cuestión tiene una interfaz amena, amigable, da respuesta de las acciones que se ejecutan en un tiempo prudencial, ayuda a eliminar un grupo de problemas relacionados con la pérdida de la información, permite trazabilidad en las actividades que se ejecutan.

Y PARA QUE CONSTE la aceptación del trabajo realizado, se firma la presente ACTA:

Lester Rodríguez Vallejo	José David Rodríguez Vargas	Javier Arias Sojo
Director	Especialista Superior de control Interno	Especialista Superior