



Universidad de las Ciencias  
Informáticas

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

**Título:** Módulo de Gestión de partes y piezas para el Local de Piezas Recuperadas de la Dirección de Gestión Tecnológica de la Universidad de las Ciencias Informáticas.

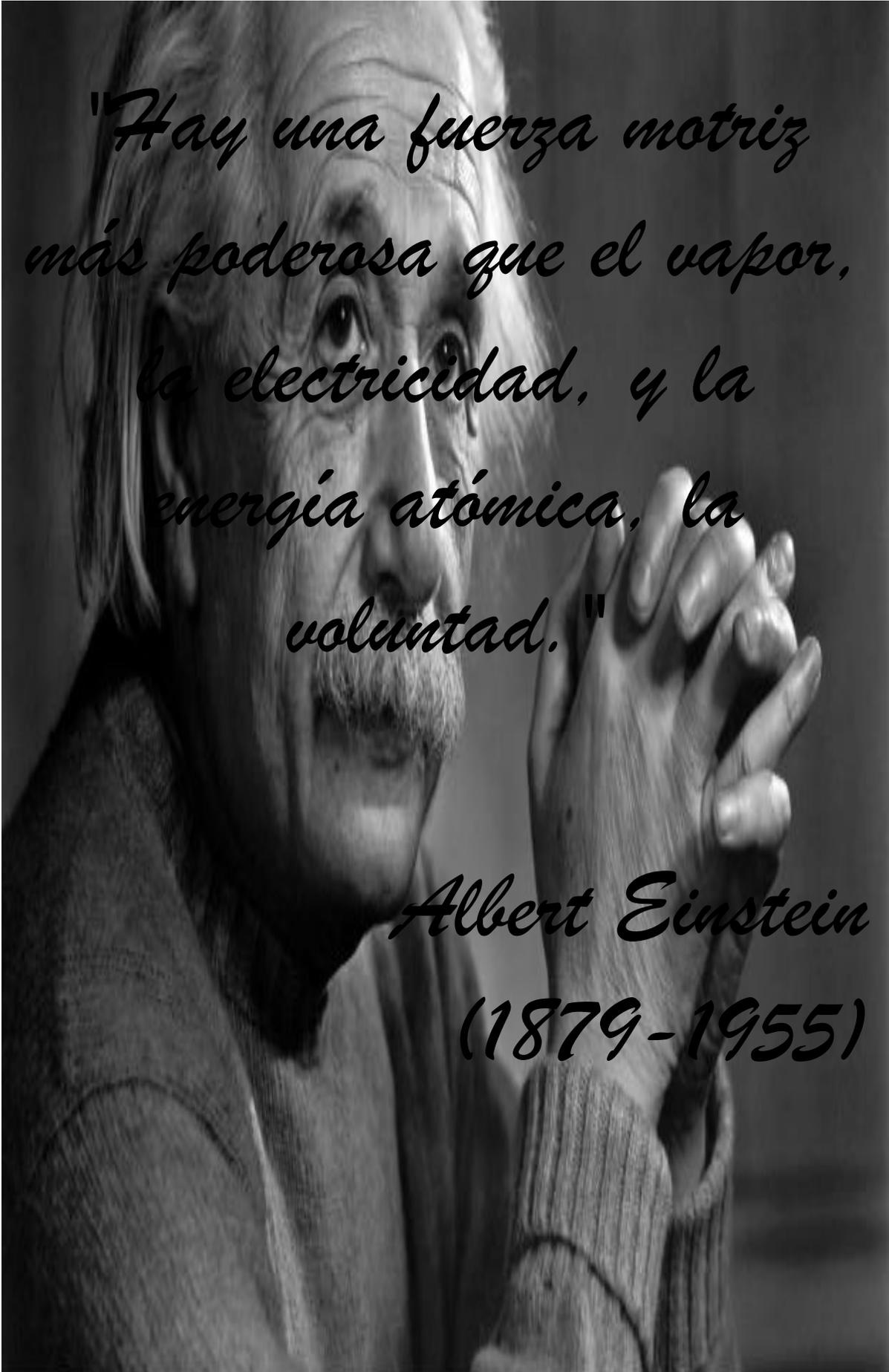
**Autora:** Beatriz Pérez Moreno

**Tutores:** Lic. Lester Rodríguez Vallejo

Lic. Angel Fabra Torres

La Habana, 27 de noviembre de 2023

“Año 64 de la Revolución”

A black and white portrait of Albert Einstein, showing his characteristic wild hair and mustache. He is looking slightly to the right of the camera with a thoughtful expression. His hands are clasped together in front of his chest. The background is dark and out of focus.

*"Hay una fuerza motriz  
más poderosa que el vapor,  
la electricidad, y la  
energía atómica, la  
voluntad."*

*Albert Einstein  
(1879-1955)*

## DECLARACIÓN DE AUTORÍA

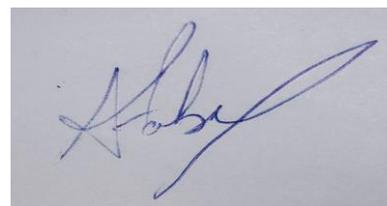
El autor del trabajo de diploma con título Módulo de Gestión de partes y piezas para el Local de Piezas Recuperadas de la Dirección de Gestión Tecnológica de la Universidad UCI concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma la presente a los 27 días del mes de noviembre del año 2023.

**Beatriz Pérez Moreno**



UCI  
DIRECCIÓN DE GESTIÓN  
TECNOLÓGICA  
Lester Rodríguez Vallejo  
Director

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**

Agradezco profundamente a mis padres por su amor incondicional, su dedicación y su comprensión en cada una de mis decisiones. Cada uno de mis éxitos está dedicado a ustedes, ya que han sido un modelo a seguir que me ha permitido convertirme en la persona que soy en la actualidad.

Mi sincero agradecimiento a mi hermana, Dunia, y a mi cuñado, José Agustín, por su constante presencia en mi vida. Su apoyo inquebrantable y su preocupación en cada una de las etapas que me ha tocado superar.

Así mismo, a mi sobrina, Karol Daniela, la niña de mis ojos, le agradezco de todo corazón por ser capaz de arrancarme una sonrisa en los momentos en que más lo he necesitado. Su alegría y cariño han sido un bálsamo para mi alma.

Quiero expresar mi profundo agradecimiento a mi esposo, Aarom, por ser mi compañero fiel en esta travesía de la vida, por estar en cada uno de los momentos de alegría y de dificultad a lo largo de todos estos años. Aprecio enormemente su capacidad para brindarme apoyo cuando más lo necesitaba, incluso cuando yo misma dudaba de mis fuerzas. Este logro que hoy celebro no habría sido posible sin tu incondicional apoyo, por lo que también es tuyo.

A milena, que, aunque hoy no puede estar aquí se lo orgullosa que se siente de mí, desde el minuto cero vi en ti una amiga, otra hermana, una mujer fuerte y valiente, gracias por siempre estar, por apoyarme, por las risas, por los regaños, por las noches de estudio, por las noches de dominó, por todos los momentos juntas, por ser la mamá de los pollitos, yo también me siento muy orgullosa de ti y de cada meta que logras y quiero que sepas que la distancia nunca nos separará.

En general a todos mis amigos, compañeros de carrera y profesores, que me han sabido aceptar y comprender con mis altos y bajos, pero en especial a Enrique, nuestro Steve, sabes que te queremos mucho, a Richard, José y Daniel, ojalá pudiera ver una última película con ustedes, a Nathalí y a Kevin por siempre estar ahí a Roxi, a Carlos y a Dayi. a todos los que de una forma u otra estuvieron siempre conmigo, los quiero mucho a todos

A mis tutores por su dedicación y esfuerzo para ayudarme a cumplir esta meta.

## **DEDICATORIA**

A mis padres quienes han sido mi más sólido apoyo en la búsqueda de mis sueños. Es gracias a su constante sacrificio y su inquebrantable fe en mí que hoy me encuentro en este lugar.

## **Resumen**

El adecuado control y gestión de inventarios son elementos críticos en cualquier entidad que requiera una administración eficiente de sus recursos. La Dirección de Gestión Tecnológica (DGT) como parte de su misión le corresponde ejecutar un correcto tratamiento del inventario de los recursos tecnológicos que se encuentran bajo su cuidado en el Local de Piezas Recuperadas (LPR). Para abordar este problema, se implementa un Sistema de Gestión de Inventario diseñado específicamente para ejecutar los procedimientos establecidos por la DGT. Este sistema permite el registro preciso y en tiempo real de las partes, piezas y herramientas disponibles en el LPR. Además, proporciona acceso en línea al inventario para que los técnicos y especialistas de la DGT puedan verificar la disponibilidad de manera remota. La propuesta moderniza y optimiza el proceso de entrega y devolución de partes, piezas y herramientas, mejorando la operatividad y disminuyendo el efecto de los riesgos asociados con los procedimientos que se ejecutan de manera manual. La solución se desarrolla utilizando Python con Django, siguiendo la metodología AUP UCI en el escenario número cuatro, se emplea la arquitectura Cliente-Servidor junto con patrones de diseño reconocidos (GOF y GRASP) para asegurar la robustez y solidez del sistema. Se aplican pruebas exhaustivas de funcionalidad para garantizar el funcionamiento correcto del sistema. Mediante este sistema, se logra una mejora en la entrada y salida de las partes técnicas al LPR, se reducen las molestias y pérdida de tiempo asociadas al proceso, y se satisfacen las necesidades de todas las partes involucradas.

**Palabras Clave:** Control, Entrada, Gestión, Salida, Sistema, Inventario

## **ABSTRACT**

Proper inventory control and management are critical elements in any entity that requires efficient resource management. The Directorate of Technology Management (DGT) faces the challenge of maintaining an inventory that manages the inflow and outflow of technical parts as part of its mission. To address this issue, the implementation of an Inventory Management System specifically designed for the needs of the DGT is proposed. This system will enable precise, real-time tracking of the parts, pieces, and tools available in the LPR (Local of Parts Recovery). Additionally, it will provide online access to the inventory, allowing DGT employees to check availability remotely. The proposal aims to modernize and optimize the process of delivering and returning parts, pieces, and tools, enhancing operational efficiency and reducing the risks associated with manual procedures. This solution will be developed using Python with Django, following the AUP UCI methodology in scenario 4. It will employ the Client-Server architecture along with recognized design patterns (GOF and GRASP) to ensure system robustness and solidity. Comprehensive functionality tests will be conducted to ensure the system operates correctly. Through this approach, it is expected to achieve improvements in the inflow and outflow of technical parts in the LPR, reduce the inconvenience and time loss associated with the process, and meet the needs of all involved parties.

**Keywords:** Control, Input, Management, Output, System, Inventory

## Índice

|  |    |
|--|----|
| Introducción .....   | 15 |
| Capítulo 1. Fundamentación Teórica. ....                           | 20 |
| 1.1. Introducción.....   | 20 |
| 1.2 Fundamentos teóricos asociados al tema.....                    | 20 |
| 1.3. Sistemas similares existentes.....                            | 21 |
| 1.3.1 Oracle Enterprise Asset Management (EAM):.....               | 22 |
| 1.3.2 SAP Asset Management: .....                                  | 23 |
| 1.3.3 IBM Maximo: .....  | 24 |
| 1.3.4 ASSETS .....   | 25 |
| 1.3.5 Análisis del estudio realizado a soluciones existentes ..... | 26 |
| Oracle Enterprise Asset Management (EAM) .....                     | 26 |
| SAP Asset Management.....  | 26 |
| IBM Maximo.....  | 26 |
| ASSETS .....   | 26 |
| 1.4. Metodología de Desarrollo de Software.....                    | 27 |
| 1.4.1 Metodología AUP-UCI.....                                     | 27 |
| 1.4.2 Descripción de las fases de AUP en su variación UCI .....    | 28 |
| 1.4.3 Escenarios en cuanto a Requisitos .....                      | 29 |
| 1.5. Lenguajes de programación. ....                               | 31 |
| 1.5.1. Python. ....  | 31 |
| 1.5.2 HTML.....  | 31 |
| 1.5.3 CSS (“Hojas de Estilo en Cascada”). Tailwind CSS .....       | 32 |
| 1.5.4 JavaScript.....  | 33 |
| 1.6. Marco de trabajo (Framework).....                             | 33 |
| 1.6.1 Django y Django REST .....                                   | 34 |
| 1.6.2 Next.js.....   | 34 |
| 1.7. Sistema gestor de Base de Datos.....                          | 35 |
| 1.7.1 PostgreSQL. ....   | 35 |

|   |    |
|---|----|
| 1.8 Otras Herramientas .....  | 36 |
| 1.8.1. Lenguaje de Modelado.....  | 36 |
| 1.8.2 Herramienta para el modelado. Visual Paradigm .....                 | 37 |
| 1.8.3 Entorno de Desarrollo Integrado. Visual Studio Code (VS Code).....  | 38 |
| Conclusiones Parciales.....   | 38 |
| Capítulo 2: Análisis y Diseño .....                                       | 40 |
| 2.1 Introducción.....   | 40 |
| 2.2 Propuesta de Solución .....   | 40 |
| 2.3 Modelo de Dominio .....   | 41 |
| 2.3.2 Definición de los conceptos del Diagrama de modelo de Dominio. .... | 41 |
| 2.4. Especificación de requisitos del software .....                      | 42 |
| 2.4.1 Requisitos Funcionales .....  | 42 |
| 2.4.2 Requisitos no funcionales .....                                     | 47 |
| 2.5 Historia de usuario .....   | 48 |
| 2.6 Diseño.....   | 50 |
| 2.6.1 Arquitectura de Software.....                                       | 51 |
| 2.7 Patrones de Diseño.....   | 53 |
| 2.7.1 Patrones Grasp.....   | 53 |
| 2.7.2 Patrones GOF.....   | 55 |
| 2.8 Diagrama de clases del diseño.....                                    | 56 |
| 2.9 Diseño de la Base de Datos .....                                      | 56 |
| 2.9.1 Descripción de las tablas de la base de datos .....                 | 57 |
| Conclusiones Parciales.....   | 57 |
| Capítulo 3: Implementación y Pruebas .....                                | 59 |
| 3.1 Diagrama de Componentes.....  | 59 |
| 3.2. Diagrama de Despliegue. ....   | 59 |
| 3.3. Implementación.....  | 60 |
| 3.3.1. Estándares de codificación .....                                   | 60 |
| 3.4. Pruebas de software.....   | 61 |

|                                       |    |
|---------------------------------------|----|
| 3.5 Niveles de Pruebas .....          | 62 |
| 3.5.1. Pruebas Unitarias .....        | 62 |
| 3.5.2. Pruebas de Integración .....   | 65 |
| 3.5.3. Pruebas de Aceptación .....    | 66 |
| 3.5.4 Pruebas de Carga y Estrés ..... | 69 |
| 3.6 Resultados del MGPPLPR .....      | 71 |
| Conclusiones parciales .....          | 73 |
| Conclusiones Generales.....           | 74 |
| Recomendaciones .....                 | 75 |
| Bibliografía.....                     | 76 |
| Anexos .....                          | 80 |
| Aval de aceptación del cliente .....  | 92 |

## Índice de Tablas

|  |    |
|--|----|
| Tabla 1 Características generales de las soluciones estudiadas. Fuente: Elaboración propia. ....             | 26 |
| Tabla 3 Descripción de las fases AUP-UCI. Fuente Elaboración propia.....                                     | 29 |
| Tabla 4: Requisitos Funcionales del Sistema. Fuente: Elaboración propia. ....                                | 47 |
| Tabla 5: Historia de Usuario<Gestionar Hdd>. Fuente: Elaboración propia. ....                                | 49 |
| Tabla 6: Historia de Usuario<Gestionar Fuentes>. Fuente: Elaboración propia. ....                            | 50 |
| Tabla 7: Historia de Usuario<Gestionar Producto>. Fuente: Elaboración propia.....                            | 50 |
| Tabla 8: Descripción de la tabla Activo. Fuente: Elaboración propia. ....                                    | 57 |
| Tabla 9 Estándares de Codificación. Fuente: Elaboración propia. ....   | 61 |
| Tabla 10 Niveles de Prueba. Fuente: Elaboración propia. ....   | 62 |
| Tabla 11 Pruebas Unitarias. Fuente: Elaboración propia. ....   | 64 |
| Tabla 12 Prueba de aceptación <Gestionar Hdd>. Fuente: Elaboración propia. ....                              | 67 |
| Tabla 13 Prueba de aceptación <Gestionar Fuente>. Fuente: Elaboración propia.....                            | 68 |
| Tabla 14 Prueba de aceptación <Gestionar Productos>. Fuente: Elaboración propia.                             | 69 |
| Tabla 15 Pruebas de Carga y Estrés. Fuente: Elaboración propia. ....   | 70 |
| Tabla 16 Historia de Usuario <Gestionar MotherBoard>. Fuente: Elaboración propia.<br>Anexo .....             | 85 |
| Tabla 17 Historia de Usuario <Gestionar Ram>. Fuente: Elaboración propia. Anexo .                            | 85 |
| Tabla 18 Historia de Usuario <Gestionar Procesador>. Fuente: Elaboración propia.<br>Anexo .....              | 86 |
| Tabla 19 Historia de Usuario <Gestionar Pieza sin Calificación >. Fuente: Elaboración<br>propia. Anexo ..... | 86 |
| Tabla 20 Historia de Usuario <Administrar Vales >. Fuente: Elaboración propia. Anexo<br>.....                | 87 |
| Tabla 21 Prueba de aceptación <Gestionar MotherBoard>. Fuente: Elaboración propia.<br>Anexo .....            | 88 |
| Tabla 22 Prueba de aceptación <Gestionar Ram>. Fuente: Elaboración propia. Anexo<br>.....                    | 89 |
| Tabla 23 Prueba de aceptación <Gestionar Procesador>. Fuente: Elaboración propia.<br>Anexo .....             | 90 |

Tabla 24 Prueba de aceptación <Gestionar Pieza sin Calificación>. Fuente: Elaboración propia. Anexo ..... 91

## Índice de Figuras

|   |    |
|---|----|
| Figura 1 Escenario 1 AUP-UCI .....  | 29 |
| Figura 2 Escenario 2 AUP-UCI .....  | 29 |
| Figura 3 Escenario 3 AUP-UCI .....  | 30 |
| Figura 4 Escenario 4 AUP-UCI .....  | 30 |
| Figura 5 Modelo de Dominio. Fuente: Elaboración Propia .....                    | 41 |
| Figura 6 Diseño de la Arquitectura. Fuente: Elaboración Propia .....            | 52 |
| Figura 7 Modelo de Base de Datos. Fuente: Elaboración Propia .....              | 57 |
| Figura 8 Diagrama de Componentes <Gestionar Hdd>. Fuente: Elaboración propia. . | 59 |
| Figura 9 Diagrama de Despliegue. Fuente: Elaboración propia. ....               | 60 |
| Figura 10 Resultado de las Pruebas Unitarias .....                              | 64 |
| Figura 11 Pruebas de Integración - GitHub .....                                 | 65 |
| Figura 12 Pruebas de Integración - GitHub .....                                 | 66 |
| Figura 13 Resultado de las Pruebas de Aceptación.....                           | 69 |
| Figura 14 Portada Inicial del Sistema.....                                      | 71 |
| Figura 15 Vista Agregar Hdd.....  | 71 |
| Figura 16 Vista Editar Hdd.....   | 72 |
| Figura 17 Vista general de los Vales.....                                       | 72 |
| Figura 18 Vista Productos.....  | 72 |
| Figura 19 Gestionar Ram. Anexo.....   | 80 |
| Figura 20 Gestionar Productos. Anexo .....                                      | 80 |
| Figura 21 Gestionar Procesador. Anexo .....                                     | 81 |
| Figura 22 Gestionar Pieza sin Calificación. Anexo .....                         | 82 |
| Figura 23 Gestionar MotherBoard. Anexo.....                                     | 83 |
| Figura 24 Gestionar Fuente. Anexo .....   | 83 |
| Figura 25 Administrar Vales. Anexo.....   | 84 |

## **Introducción**

El mundo de la informática se encuentra en evolución constante, provocando de forma considerable impactos positivos en diferentes esferas de la sociedad tales como el gobierno, las telecomunicaciones, el deporte, la educación y la salud, entre otras, al proveer herramientas, tecnologías y metodologías que facilitan, agilizan y optimizan su gestión. El uso de estos recursos potencia la automatización de varios procesos que antes se hacían manualmente, 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 de información intervienen muchos factores, siendo uno de los principales 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. Estos sistemas ofrecen múltiples ventajas en distintas áreas de la sociedad. Una de las áreas beneficiadas es la gestión de recursos tecnológicos, ya que, con la informatización de sus procesos, se tiene un mejor seguimiento de los recursos que se ofrecen y un mejor control de los mismos.

En Cuba existen instituciones y centros investigativos 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 Universidad de las Ciencias Informáticas (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 la misión, la universidad cuenta con una estructura organizativa dividida en varias áreas, incluyendo aquellas que proporcionan servicios cruciales para el funcionamiento interno de la institución. Una de estas áreas es la Dirección de Gestión Tecnológica (DGT), encargada de llevar a cabo diversos procesos, con especial énfasis en la gestión de incidencias tecnológicas y la eficiente administración de los recursos tecnológicos disponibles.

Siguiendo el procedimiento establecido en el Código PE-042-01 de Gestión de Incidencias por el Grupo de Asistencia Técnica (GAT), cuando un cliente informa de una incidencia a través del sistema en la dirección URL incidencias.uci.cu, el técnico responsable de atender la incidencia obtiene los datos necesarios del sistema o, en su caso, de los directivos de la DGT. A continuación, se desplaza al lugar donde se ha reportado la incidencia y solicita la presencia de los responsables correspondientes según el área afectada. El técnico ejecuta el servicio y, en caso de que sea necesario sustituir alguna parte o pieza informática, realiza una solicitud al jefe de grupo por correo electrónico.

Luego, el jefe de grupo verifica la disponibilidad de las partes y piezas en el Local de Piezas Recuperadas (LPR), de acuerdo con lo estipulado en el Procedimiento PE-024-07 de Autorización y Asignación de Partes y Piezas. Si hay disponibilidad, se emite una autorización por correo electrónico con la lista de lo solicitado. El técnico se dirige al LPR, donde se le entrega un vale con las piezas autorizadas. Una vez que ha reemplazado la parte o pieza defectuosa, el técnico regresa al LPR para devolver la pieza dañada, y se genera un vale de devolución.

Cuando se requieren partes o piezas informáticas del Almacén 9 de la UCI, el director debe consultar o verificar manualmente qué elementos están disponibles, ya que no existe un sistema digital que permita verificar la disponibilidad de manera automática. Cada vez que se introduce un nuevo insumo, parte o pieza en el LPR, se refleja el mismo en una tarjeta de Control para llevar un registro adecuado de las entradas. Este proceso resulta complicado, ya que para acceder a la disponibilidad en el LPR es necesario visitarlo personalmente y revisar de forma manual las tarjetas de control, lo que puede llevar tiempo considerable debido a la diversidad de partes y piezas existentes. En los casos en que no se pueda acceder al local, se produce un retraso en el proceso.

Además, todo el procedimiento se realiza de manera manual, lo que ralentiza significativamente la eficiencia del proceso

La situación descrita anteriormente trae consigo, además, los siguientes aspectos:

- 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. Esta información debe quedar almacenada al menos por cinco años por lo que en caso de sufrir deterioro la memoria histórica se ve afectada.
- El procesamiento manual de los datos implica una alta probabilidad de errores humanos durante la manipulación y registro de la información. La introducción errónea de datos repercute negativamente en la calidad y utilidad de la información recabada.
- Al depender exclusivamente de comunicaciones vía correo electrónico para la autorización de procesos, se genera una situación de vulnerabilidad ante interrupciones del servicio de correo. Esto provoca retrasos e ineficiencias en la ejecución de las actividades establecidas que requieren dichas autorizaciones. Es necesario implementar canales de comunicación y aprobación más confiables y robustos.

Teniendo en cuenta la situación problemática antes descrita, se identifica como **problema de la investigación:** ¿Cómo mejorar el proceso de entrega y devolución de las partes, piezas y herramientas de la Dirección de Gestión Tecnológica?

**Objeto de estudio:** Procesos de gestión de entrega y devolución de piezas y herramientas.

**Objetivo general:** Desarrollar un módulo de un sistema informático que permita la mejora del proceso de entrega y devolución de las partes, piezas y herramientas de la Dirección de Gestión Tecnológica.

**Campo de acción:** Procesos de gestión de entrega y devolución de las partes, piezas y herramientas del Local de Piezas Recuperadas de la Dirección de Gestión Tecnológica.

Para dar cumplimiento al objetivo general y teniendo en cuenta los elementos mencionados se obtuvo un conjunto de **tareas de investigación**, las cuales se listan a continuación:

**Tareas de investigación:**

- ✓ Estudio bibliográfico del tema en cuestión y selección de la metodología y herramientas para la elaboración del marco teórico de la investigación.
- ✓ Identificación de los requisitos y elaboración del diseño de la aplicación para la selección de las características y funcionalidades del sistema.
- ✓ Diseño de una Base de Datos para gestionar la información relacionada con el proceso que se automatiza.
- ✓ Implementación del sistema para gestionar la entrega y devolución de partes y piezas en la Dirección de Gestión Tecnológica de la Universidad de las Ciencias Informáticas.
- ✓ 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: para estudiar la trayectoria histórica real del fenómeno y las tendencias del uso actual de sistemas de gestión de entrega y devolución de piezas y herramientas con el fin de seleccionar las más apropiadas para darle cumplimiento al objetivo general de la investigación.
- ✓ Analítico-Sintético: para realizar un análisis de los documentos, las publicaciones, bibliografías y en general de la información relacionada con sistemas de gestión de entrega y devolución de piezas y herramientas 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 de Gestión de partes y piezas.
- ✓ Modelación: Se utiliza 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:

- ✓ **Entrevista:** A especialistas y técnicos de la DGT para evaluar y constatar cómo se realiza el proceso de entrega y devolución de piezas y herramientas.
- ✓ **Análisis de documentos:** permite el estudio y valoración crítica de la bibliografía específica vinculada con el tema, así como de documentos normativos de la DGT que sirvieron para la construcción de los fundamentos teóricos recogidos en el capítulo I.

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. Los capítulos que lo conforman son los siguientes:

**Capítulo 1: Fundamentación teórica de la investigación.** Se exponen los elementos teóricos que sustentan la investigación. Se realiza un estudio de las soluciones similares y se analizan las metodologías, tecnologías y herramientas que se ajustan al desarrollo de la investigación, justificando la selección y utilización de cada una de ellas.

**Capítulo 2: Análisis y Diseño del Sistema.** En este capítulo se realiza un estudio del negocio con el objetivo de determinar las necesidades de información. A partir de dichas necesidades se identifican los requisitos funcionales, se obtendrá el diseño de la base de datos y los diagramas de clases, además se define la arquitectura y los patrones de diseño a usar en la realización de la aplicación.

**Capítulo 3: Implementación y pruebas.** Este capítulo se refiere a la implementación de la solución, se aborda específicamente cómo se realiza la implementación de los subsistemas de almacenamiento, visualización e integración, teniendo en cuenta los requisitos y necesidades del negocio. De igual forma se especifican las pruebas realizadas y los resultados arrojados por estas.

## **Capítulo 1. Fundamentación Teórica.**

### **1.1. Introducción.**

Este capítulo presenta los fundamentos teóricos que respaldan la investigación. Se explican y vinculan los conceptos que permiten una mejor comprensión de la situación problemática planteada. Se realiza un análisis de sistemas y soluciones similares. Además, se examinan las tecnologías y herramientas utilizadas para alcanzar el objetivo general de la investigación, y se define la metodología empleada para su desarrollo.

### **1.2 Fundamentos teóricos asociados al tema.**

**Gestión Tecnológica:** La UCI cuenta con una infraestructura tecnológica capaz de soportar los proyectos de desarrollo de Software previstos. Para ello cuenta con Computadoras conectadas en Red y sus correspondientes Unidades de Respaldo de Energía (UPS), equipos de Copiado e Impresión, Laptops y Dispositivos Móviles entre Teléfonos Celulares, Lectores de Libros Electrónicos y Tablets. (Impacto Económico y Social de los Servicios Técnicos de la Dirección de Gestión Tecnológica., 2021)

Los dispositivos antes mencionados requieren de atención técnica especializada para su adquisición, instalación, configuración, mantenimiento programado, reparación, modernización y renovación. Encargada de estas tareas es la Dirección de Gestión Tecnológica (DGT), subordinada a la Dirección General de Tecnología y tiene como misión: “Fortalecer y mantener la infraestructura tecnológica de la UCI, para impulsar los procesos de formación, producción, investigación y administrativos.

**Funciones de la DGT:** Sus funciones están centradas en brindar un servicio técnico y de gestión con calidad. Como objetivo principal de trabajo que se tiene en la DGT es: “brindar aseguramiento tecnológico a los procesos fundamentales de la universidad, con agilidad y profesionalidad, con organización y control de los recursos utilizados, logrando un impacto perceptible en la Universidad”. (Impacto Económico y Social de los Servicios Técnicos de la Dirección de Gestión Tecnológica., 2021)

**Reportes o Incidencias.** Los reportes de tecnología para el Grupo de Asistencia Técnica (GAT) son realizados por los trabajadores de la UCI a través del sitio <http://incidencias.uci.cu> (anteriormente <http://tecnologia.uci.cu>). Esta es la vía de atención básica a los problemas que surgen en medios tecnológicos. Como parte de los reportes se pueden realizar ajustes en los equipos, tanto de Hardware como de Software. En caso de roturas se realizan reparaciones, con piezas nuevas o con piezas recuperadas en el Taller de la DGT. Las piezas nuevas pueden ser obtenidas a través de Copextel o de las que la DGT adquiere mediante compra, mayoritariamente a

Copextel o a Cimex. Los mantenimientos son programados o solicitados por los usuarios.

La atención a reportes la realizan los Técnicos del GAT. Esto implica no depender de los servicios especializados de empresas como Copextel y Tecún (Cimex). Por tanto, es evidente que la atención a reportes por el GAT ahorra a la UCI una cifra considerable de recursos financieros, por conceptos de reparación y revisión y diagnósticos. (Impacto Económico y Social de los Servicios Técnicos de la Dirección de Gestión Tecnológica., 2021)

**Piezas Recuperadas:** Es considerable el ahorro de recursos financieros a través de las reparaciones que se realizan con piezas recuperadas. Estas son piezas reparadas en el Taller de la DGT o piezas en buen estado que han sido utilizadas en computadoras (PC) que se modernizan y luego se reutilizan, previa revisión y mantenimiento técnico. Nótese que, al reparar una computadora con piezas reparadas o recuperadas, baja el nivel de adquisición de piezas que la UCI realiza a Cimex o Copextel y, por tanto, ahorros de recursos financieros. (Impacto Económico y Social de los Servicios Técnicos de la Dirección de Gestión Tecnológica., 2021)

**Procesos de gestión de entrega y devolución de piezas y herramientas:** Cuando se registra una incidencia, el técnico responsable acude al lugar para su diagnóstico y resolución. En caso de requerir el reemplazo de algún componente hardware o herramienta para el trabajo y poder restaurar el servicio, envía mediante el correo una solicitud de partes al encargado del inventario. Se verifica la disponibilidad de partes y piezas técnicas en el LPR. De existir se aprueba a través de una notificación vía correo. El técnico recibe las partes autorizadas del LPR, registrando el préstamo en un vale de entrega, luego, sustituye el componente defectuoso por el repuesto funcional para restablecer el servicio. Finalmente, el técnico retorna al almacén el componente defectuoso, registrando la devolución en un vale que sería el de entrada para su reparación o baja.

Actualmente no se cuenta con un sistema que satisfaga las necesidades de la Dirección de Gestión Tecnológica. Para el desarrollo de una propuesta de solución se hace necesario examinar los sistemas similares al que se quiere desarrollar.

### **1.3. Sistemas similares existentes.**

Tanto a nivel internacional como nacional se han desarrollado distintos sistemas que apoyan la gestión y control de piezas y repuestos de forma genérica o se dedican directamente a la gestión de estos, integrando entre sus procesos otras funcionalidades,

diseñados para manejar la información incluyendo la gestión de inventario y órdenes de trabajo.

Algunos ejemplos de sistemas de este tipo a nivel internacional son:

**1.3.1 Oracle Enterprise Asset Management (EAM):** es una solución completa de gestión de activos empresariales ofrecida por Oracle. Algunas de las funcionalidades que brinda el sistema son: (Simmons, 2019)

Funcionalidades de Oracle Enterprise Asset Management:

1. Gestión integral de activos: Oracle EAM permite gestionar todos los aspectos relacionados con los activos físicos de una organización, incluyendo la planificación, adquisición, mantenimiento, seguimiento y desinversión de activos.
2. Mantenimiento preventivo y correctivo: La solución ofrece funcionalidades para la programación y ejecución de tareas de mantenimiento preventivo y correctivo. Esto incluye la planificación de actividades, asignación de recursos, gestión de órdenes de trabajo y seguimiento del progreso de las tareas de mantenimiento.
3. Gestión de inventario: Oracle EAM incluye capacidades de gestión de inventario para asegurar el suministro adecuado de piezas de repuesto y materiales necesarios para el mantenimiento de activos. Esto implica la gestión de stocks, control de ubicaciones, seguimiento de movimientos y reposición de inventario.
4. Análisis y generación de informes: La solución ofrece herramientas de análisis y generación de informes para evaluar el rendimiento de los activos, identificar tendencias, optimizar estrategias de mantenimiento y realizar análisis de costos. También proporciona informes personalizados y paneles de control para una visión rápida y efectiva del estado de los activos. (ORACLE, 2023)

Entre las principales desventajas que presenta este software se encuentran:

1. Licencia de pago: Oracle EAM es una solución comercial que requiere una licencia de pago. Esto implica un costo asociado y puede variar según los módulos y funcionalidades seleccionadas.
2. Software propietario: Oracle EAM es un software propietario, lo que significa que no es de código abierto ni gratuito. Esto limita la capacidad de modificar y adaptar el software según las necesidades.
3. Complejidad de implementación: La implementación de Oracle EAM puede ser compleja y requerir conocimientos técnicos especializados.

Al evaluar estas desventajas junto con los beneficios y funcionalidades de Oracle EAM (ORACLE, 2023) se arriba a la conclusión que este sistema no satisface los requerimientos planteados por lo que no es posible utilizarlo como respuesta al problema en cuestión.

**1.3.2 SAP Asset Management:** es un componente del software empresarial SAP Enterprise Asset Management (EAM) que se enfoca en la gestión de activos físicos dentro de una organización. Proporciona funcionalidades para planificar, mantener y optimizar los activos de una empresa a lo largo de todo su ciclo de vida. (SAP, 2023)

Algunas funcionalidades que brinda el sistema son:

1. **Gestión integral de activos:** SAP Asset Management abarca todo el ciclo de vida de los activos, desde la planificación hasta la retirada. Proporciona capacidades para la gestión de activos, incluyendo la planificación, la adquisición, el mantenimiento, la programación de tareas y la desinversión.
2. **Mantenimiento preventivo y correctivo:** La solución permite la programación y ejecución de tareas de mantenimiento preventivo y correctivo. También ofrece funciones para la gestión de órdenes de trabajo, la asignación de recursos, la programación de mantenimiento basada en condiciones y la gestión de servicios externos.
3. **Integración con otros sistemas:** SAP Asset Management se integra con otros módulos y sistemas de SAP, como el módulo de gestión de inventario (SAP MM) y el módulo de gestión de compras (SAP PM). Esto facilita la sincronización de datos y la gestión de activos en el contexto más amplio de los procesos empresariales.
4. **Análisis y generación de informes:** La solución ofrece capacidades de análisis y generación de informes para evaluar el rendimiento de los activos, identificar tendencias, tomar decisiones basadas en datos y optimizar las estrategias de mantenimiento.

Sin embargo, es importante tener en cuenta las siguientes desventajas:

1. **Licencia de pago:** SAP Asset Management es una solución comercial que requiere una licencia de pago.
2. **Complejidad de implementación:** La implementación de SAP Asset Management puede ser compleja y requerir conocimientos técnicos especializados.

3. Dependencia de la infraestructura SAP: Si no se utiliza la infraestructura de SAP, la integración con sistemas existentes puede ser un desafío adicional.

En resumen, SAP Asset Management ofrece una amplia gama de funcionalidades para la gestión de activos empresariales, incluyendo mantenimiento, planificación, integración y análisis. Sin embargo, se consideran los aspectos negativos relacionados con la licencia de pago, la complejidad de implementación y la dependencia de la infraestructura SAP por lo que no se elige esta solución para dar respuesta al problema planteado.

**1.3.3 IBM Maximo:** es una solución de gestión de activos y mantenimiento desarrollada por IBM. Se trata de un sistema de gestión empresarial que se utiliza para administrar y optimizar los activos físicos, como equipos, instalaciones, vehículos y otros activos, a lo largo de su ciclo de vida. Algunas funcionalidades que brinda el sistema son: (Maximo, 2023)

1. Gestión integral de activos: IBM Maximo permite administrar todos los aspectos relacionados con los activos físicos de una organización, incluyendo el registro de activos, la gestión de ubicaciones, la programación de mantenimiento, el seguimiento del ciclo de vida y la gestión de la desinversión.
2. Mantenimiento preventivo y correctivo: La solución proporciona capacidades para la planificación y ejecución de tareas de mantenimiento preventivo y correctivo. Esto incluye la programación de actividades, la asignación de recursos, la gestión de órdenes de trabajo y el seguimiento del progreso de las tareas de mantenimiento.
3. Gestión de inventario y repuestos: IBM Maximo ofrece funcionalidades para la gestión de inventario y repuestos necesarios para el mantenimiento de los activos. Esto incluye la gestión de stocks, la reposición de inventario, el control de ubicaciones y la optimización de los niveles de inventario.
4. Gestión de contratos y proveedores: La solución permite la gestión de contratos de servicio con proveedores externos y el seguimiento de las actividades y el rendimiento de los proveedores. También facilita la gestión de acuerdos de nivel de servicio (SLA) y la generación de informes relacionados con la gestión de proveedores.

Entre las principales desventajas que presenta este software se encuentran:

1. Licencia de pago: IBM Maximo es una solución comercial que requiere una licencia de pago.
2. Software propietario: IBM Maximo es un software propietario, lo que significa que no es de código abierto ni gratuito.
3. Dependencia del proveedor: Al utilizar una solución comercial como IBM Maximo, existe una dependencia del proveedor para el soporte, las actualizaciones y el mantenimiento continuo del software.

Al evaluar estas desventajas junto con las funcionalidades y beneficios de IBM Maximo de determina que el mismo no se ajusta completamente a las necesidades planteadas.

**1.3.4 ASSETS** En la UCI, el Sistema de Auditoría de Activos Fijos y el Sistema de Conciliación de Activos Fijos funcionan en conjunto con ASSETS, el cual es un sistema centralizado por la Dirección de Contabilidad y Finanzas, es un Sistema de Gestión Integral estándar y parametrizado que permite:

1. Control de los procesos de compras, ventas, producción, taller, inventario, finanzas, contabilidad, presupuesto, activos fijos, útiles y herramientas y recursos humanos.
2. Dispone de métodos novedosos para administración y planificación de inventarios, así como una amplia gama de análisis y consultas que le permitirán no sólo conocer exactamente la situación actual, sino proyectar decisiones futuras.
3. Es un sistema flexible, amigable, con ayuda en línea que puede ser instalado en una microcomputadora o sobre varias, funcionando en ambiente multiusuario incluidas estaciones remotas.
4. Proporciona opciones de seguridad que le permiten limitar el acceso a los diferentes procesos del sistema de acuerdo con el perfil de cada usuario.

Entre las principales desventajas que presenta este software se encuentran:

1. Presenta inconsistencia en la integridad de los datos.
2. Imposibilita ver reportes por diferentes conceptos.
3. No es concurrente e inconvenientemente solo pueden interactuar con la aplicación un grupo muy pequeño de usuarios, donde se tiene que pagar una licencia de código por cada uno de ellos, impidiendo a los usuarios responsables de los AFT de las diferentes áreas de responsabilidad de la UCI tener acceso al mismo.

4. Al ser un sistema propietario bajo licencia no garantiza la soberanía tecnológica para Cuba pues recibe soporte de una compañía propietaria que impide la colaboración de nuestro país con nuevas funcionalidades. (CLEMENTE, 2012)

### 1.3.5 Análisis del estudio realizado a soluciones existentes

A modo de resumen y con el fin de establecer comparaciones que inciden en la toma de decisiones se presenta a continuación una tabla resumen donde se recogen los datos más significativos de cada una de las soluciones homólogas estudiadas, con criterios seleccionados por la autora de la presente investigación teniendo en cuenta los requerimientos planteados por el cliente:

| <b>SOLUCIONES</b><br><b>CRITERIOS</b> | <b>Oracle Enterprise Asset Management (EAM)</b> | <b>SAP Asset Management</b> | <b>IBM Maximo</b> | <b>ASSETS</b> |
|---------------------------------------|---|-----------------------------|-------------------|---------------|
| Gestión Integral de Activos           | Si  | Si                          | Si                | Si            |
| Gestión de inventarios y repuestos    | Si  | Si                          | Si                | Si            |
| Análisis y generación de Informes     | Si  | Si                          | No                | Si            |
| Tipo de software Libre                | No  | No                          | No                | No            |
| Aplicación Web                        | Si  | Si                          | Si                | Si            |

*Tabla 1 Características generales de las soluciones estudiadas. Fuente: Elaboración propia.*

Cada uno de los sistemas que aquí se han expuesto tienen como semejanza que cada uno está desarrollado bajo un software propietario por lo que requieren una licencia de pago, lo que la utilización de alguno de ellos implica estar atados a las condiciones establecidas por los proveedores, incluyendo restricciones en la personalización, actualizaciones y soporte técnico. Por estos motivos no constituyen una solución factible para la Dirección de Tecnología, ya que no cumplen con la independencia tecnológica que se quiere alcanzar en el país.

Los sistemas analizados poseen características semejantes a las necesidades planteadas, se puede afirmar que a pesar de no constituir soluciones factibles el estudio de estas plataformas permitió entender el funcionamiento de como realizan el control, planificación y gestión de activos desde la entrada hasta la retirada, así como las formas de evaluación y métodos novedosos para administración y planificación de inventarios que utilizan, estas características se tienen en cuenta a la hora del desarrollo y se profundiza buscando la eficiencia del módulo, para ello es necesario la definición de una metodología a seguir.

#### **1.4. 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.

##### **1.4.1 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 decide 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).)

#### 1.4.2 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.

| 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   |
| Construcción |                         |   |

|            |        |   |
|------------|--------|---|
| Transición |        | 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. |
|            | Cierre | En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.   |

Tabla 3 Descripción de las fases AUP-UCI. Fuente Elaboración propia.

### 1.4.3 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, la autora 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.

## **1.5. 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 éste 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)

### **1.5.1. 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, 2008) (infostudio, 2011)

- Consistencia tecnológica: Al estar desarrollado el sistema Backend principal en Python, construir el nuevo módulo en el mismo lenguaje permite mantener un stack tecnológico 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 testado 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 bugs 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.

### **1.5.2 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:

1. HTML es el lenguaje base para crear el contenido y la estructura de las páginas web. Es un estándar necesario en la web.
2. Se separa el contenido de la presentación. El HTML se enfoca en la información, mientras que con CSS se define la parte visual.
3. Es soportado por todos los navegadores web permitiendo crear contenido accesible desde cualquier dispositivo.
4. Permite incorporar contenidos multimedia como imágenes, videos y audio de forma sencilla.
5. Existe gran compatibilidad con otros lenguajes web como JavaScript, CSS, XML, etc.
6. Hay muchas librerías, frameworks y herramientas que potencian y facilitan el desarrollo en HTML.

### **1.5.3 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 puedes aplicar directamente en tu código HTML para estilizar elementos y diseñar tu sitio 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 cachear los archivos CSS externos.
- Tailwind CSS se puede combinar fácilmente con JavaScript y frameworks de JavaScript

#### **1.5.4 JavaScript.**

Para el desarrollo del módulo del lado del Frontend que se integrará al sistema existente, se ha decidido utilizar JavaScript con el fin de mantener un stack tecnológico 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.

#### **1.6. 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.

### **1.6.1 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.

### **1.6.2 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)

1. 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.

2. 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 de imágenes automática. Esto acelera el desarrollo y permite a los desarrolladores concentrarse en la lógica de la aplicación en lugar de la configuración.
3. 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.
4. Tiene una comunidad activa y una gran cantidad de recursos disponibles, lo que facilita encontrar ayuda, tutoriales y bibliotecas adicionales.
5. Es compatible con múltiples plataformas de alojamiento, lo que facilita el despliegue.
6. Aunque Next.js tiene una estructura predefinida, también es lo suficientemente flexible como para adaptarse a necesidades específicas.

### **1.7. 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 con base en 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.

#### **1.7.1 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 extensions, usuario 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.

## **1.8 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:

### **1.8.1. 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.)

### **1.8.2 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 open source 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.

### **1.8.3 Entorno de Desarrollo Integrado. Visual Studio Code (VS Code)**

Un Entorno Integrado de Desarrollo en inglés Integrated Development Environment (IDE) es un programa compuesto por un conjunto de herramientas para un desarrollador 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:

1. VS Code es completamente gratuito y de código abierto.
2. Está disponible para Windows, macOS y Linux, lo que lo hace accesible para una amplia variedad de usuarios en diferentes sistemas operativos.
3. Es compatible con una amplia gama de lenguajes de programación, lo que lo hace versátil y adecuado para desarrolladores que trabajan en diferentes tecnologías.
4. 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.
5. Ofrece una sólida capacidad de depuración integrada que permite realizar un seguimiento y solucionar problemas en el código de manera eficiente.
6. VS Code está diseñado para ser rápido y eficiente, incluso cuando se trabaja en proyectos grandes.

### **Conclusiones Parciales**

La información presentada en este capítulo sobre los sistemas informáticos que se analizan y que utilizan elementos de interés, enriqueció el conocimiento del dominio del problema en cuestión. Aunque las plataformas similares estudiadas no cumplieron con todas las necesidades del módulo actual, se aprendió sobre cómo funcionan los

sistemas de este tipo y se seleccionaron algunas características importantes para el desarrollo del módulo.

La elección de las herramientas apropiadas, en combinación con la metodología AUP en su variante UCI, así como la selección de un lenguaje de programación y un lenguaje de modelado, se traducirá en la definición precisa de la estructura y el diseño, aportando información esencial para la ejecución exitosa del módulo y garantizando la adhesión a las mejores prácticas en el proceso de desarrollo de software. Además de agilizar el desarrollo, y brindar una solución completa y eficiente en la implementación.

## **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 identifican 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**

Actualmente en la DGT se desarrolla un sistema informático para llevar a cabo la gestión de la información que generan los procesos del área. Dicho sistema ha sido concebido por módulos, asociados a grupos de procesos definidos en el área. Con el objetivo de optimizar y agilizar el proceso de entrega y devolución de partes, piezas y herramientas, se propone la implementación de un sistema de gestión integral. Esta propuesta tiene como objetivo mejorar la eficiencia operativa y reducir los riesgos asociados con los procedimientos actuales.

Para el desarrollo de la solución se pondrá en funcionamiento un Sistema de Gestión de Inventario Digital (SGID) diseñado específicamente para las necesidades de la DGT el cual permitirá un registro preciso y en tiempo real de todas las partes, piezas y herramientas disponibles en el Local de Piezas Recuperadas (LPR), además se proporcionará acceso en línea al inventario del LPR para que los empleados de la DGT puedan verificar la disponibilidad de partes y piezas de forma remota, esto eliminará la necesidad de desplazamientos físicos al local, ahorrando tiempo y recursos. También cada movimiento de partes y piezas quedará registrado electrónicamente, incluyendo entregas y devoluciones reduciendo el riesgo de pérdida de datos y errores en la gestión.

Esta propuesta de solución busca modernizar y optimizar el proceso de entrega y devolución de partes, piezas y herramientas en la DGT. La transición hacia un sistema digital proporcionará beneficios notables en términos de eficiencia y precisión, al tiempo que reducirá los riesgos asociados con los procedimientos manuales.

### 2.3 Modelo de Dominio

Un modelo de dominio o conceptual se enfoca en los conceptos y relaciones clave que son relevantes para el sistema, y se utiliza para identificar y definir los objetos, entidades, atributos, relaciones y comportamientos que forman parte del sistema. es el artefacto más importante a crear durante el análisis orientado a objetos. La designación de modelo de dominio ofrece la ventaja de subrayar fuertemente una concentración en los conceptos del dominio, no en las entidades del software. (Spiegato, 2023)

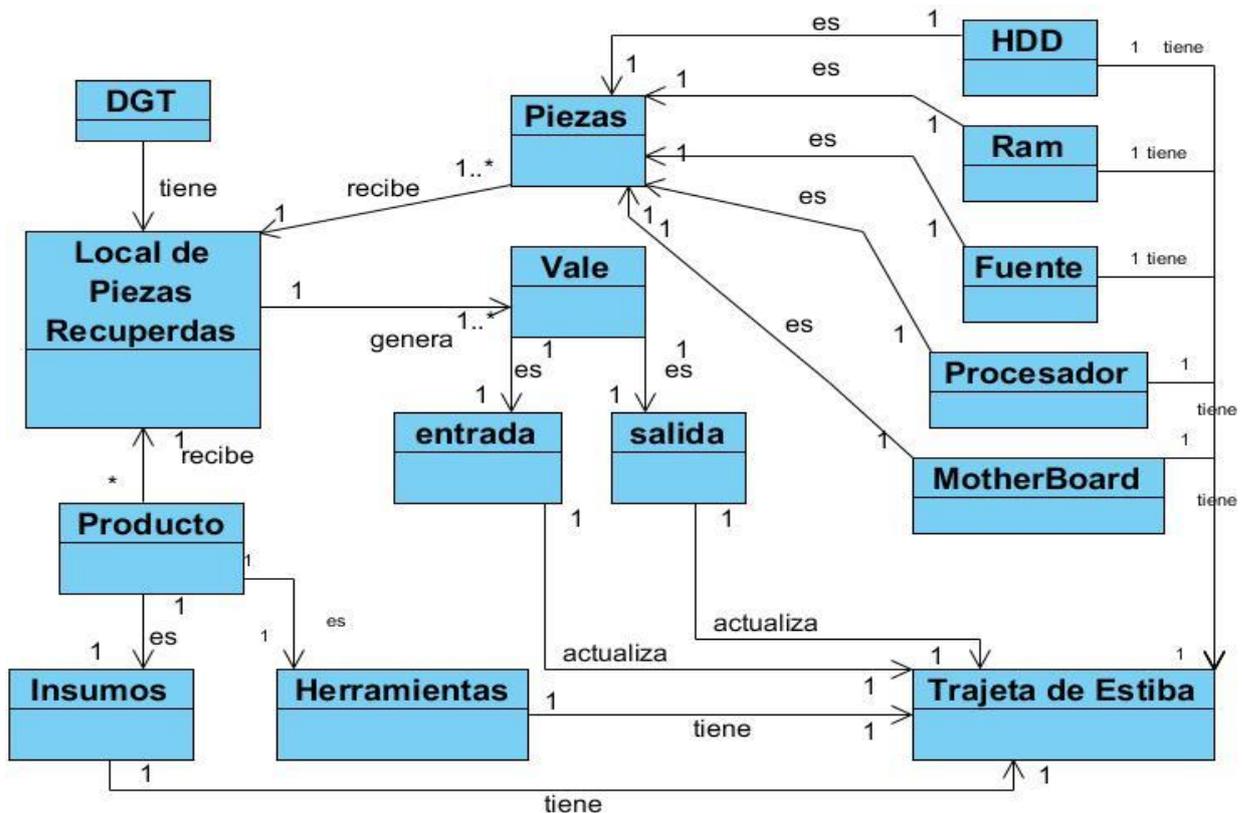


Figura 5 Modelo de Dominio. Fuente: Elaboración Propia

#### 2.3.2 Definición de los conceptos del Diagrama de modelo de Dominio.

**DGT:** Dirección de Gestión Tecnológica de la Universidad de las Ciencias Informáticas.

**Local de piezas Recuperadas:** Instalación que constituye un "almacén" donde se guardan tanto piezas, herramientas e insumos.

**Vale:** Documento que registra la entrada o salida de las piezas o productos a modo de control.

**Tarjeta de Estiba:** Documento que indica la constancia de la cantidad que hay de un producto determinado en el local.

**Producto:** Designación genérica que engloba tanto a los insumos como a las herramientas contenidas en el LPR.

**Insumos:** Materiales consumibles utilizados en los procesos de reparación.

**Herramientas:** Instrumental empleado por los operarios durante las labores de reparación.

**Piezas:** Componentes de las computadoras, tanto internos como externos ya sean nuevos o recuperados.

## 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:

| <b>No.</b>  | <b>Nombre</b> | <b>Descripción</b>  | <b>Prioridad</b> | <b>Complejidad</b> |
|-------------|---------------|---|------------------|--------------------|
| <b>RF 1</b> | Agregar Hdd   | El sistema permitirá agregar un nuevo Hdd (capacidad, tipo, No_serie, marca, modelo, clave, descripción ) | <i>Alta</i>      | <i>Alta</i>        |
| <b>RF 2</b> | Modificar Hdd | El sistema permitirá modificar los datos del Hdd anteriormente creado. (capacidad, tipo, No_serie,        | <i>Media</i>     | <i>Alta</i>        |

|              |                       |  |              |              |
|--------------|-----------------------|--|--------------|--------------|
|              |                       | marca, modelo, clave, descripción)   |              |              |
| <b>RF 3</b>  | Eliminar Hdd          | El sistema deberá permitir eliminar un Hdd anteriormente creado.   | <i>Media</i> | <i>Media</i> |
| <b>RF 4</b>  | Listar Hdd            | El sistema deberá permitir listar todos los Hdd creados anteriormente.   | <i>Alta</i>  | <i>Baja</i>  |
| <b>RF 5</b>  | Obtener Hdd           | El sistema deberá permitir obtener la información detallada de un Hdd  | <i>Baja</i>  | <i>Media</i> |
| <b>RF 6</b>  | Agregar MotherBoard   | El sistema permitirá agregar una nueva MotherBoard (socket, generación, No_serie, marca, modelo, clave, descripción)                               | <i>Alta</i>  | <i>Alta</i>  |
| <b>RF 7</b>  | Modificar MotherBoard | El sistema permitirá modificar los datos de la MotherBoard anteriormente creada. (socket, generación, No_serie, marca, modelo, clave, descripción) | <i>Media</i> | <i>Alta</i>  |
| <b>RF 8</b>  | Eliminar MotherBoard  | El sistema deberá permitir eliminar una MotherBoard anteriormente creada.  | <i>Media</i> | <i>Media</i> |
| <b>RF 9</b>  | Listar MotherBoard    | El sistema deberá permitir listar todas las MotherBoard creadas anteriormente.   | <i>Alta</i>  | <i>Baja</i>  |
| <b>RF 10</b> | Obtener MotherBoard   | El sistema deberá permitir obtener la información detallada de una MotherBoard   | <i>Baja</i>  | <i>Media</i> |
| <b>RF 11</b> | Agregar Ram.          | El sistema permitirá agregar una nueva Ram. (capacidad, tipo, pines, No_serie, marca, modelo, clave, descripción)                                  | <i>Alta</i>  | <i>Alta</i>  |

|              |                  |   |              |              |
|--------------|------------------|---|--------------|--------------|
| <b>RF 12</b> | Modificar Ram    | El sistema permitirá modificar los datos de la Ram anteriormente creada.<br>(capacidad, tipo, pines, No_serie, marca, modelo, clave, descripción) | <i>Media</i> | <i>Alta</i>  |
| <b>RF 13</b> | Eliminar Ram     | El sistema deberá permitir eliminar una Ram anteriormente creada.   | <i>Media</i> | <i>Alta</i>  |
| <b>RF 14</b> | Listar Ram       | El sistema deberá permitir listar todas las Ram creadas anteriormente.  | <i>Alta</i>  | <i>Baja</i>  |
| <b>RF 15</b> | Obtener Ram      | El sistema deberá permitir obtener la información detallada de una Ram  | <i>Baja</i>  | <i>Media</i> |
| <b>RF 16</b> | Agregar Fuente   | El sistema permitirá agregar una nueva Fuente (voltaje, tipo, No_serie, marca, modelo, clave, descripción)  | <i>Alta</i>  | <i>Alta</i>  |
| <b>RF 17</b> | Modificar Fuente | El sistema permitirá modificar los datos de la Fuente anteriormente creada. (voltaje, tipo, No_serie, marca, modelo, clave, descripción)          | <i>Media</i> | <i>Alta</i>  |
| <b>RF 18</b> | Eliminar Fuente  | El sistema deberá permitir eliminar una Fuente anteriormente creado   | <i>Media</i> | <i>Alta</i>  |
| <b>RF 19</b> | Listar Fuente    | El sistema deberá permitir listar todas las Fuentes creadas anteriormente.  | <i>Alta</i>  | <i>Baja</i>  |
| <b>RF 20</b> | Obtener Fuente   | El sistema deberá permitir obtener la información detallada de una Fuente   | <i>Baja</i>  | <i>Media</i> |

|              |                      |   |              |              |
|--------------|----------------------|---|--------------|--------------|
| <b>RF 21</b> | Agregar Procesador   | El sistema permitirá agregar un nuevo Procesador (modelo, socket, No_serie, marca, modelo, clave, descripción)                                  | <i>Alta</i>  | <i>Alta</i>  |
| <b>RF 22</b> | Modificar Procesador | El sistema permitirá modificar los datos del Procesador anteriormente creado.<br>(modelo, socket, No_serie, marca, modelo, clave, descripción)  | <i>Media</i> | <i>Alta</i>  |
| <b>RF23</b>  | Eliminar Procesador  | El sistema deberá permitir eliminar un Procesador anteriormente creado  | <i>Media</i> | <i>Alta</i>  |
| <b>RF 24</b> | Listar Procesador    | El sistema deberá permitir listar todos los Procesadores creados anteriormente.   | <i>Alta</i>  | <i>Baja</i>  |
| <b>RF 25</b> | Obtener Procesador   | El sistema deberá permitir obtener la información detallada de un Procesador  | <i>Baja</i>  | <i>Media</i> |
| <b>RF 26</b> | Agregar Producto     | El sistema permitirá agregar un nuevo Producto (cantidad, unidad de medida, tipo, No_serie, clave, descripción)                                 | <i>Alta</i>  | <i>Alta</i>  |
| <b>RF 27</b> | Modificar Producto   | El sistema permitirá modificar los datos del Producto anteriormente creado.<br>(cantidad, unidad de medida, tipo, No_serie, clave, descripción) | <i>Media</i> | <i>Alta</i>  |
| <b>RF 28</b> | Eliminar Producto    | El sistema deberá permitir eliminar un Producto anteriormente creado  | <i>Media</i> | <i>Alta</i>  |
| <b>RF 29</b> | Listar Producto      | El sistema deberá permitir listar todos los Productos creados anteriormente.  | <i>Alta</i>  | <i>Baja</i>  |

|              |                                  |  |              |              |
|--------------|----------------------------------|--|--------------|--------------|
| <b>RF 30</b> | Obtener Producto                 | El sistema deberá permitir obtener la información detallada de un Producto   | <i>Baja</i>  | <i>Media</i> |
| <b>RF 31</b> | Agregar Pieza sin Calificación   | El sistema permitirá agregar una nueva pieza que no entra en las categorías definidas con los siguientes campos: (No_serie, modelo, clave, descripción, no_inventario)                         | <i>Alta</i>  | <i>Alta</i>  |
| <b>RF 32</b> | Modificar Pieza sin Calificación | El sistema permitirá modificar una nueva pieza que no entra en las categorías definidas, anteriormente creada con los siguientes campos: (No_serie, modelo, clave, descripción, no_inventario) | <i>Media</i> | <i>Alta</i>  |
| <b>RF 33</b> | Eliminar Pieza sin Calificación  | El sistema permitirá eliminar una nueva pieza que no entra en las categorías definidas, anteriormente creada con los siguientes campos: (No_serie, modelo, clave, descripción, no_inventario)  | <i>Media</i> | <i>Alta</i>  |
| <b>RF 34</b> | Listar Pieza sin Calificación    | El sistema permitirá listar todas las piezas anteriormente creadas   | <i>Alta</i>  | <i>Baja</i>  |
| <b>RF 35</b> | Obtener Pieza sin Calificación   | El sistema deberá permitir obtener la información detallada de una pieza en específico   | <i>Baja</i>  | <i>Media</i> |
| <b>RF 36</b> | Crear Vale de entrada            | El sistema permitirá crear un nuevo Vale de Entrada (No_vale, personal, user_id)   | <i>Alta</i>  | <i>Alta</i>  |
| <b>RF 37</b> | Crear Vale de salida             | El sistema permitirá crear un nuevo Vale de Salida (No_vale, personal)   | <i>Alta</i>  | <i>Alta</i>  |

|              |   |   |              |              |
|--------------|---|---|--------------|--------------|
| <b>RF 38</b> | Editar Vale   | El sistema permitirá editar los datos del Vale anteriormente creado. (No_vale, personal)  | <i>Media</i> | <i>Alta</i>  |
| <b>RF 39</b> | Listar Vale   | El sistema deberá permitir listar todos los Vales creados anteriormente.  | <i>Alta</i>  | <i>Baja</i>  |
| <b>RF 40</b> | Obtener la cantidad de vales por tipo                               | A partir de la información existentes de los vales, el sistema deberá filtrar los vales por tipo y mostrar la cantidad de vales existentes.                                 | <i>Alta</i>  | <i>Media</i> |
| <b>RF 41</b> | Obtener cantidad de piezas, productos y herramientas en el almacén. | A partir de la información existentes de las piezas, productos y herramientas el sistema deberá mostrar la cantidad total de piezas productos y herramientas en el almacén. | <i>Alta</i>  | <i>Media</i> |
| <b>RF 42</b> | Obtener cantidad de piezas por tipo en el almacén.                  | A partir de la información existentes de las piezas, el sistema deberá filtrar las piezas por tipo y mostrar la cantidad de piezas existentes.                              | <i>Alta</i>  | <i>Media</i> |

*Tabla 4: Requisitos Funcionales del Sistema. Fuente: Elaboración propia.*

#### **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 clicks 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 4 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 4 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 2 segundos con 50 usuarios concurrentes.

### **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 por encima de la versión 90.

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%.

## **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 #4. En este escenario, el proyecto evaluó previamente el negocio a informatizar y como resultado obtuvo un negocio muy bien definido. Además, el cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. El artefacto que se genera a través de este escenario es la historia de usuario (Villar, 2023)

Las historias de usuario describen una funcionalidad o característica específica del sistema desde la perspectiva del usuario final. Cada tarjeta de usuario generalmente incluye una breve descripción de lo que el usuario espera lograr y por qué es importante para ellos, así como los criterios de aceptación para determinar cuándo se ha completado la funcionalidad. Son una técnica efectiva para garantizar que el desarrollo de software se centre en las necesidades del usuario final y para fomentar la colaboración entre el equipo de desarrollo y los usuarios. Además, las historias de usuario son flexibles y pueden ser actualizadas y refinadas a medida que se desarrolla el sistema y se aprende más sobre las necesidades del usuario.

| Historia de Usuario   |                                       |
|---|---------------------------------------|
| <b>Número: 1</b>  | <b>Nombre de la HU:</b> Gestionar Hdd |
| <b>Programador:</b> Beatriz Pérez Moreno  | <b>Iteración Asignada:</b> 1era       |
| <b>Prioridad:</b> Alta  | <b>Tiempo Estimado:</b> 5 días        |
| <b>Riesgo en Desarrollo:</b> Alto   | <b>Tiempo Real:</b> 3 días            |
| <p><b>1. Objetivo:</b></p> <p>- Ofrece al usuario una vista que permite agregar, eliminar, editar y listar los Hdd con su capacidad y tipo.</p> <p><b>2. Flujo de la acción a realizar:</b></p> <p>-El sistema debe permitir agregar, eliminar, editar y listar el/los Hdd con su capacidad y tipo.</p> <p><b>3. Flujo alternativo</b></p> <p>- El sistema debe permitir ir “Atrás” o “Cancelar” en caso de que la opción seleccionada no sea la deseada.</p> |                                       |

Tabla 5: Historia de Usuario<Gestionar Hdd>. Fuente: Elaboración propia.

| Historia de Usuario                      |  |
|--|--|
| <b>Número:</b>                           | <b>Nombre de la HU:</b> Gestionar Fuente |
| <b>Programador:</b> Beatriz Pérez Moreno | <b>Iteración Asignada:</b> 1era          |

|   |                                |
|---|--------------------------------|
| <b>Prioridad:</b> Alta  | <b>Tiempo Estimado:</b> 5 días |
| <b>Riesgo en Desarrollo:</b> Alto   | <b>Tiempo Real:</b> 3 días     |
| <p><b>1. Objetivo:</b></p> <p>- Ofrece al usuario una vista que permite agregar, eliminar, editar y listar las fuentes con su voltaje y tipo.</p> <p><b>2. Flujo de la acción a realizar:</b></p> <p>-El sistema debe permitir agregar, eliminar, editar y listar la/las fuentes con su voltaje y tipo.</p> <p><b>3. Flujo alterno</b></p> <p>- El sistema debe permitir ir “Atrás” o “Cancelar” en caso de que la opción seleccionada no sea la deseada.</p> |                                |

Tabla 6: Historia de Usuario<Gestionar Fuentes>. Fuente: Elaboración propia.

| Historia de Usuario   |  |
|---|--|
| <b>Número:</b>  | <b>Nombre de la HU:</b> Gestionar Producto |
| <b>Programador:</b> Beatriz Pérez Moreno  | <b>Iteración Asignada:</b> 1era            |
| <b>Prioridad:</b> Alta  | <b>Tiempo Estimado:</b> 7 días             |
| <b>Riesgo en Desarrollo:</b> Alto   | <b>Tiempo Real:</b> 6 días                 |
| <p><b>1. Objetivo:</b></p> <p>- Ofrece al usuario una vista que permite agregar, eliminar, editar y listar los productos con su unidad de medida, cantidad y tipo.</p> <p><b>2. Flujo de la acción a realizar:</b></p> <p>-El sistema debe permitir agregar, eliminar, editar y listar el/los productos con su unidad de medida, cantidad y tipo.</p> <p><b>3. Flujo alterno</b></p> <p>- El sistema debe permitir ir “Atrás” o “Cancelar” en caso de que la opción seleccionada no sea la deseada.</p> |  |

Tabla 7: Historia de Usuario<Gestionar Producto>. Fuente: Elaboración propia.

## 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 Dirección de gestión Tecnológica 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 contruidos 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.

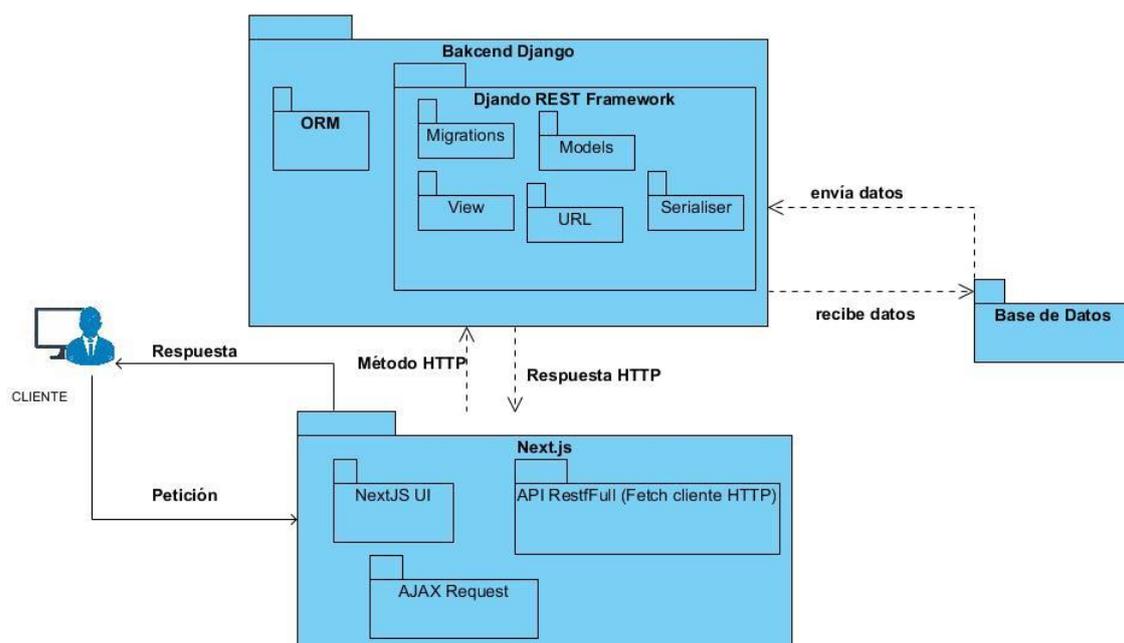


Figura 6 Diseño de la Arquitectura. Fuente: Elaboración Propia

## 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.

Los patrones de diseño 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. (2004)

Los patrones se evidencian en dos grupos como se muestra en la ilustración

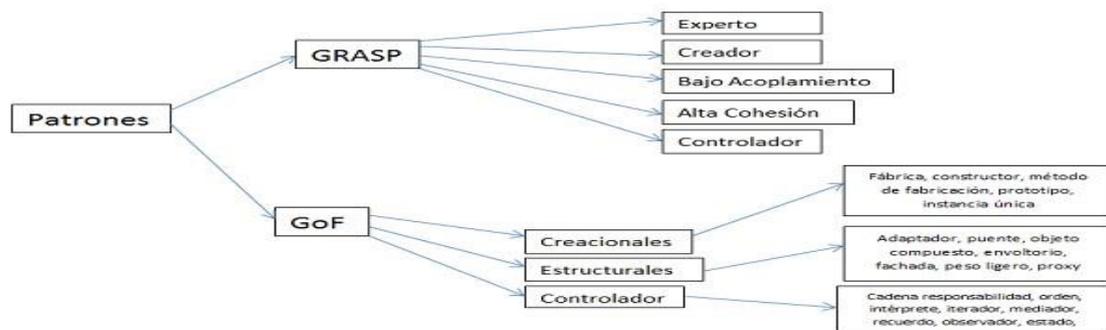


Figura 7 Patrones de Diseño

### 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**.

```

class ValeSerializerRead(serializers.ModelSerializer):
    activos = ActivoSerializer(many=True)
    user = SerializadorDeUsuarioLecturaConPerfil()

    👤 Aarom Cárdenas

    class Meta:
        model = Vales
        fields = '__all__'

```

Figura 8 Patrón Creador. Fuente: Elaboración propia

- 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**.

```

class ValeViewSet(viewsets.ModelViewSet):
    """Genera el CRUD de las nom de marcas"""
    queryset = Vales.objects.all()
    permission_classes = (IsAuthenticated,
                          DjangoModelPermissions,)

    👤 Aarom Cárdenas

    def get_serializer_class(self):
        if self.request.method == 'GET':
            return ValeSerializerRead
        else:
            return ValeSerializerWrite

```

Figura 9 Patrón Controlador. Fuente: Elaboración propia

- 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 Vales(models.Model):

    @ Aarom Cárdenas
    class Tipo(models.TextChoices):
        ENTRADA = 'Entrada', _('Entrada')
        SALIDA = 'Salida', _('Salida')

    numero_vale = models.CharField(unique=True, max_length=255)
    tipo = models.CharField(
        max_length=255, choices=Tipo.choices, default=Tipo.ENTRADA)
    user = models.ForeignKey(Usuario, models.CASCADE)
    activos = models.ManyToManyField(Activo, through="ActivosVales")
    personal = models.CharField(max_length=255)

```

Figura 10 Patrón Experto. Fuente: Elaboración propia

### 2.7.2 Patrones GOF

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:

@api\_view se utiliza para la creación de un controlador(view) api basado en funciones.

```

@api_view(['GET'])
def cant_piezas_in_stock(request):
    try:
        piezas = Piezas.objects.filter(en_uso=False)
        return Response(data: {
            'piezas': piezas.count()
        }, status=status.HTTP_200_OK)
    except Exception as e:
        return Response(data: {'error': str(e)}, status=status.HTTP_400_BAD_REQUEST)

```

Figura 11 Patrón GOF, Decorador. Fuente: Elaboración propia

## 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 usuario. 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.

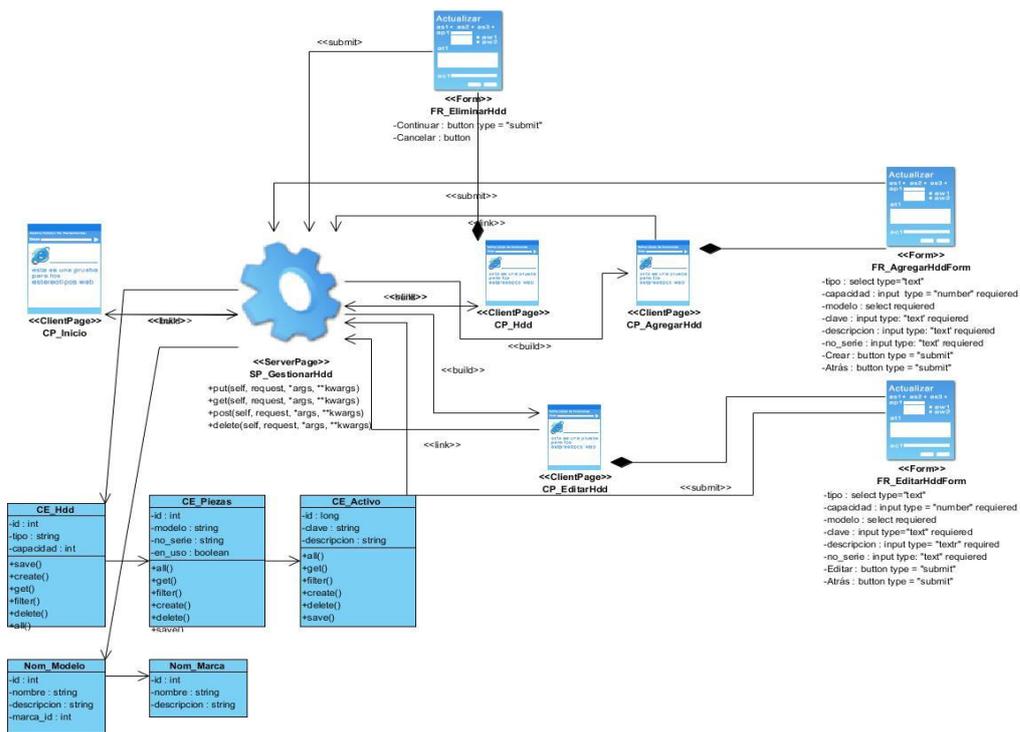


Figura 12 Diagrama de Clases. Fuente: Elaboración propia

## 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

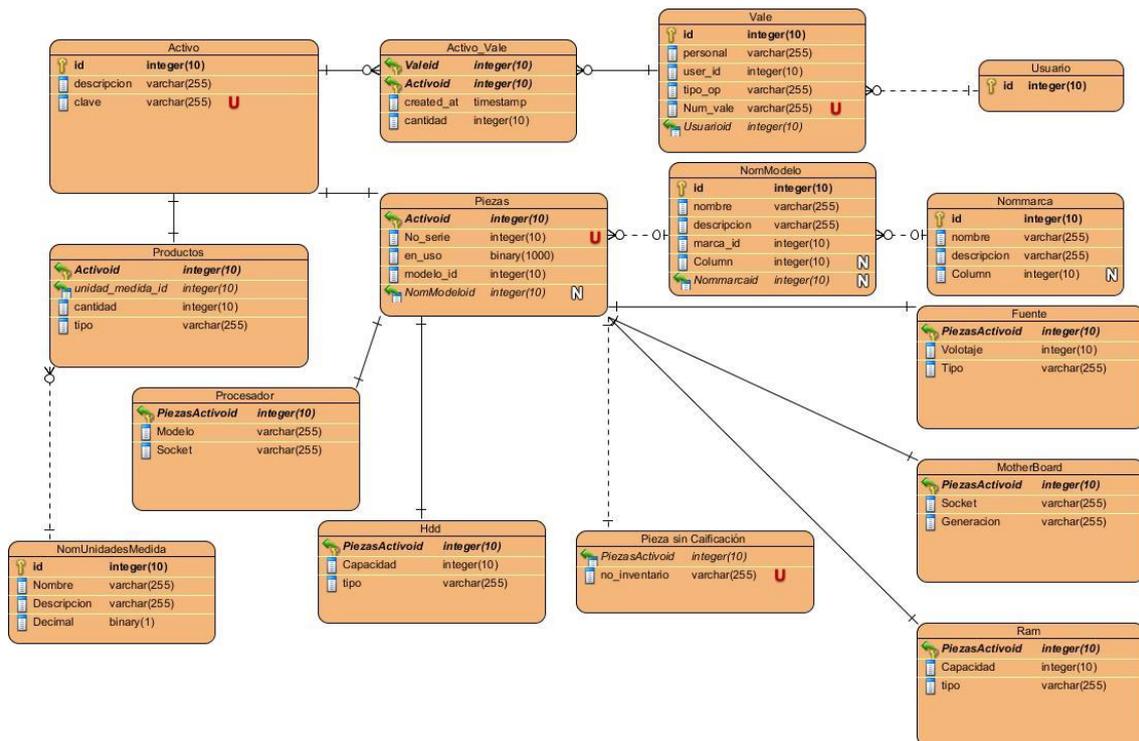


Figura 7 Modelo de Base de Datos. Fuente: Elaboración Propia

### 2.9.1 Descripción 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 “Activo”.

| Descripción: En esta tabla se agrupa la información correspondiente a los datos comunes de todas las piezas y productos |         |  |
|---|---------|--|
| Atributo  | Tipo    | Descripción                              |
| Id  | integer | Etiqueta única que identifica el objeto. |
| Descripción   | Varchar | Una descripción breve del objeto.        |
| Clave   | Varchar | Clave única para cada objeto.            |

Tabla 8: Descripción de la tabla Activo. Fuente: Elaboración propia.

### Conclusiones Parciales

La especificación de requisitos para el módulo a implementar se basó en 42 requisitos funcionales y 10 no funcionales, lo que permitió tener una idea clara de lo que se debía implementar. Además, los artefactos generados durante la etapa de análisis y diseño utilizando la metodología AUP en su variante UCI escenario número cuatro, permitieron establecer las bases necesarias para la implementación del módulo y definir de manera detallada las relaciones entre las entidades del sistema.

La investigación y definición de la arquitectura de software y los patrones de diseño a utilizar, contribuyeron a lograr una implementación que se centró en el diseño previamente realizado, al mismo tiempo el diseño de la base de datos definió la estructura que debe tener el sistema la cual determinó el conjunto de esquemas de relación con sus atributos, dominios de atributos, claves primarias, claves foráneas, entre otros.

## 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 Hdd:

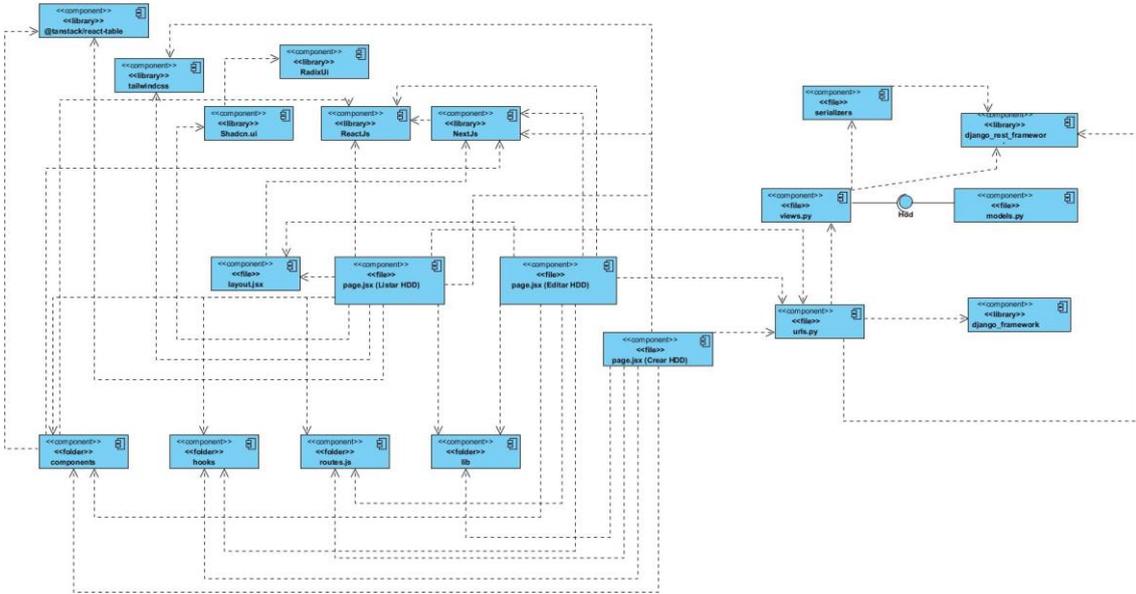


Figura 8 Diagrama de Componentes <Gestionar Hdd>. Fuente: Elaboración propia.

### 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.

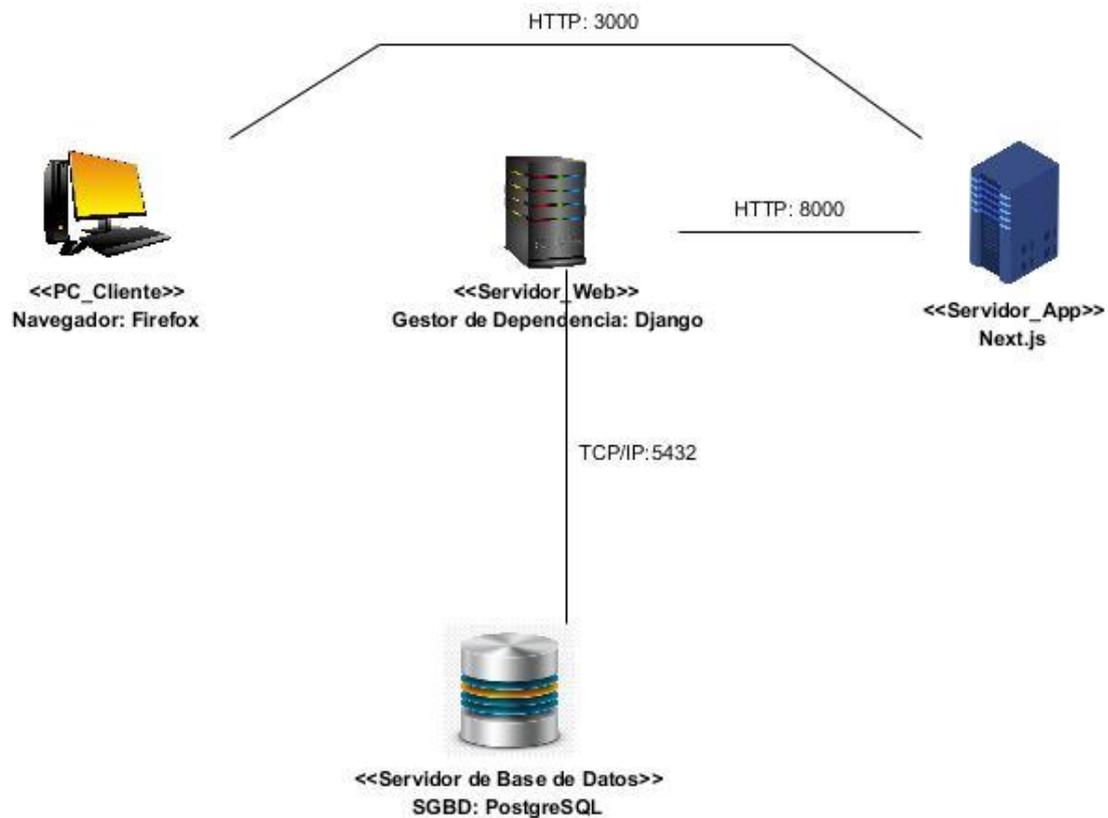


Figura 9 Diagrama de Despliegue. Fuente: Elaboración propia.

### 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 fuente, 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.

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

*Tabla 9 Estándares de Codificación. Fuente: Elaboración propia.*

### 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 posibles 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 son: Pruebas Unitarias, Pruebas de Aceptación, Pruebas de Integración y

Pruebas de Carga y Estrés. A continuación, se describen estos tipos de pruebas a través de los principales elementos y los resultados obtenidos en cada una de estas. (RS, 2007)

### 3.5 Niveles de Pruebas

| Etapa      | Nivel de Prueba              | Objetivo   |
|------------|------------------------------|--|
| Desarrollo | Prueba de Sistema Unitarias. | Las pruebas unitarias consisten en comprobar que un método concreto del código de producción funciona correctamente. |
| Despliegue | Prueba de Integración.       | Es un mecanismo de testeo de software, donde se realiza un análisis de los procesos de unión de los componentes.     |
| Despliegue | Prueba de Aceptación         | Esta prueba tiene como fin validar si el software satisface las necesidades del usuario final.                       |

Tabla 10 Niveles de Prueba. Fuente: Elaboración propia.

También se realizarán las pruebas de carga y estrés, aunque relacionadas con los **niveles de prueba** del software, generalmente no se clasifican directamente dentro de los niveles tradicionales de pruebas: unitarias, integración, sistema y aceptación. En cambio, estas pruebas forman parte de un conjunto más amplio de pruebas conocidas como Pruebas de Rendimiento. Estas pruebas se enfocan en evaluar cómo se comporta el sistema bajo ciertas condiciones de operación y se pueden realizar en varias etapas del desarrollo.

#### 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.

| Prueba  | Resultado dado por Unittest  | Resultado                     |
|---|--|-------------------------------|
| Agregar Hdd en base de datos exitosamente con sus datos validados | <pre> 107 class TestHddsModel(TestCase): 108 109     def setUp(self): 110         self.marca = NomMarca.objects.create(nombre="MarcaFuentesTest") 111         self.modelo = NomModelo.objects.create( 112             nombre="ModeloFuentesTest", 113             marca=self.marca 114         ) 115 116     def test_creation(self): 117         hdd = Hdds.objects.create( 118             descripcion="HDD de prueba", 119             clave="ClaveHDD123", 120             modelo=self.modelo, 121             no_serie="SerieHDD123", 122             capacidad=1000, 123             tipo=Hdds.Tipo.SATA, 124         ) 125         self.assertIsInstance(hdd, Hdds) </pre> <p>PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN <u>TERMINAL</u> PUERTOS GITLENS COMENTARIOS</p> <pre> (.venv) PS D:\SCHOOL\TESIS\backend&gt; python.exe .\manage.py test main.test.db.test_create Found 11 test(s). Creating test database for alias 'default'... System check identified no issues (0 silenced). ..... ----- Ran 11 tests in 0.422s  OK Destroying test database for alias 'default'... (.venv) PS D:\SCHOOL\TESIS\backend&gt; </pre> | Se agrega un Hdd exitosamente |

|   |  |                                     |
|---|--|-------------------------------------|
| <p>Editar Hdd en base de datos previamente creado con sus datos validados</p> | <pre> 49 class TestHddsModelEdit(TestCase): 50 51     def setUp(self): 52         self.marca = NomMarca.objects.create(nombre="MarcaFuentesTest") 53         self.modelo = NomModelo.objects.create( 54             nombre="ModeloTest", marca=self.marca) 55         self.hdd = Hdds.objects.create( 56             descripcion="HDD original", 57             clave="ClaveHDDOriginal", 58             modelo=self.modelo, 59             no_serie="SerieHDDOriginal", 60             capacidad=512, 61             tipo=Hdds.Tipo.SATA, 62         ) 63 64     def test_edit(self): 65         self.hdd.capacidad = 1024 66         self.hdd.save() 67         self.assertEqual(self.hdd.capacidad, 1024) </pre> <p>PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS GITLENS COMENTARIOS</p> <p>Destroying test database for alias 'default'...</p> <p>(.venv) PS D:\SCHOOL\TESIS\backend&gt; python.exe .\manage.py test main.test.db.test_edit</p> <p>Found 11 test(s).</p> <p>Creating test database for alias 'default'...</p> <p>System check identified no issues (0 silenced).</p> <p>.....</p> <p>-----</p> <p>Ran 11 tests in 0.400s</p> <p>OK</p> <p>Destroying test database for alias 'default'...</p> <p>(.venv) PS D:\SCHOOL\TESIS\backend&gt;  </p> | <p>Se edita un Hdd exitosamente</p> |
|---|--|-------------------------------------|

Tabla 11 Pruebas Unitarias. Fuente: Elaboración propia.



Figura 10 Resultado de las Pruebas Unitarias

### 3.5.2. Pruebas de Integración

Las pruebas de integración se centran en verificar cómo diferentes partes del sistema interactúan entre sí. El propósito es asegurarse de que todos los componentes encajen correctamente y trabajen juntos sin problemas. Estas pruebas son como verificar si todas las piezas de un reloj funcionan armoniosamente para mostrar la hora correcta. (Díaz, 2014)

En el caso de la solución desarrollada, para la verificación de una correcta interoperabilidad entre los módulos desarrollados, se llevaron a cabo las acciones siguientes:

- Integración del Módulo Gestión de partes y piezas para el Local de Piezas Recuperadas de la Dirección de Gestión Tecnológica de la UCI al Módulo Gestión de Configuración para el Sistema de Información de la Dirección de Gestión Tecnológica. Se muestra, en las Figuras 10 y 11, la ejecución de las de las pruebas de integración realizadas al sistema:

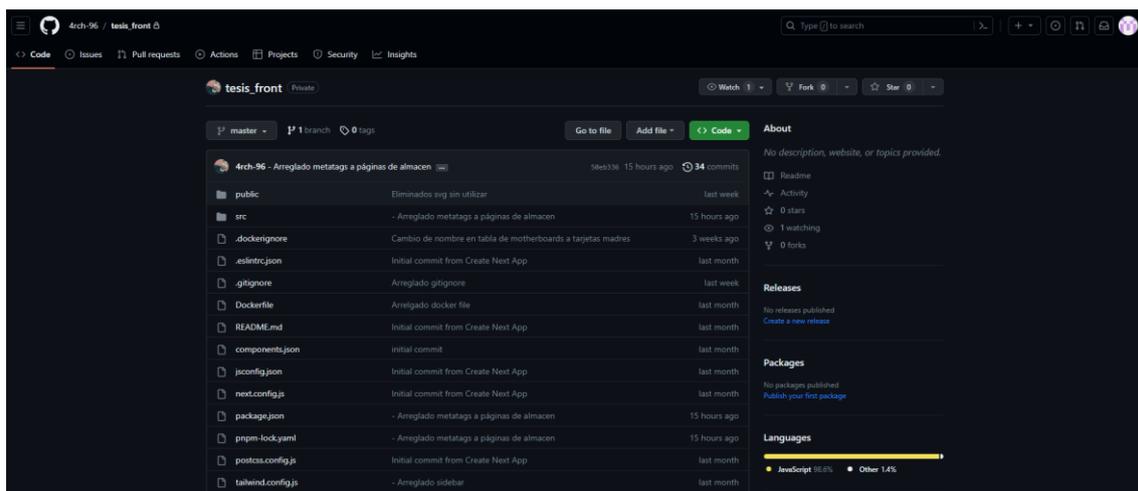


Figura 11 Pruebas de Integración - GitHub

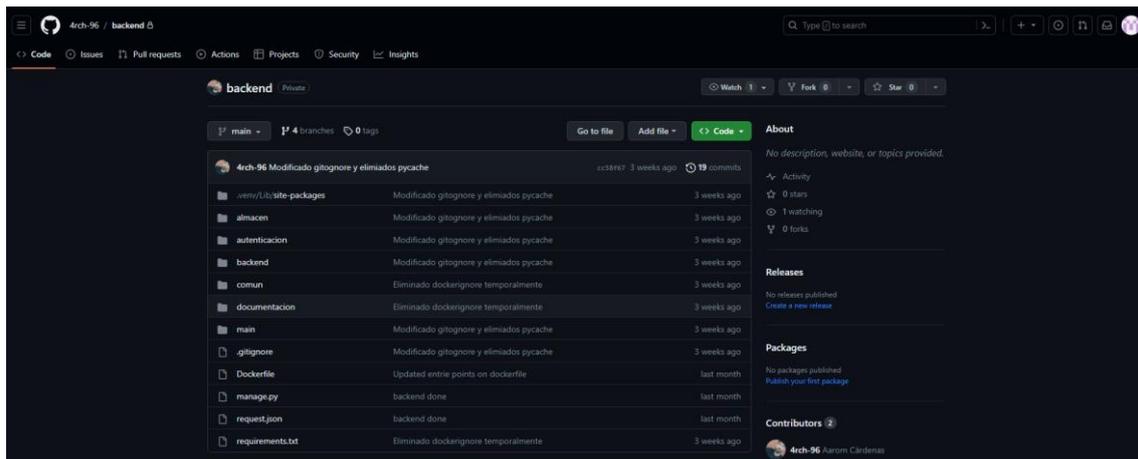


Figura 12 Pruebas de Integración - GitHub

Con las pruebas de integración realizadas se obtuvo un Sistema, en el cual ya se encuentran en funcionamiento el módulo Gestión de partes y piezas para el Local de Piezas Recuperadas de la DGT de la UCI y el Módulo Gestión de Configuración para el Sistema de Información de la Dirección de Gestión Tecnológica.

### 3.5.3. Pruebas de Aceptación

Son el nivel final de pruebas y suelen ser realizadas por el usuario final o el cliente. Estas pruebas se basan en escenarios y flujos de trabajo completos, su objetivo es asegurar que el sistema es funcional y apto para el uso en el entorno del usuario y que cumple con los requisitos y las expectativas del cliente.

|   |
|---|
| Caso de Prueba de Aceptación  |
| Número de Historia de Usuario: 1  |
| Nombre: : Gestionar HDD   |
| Descripción: Ofrece al usuario una vista que permite agregar, eliminar, editar y listar los HDD con su capacidad y tipo.  |
| Condiciones de Ejecución: <ul style="list-style-type: none"> <li>• El usuario ha iniciado sesión en el sistema.</li> <li>• El usuario tiene permisos para agregar, eliminar y editar discos duros.</li> </ul>   |
| Entrada/Pasos de Ejecución: <ul style="list-style-type: none"> <li>• <b>Paso 1:</b> Acceder a la pantalla del listado de discos duros<br/> Agregar un nuevo disco duro.<br/> En la pantalla de listado de discos duros, hacer clic en el botón "Agregar Disco Duro".<br/> Completar los siguientes campos:</li> </ul> |

|  |
|--|
| <p>Tipo de Disco Duro: [Especificar un tipo de disco duro válido, por ejemplo, "SSD"]</p> <p>Capacidad del Disco Duro: [Especificar una capacidad válida en GB, por ejemplo, "500GB"]</p> <p>Hacer clic en el botón "Agregar" para confirmar la adición del disco duro y verificar la adición del disco duro.</p> <ul style="list-style-type: none"> <li>• <b>Paso 2:</b> Editar un HDD existente.</li> </ul> <p>En la vista de gestión de HDD, identificar un HDD de la lista que se desea editar. Hacer clic en el botón "Editar" junto al HDD seleccionado.</p> <p>Modificar los detalles del HDD, como Tipo de Disco Duro y Capacidad del Disco. Hacer clic en el botón "Guardar" para confirmar los cambios.</p> <ul style="list-style-type: none"> <li>• <b>Paso 3:</b> Eliminar un HDD existente.</li> </ul> <p>En la vista de gestión de HDD, identificar un HDD de la lista que se desea eliminar. Hacer clic en el botón "Eliminar" junto al HDD seleccionado.</p> <p>Confirmar la eliminación en el cuadro de diálogo de confirmación.</p> <ul style="list-style-type: none"> <li>• <b>Paso 4:</b> Listar los HDD existentes.</li> </ul> <p>Verificar que la lista de HDD se muestra correctamente en la vista de gestión de HDD.</p> |
| <p>Resultado Esperado: El disco duro se agrega, elimina, edita y lista correctamente al sistema y los datos se muestren correctamente en la lista de discos duros.</p>   |
| <p>Evaluación de la Prueba: Satisfactoria</p>  |

Tabla 12 Prueba de aceptación <Gestionar Hdd>. Fuente: Elaboración propia.

|  |
|--|
| <p>Caso de Prueba de Aceptación</p>  |
| <p>Número de Historia de Usuario: 2</p>  |
| <p>Nombre: : Gestionar Fuentes</p>   |
| <p>Descripción: Ofrece al usuario una vista que permite agregar, eliminar, editar y listar las fuentes con su voltaje y tipo.</p>  |
| <p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> <li>• El usuario ha iniciado sesión en el sistema.</li> <li>• El usuario tiene permisos para agregar, eliminar y editar las fuentes.</li> </ul>  |
| <p>Entrada/Pasos de Ejecución:</p> <ul style="list-style-type: none"> <li>• <b>Paso 1:</b> Acceder a la pantalla del listado de las fuentes.</li> </ul> <p>Agregar una nueva fuente.</p> <p>En la pantalla de listado de las fuentes, hacer clic en el botón "Agregar Fuente".</p> |

|  |
|--|
| <p>Completar los siguientes campos: "Voltaje" y "Tipo".</p> <p>Hacer clic en el botón "Agregar" para confirmar la adición de la fuente y verificar la adición.</p> <ul style="list-style-type: none"> <li>• <b>Paso 2:</b> Editar una fuente existente.</li> </ul> <p>En la vista de gestión de Fuentes, identificar una fuente de la lista que se desea editar.</p> <p>Hacer clic en el botón "Editar" junto a la fuente seleccionada.</p> <p>Modificar los detalles de la fuente, como Tipo y Voltaje.</p> <p>Hacer clic en el botón "Guardar" para confirmar los cambios.</p> <ul style="list-style-type: none"> <li>• <b>Paso 3:</b> Eliminar una Fuente existente.</li> </ul> <p>En la vista de gestión de Fuentes, identificar una fuente de la lista que se desea eliminar.</p> <p>Hacer clic en el botón "Eliminar" junto a la Fuente seleccionada.</p> <p>Confirmar la eliminación en el cuadro de diálogo de confirmación.</p> <ul style="list-style-type: none"> <li>• <b>Paso 4:</b> Listar las fuentes existentes.</li> </ul> <p>Verificar que la lista de Fuentes se muestra correctamente en la vista de gestión.</p> |
| <p>Resultado Esperado: Las fuentes se agregan, eliminan, editan y listan correctamente al sistema y los datos se muestren correctamente en la lista de Fuentes.</p>  |
| <p>Evaluación de la Prueba: Satisfactoria</p>  |

Tabla 13 Prueba de aceptación <Gestionar Fuente>. Fuente: Elaboración propia.

|  |
|--|
| <p>Caso de Prueba de Aceptación</p>  |
| <p>Número de Historia de Usuario: 3</p>  |
| <p>Nombre: : Gestionar Producto</p>  |
| <p>Descripción: Ofrece al usuario una vista que permite agregar, eliminar, editar y listar los Productos con su cantidad, unidad de medida, tipo, No_serie, clave y descripción.</p>   |
| <p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> <li>• El usuario ha iniciado sesión en el sistema.</li> <li>• El usuario tiene permisos para agregar, eliminar y editar los productos.</li> </ul>  |
| <p>Entrada/Pasos de Ejecución:</p> <ul style="list-style-type: none"> <li>• <b>Paso 1:</b> Acceder a la pantalla del listado de los productos</li> </ul> <p>Agregar un nuevo producto.</p> <p>En la pantalla de listado de productos, hacer clic en el botón "Agregar producto".</p> <p>Completar los siguientes campos: cantidad, unidad de medida, tipo, No_serie, clave y descripción</p> |

Hacer clic en el botón "Agregar" para confirmar la adición del producto y verificar la adición.

- **Paso 2:** Editar un Producto existente.

En la vista de gestión de Productos, identificar un Producto de la lista que se desea editar.

Hacer clic en el botón "Editar" junto al Producto seleccionado.

Modificar los detalles del Producto.

Hacer clic en el botón "Guardar" para confirmar los cambios.

- **Paso 3:** Eliminar un Producto existente.

En la vista de gestión de Productos, identificar un Producto de la lista que se desea eliminar.

Hacer clic en el botón "Eliminar" junto al Producto seleccionado.

Confirmar la eliminación en el cuadro de diálogo de confirmación.

- **Paso 4:** Listar los Productos existentes.

Verificar que la lista de Productos se muestra correctamente en la vista.

Resultado Esperado: El producto se agrega, elimina, edita y lista correctamente al sistema y los datos se muestren correctamente en la lista.

Evaluación de la Prueba: Satisfactoria

Tabla 14 Prueba de aceptación <Gestionar Productos>. Fuente: Elaboración propia.

#### [Aval de aceptación del cliente](#)



Figura 13 Resultado de las Pruebas de Aceptación

### 3.5.4 Pruebas de Carga y Estrés

La prueba de carga y estrés son pruebas que se utilizan para determinar los límites del sistema, donde el objetivo es verificar la estabilidad y fiabilidad del sistema en condiciones extremas. Para la realización de esta prueba se utilizó la herramienta Apache JMeter, Las pruebas se realizaron desde un ordenador con 8GB de RAM,

microprocesador Intel Core i5 7th Gen, con 2.60 GHz y sistema operativo Windows 10. A continuación, se describen las variables que miden el resultado de las pruebas de carga y estrés realizadas al módulo:

**Muestra:** Cantidad de peticiones realizadas para cada URL.

**Media:** Tiempo promedio en milisegundos en el que se obtienen los resultados.

**Mediana:** Tiempo en milisegundos en el que se obtuvo el resultado que ocupa la posición central.

**Min:** Tiempo mínimo que demora un hilo en acceder a una página.

**Max:** Tiempo máximo que demora un hilo en acceder a una página.

**Línea 90 %:** Máximo tiempo utilizado por el 90 % de la muestra, al resto de la misma le llevo más tiempo.

**% Error:** Por ciento de error de las páginas que no se llegaron a cargar de manera satisfactoria.

**Rendimiento (Rend):** El rendimiento se mide en cantidad de solicitudes por segundo.

**Kb/s:** Velocidad de carga de las páginas.

Como se muestra en la siguiente tabla, se simularon las peticiones realizadas al módulo por un total de 50,100 y 150 usuarios simultáneamente en cada caso. Se obtuvieron los siguientes resultados:

| Usuarios | Muestras | Media | Mediana  | Min | Max   | Línea 90% | % Error | Rend | Kb/s   |
|----------|----------|-------|----------|-----|-------|-----------|---------|------|--------|
| 50       | 132      | 16503 | 16512.35 | 65  | 48466 | 2902      | 0.02%   | 5    | 6276.7 |
| 100      | 163      | 22779 | 18542.44 | 69  | 48763 | 2591      | 0.50%   | 10   | 5717.6 |
| 150      | 190      | 24769 | 20745.30 | 73  | 48935 | 2193      | 1.25%   | 15   | 5434.7 |

*Tabla 15 Pruebas de Carga y Estrés. Fuente: Elaboración propia.*

Las pruebas realizadas muestran que el Sistema es capaz de responder a 132 peticiones de 50 usuarios conectados simultáneamente en un tiempo promedio de 16503 milisegundos (1.6 segundos aproximadamente) con 0.02 % de error, esto evidencia que el sistema puede procesar la carga esperada. Por otra parte, se realizaron 163 peticiones iniciadas por 100 usuarios y en este caso el sistema respondió en 22779 milisegundos (2.2 segundos aproximadamente) como tiempo promedio. Esto demuestra que el sistema puede procesar la carga esperada, aunque no fue capaz de responder correctamente el 0.50% de las peticiones realizadas. Por último, y con el objetivo de analizar el comportamiento del sistema en condiciones extremas, se realizó una prueba

de estrés para un conjunto de 150 usuarios conectados simultáneamente. En este caso, el sistema responde a las 190 peticiones en un tiempo promedio de 24769 milisegundos (2.4 segundos aproximadamente), pero con un porcentaje de error de 1.25%. Este resultado está estrechamente relacionado al entorno donde se realizó la prueba, el cual no es un servidor dedicado sino un cliente habilitado para realizar la prueba.

Una vez finalizadas las pruebas de Carga y Estrés realizadas al Módulo Gestión de partes y piezas para el Local de Piezas Recuperadas de la Dirección de Gestión Tecnológica de la Universidad, se pueden concluir que el sistema actualmente soporta la interacción simultánea de varios usuarios al mismo tiempo, lo que se considera como un punto positivo que permitirá el correcto uso del sistema.

### 3.6 Resultados del MGPPLPR



Figura 14 Portada Inicial del Sistema

The screenshot shows the 'Nuevo Disco Duro' form. The title is 'Nuevo Disco Duro' with the subtitle 'Entre la información relacionada al Disco Duro'. The form contains the following fields:

- Descripción:** Text input field.
- Clave:** Text input field.
- No. Serie:** Text input field.
- Tipo:** Dropdown menu with 'SATA' selected.
- Capacidad:** Text input field with '0' entered.
- Modelo:** Dropdown menu.

At the bottom of the form, there are two buttons: 'ATRÁS' (red) and 'CREAR' (dark blue). At the bottom of the page, there is a copyright notice: '© 2023. creado para Direccion de Gestion Tecnologica para una mejor administración.' and a contact email: 'Contactenos: lesterr@uclcu'.

Figura 15 Vista Agregar Hdd

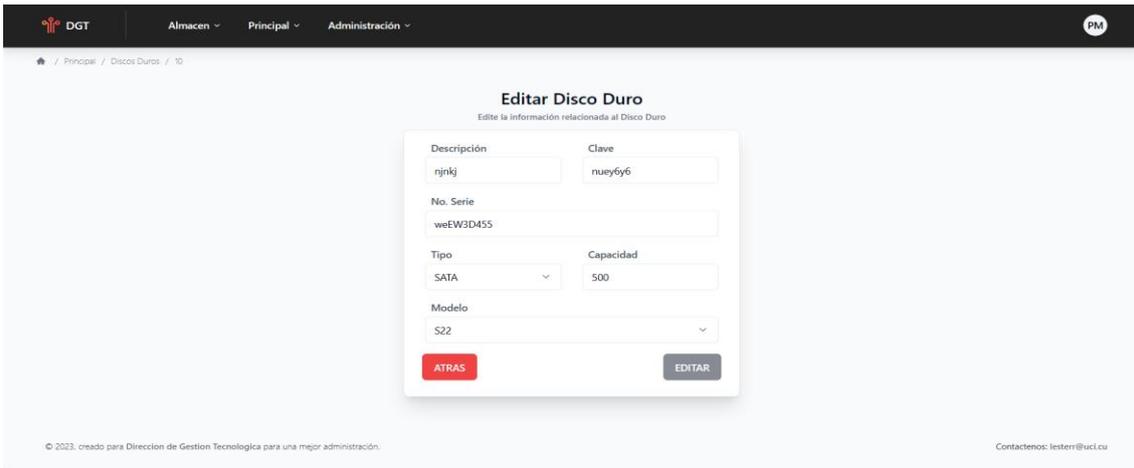


Figura 16 Vista Editar Hdd

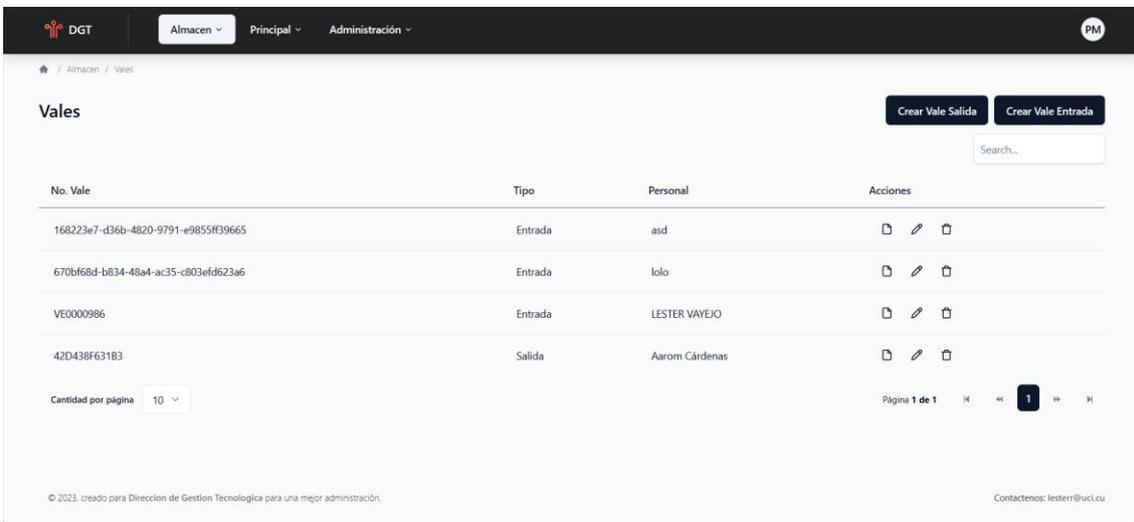


Figura 17 Vista general de los Vales

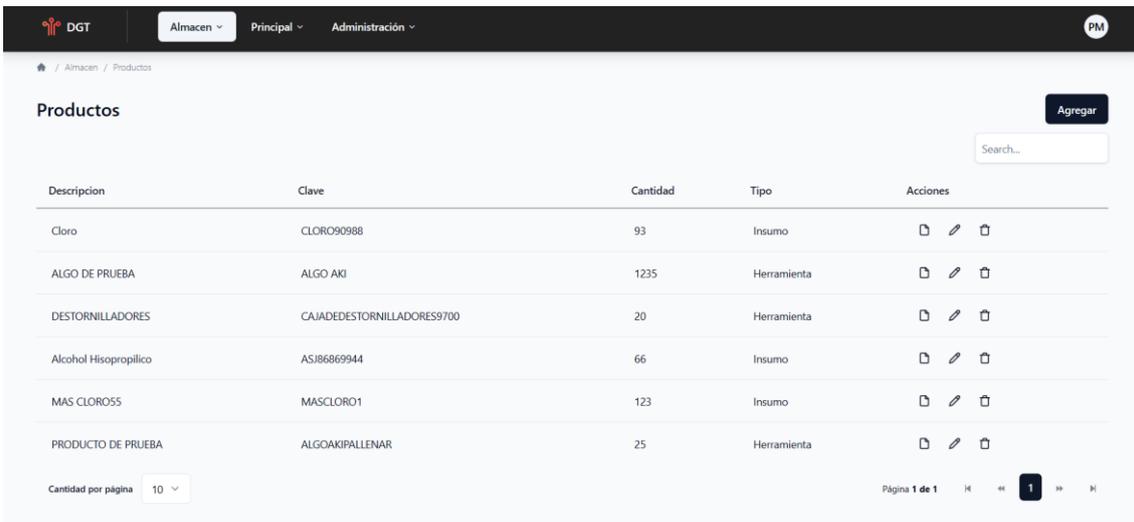


Figura 18 Vista Productos

## **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 como las unitarias y las de aceptación permitieron la obtención de un software que responde a las especificaciones del cliente y disminuyeron el riesgo de que falle en ejecución, se detectaron varias no conformidades que fueron resueltas de forma satisfactoria, logrando que la aplicación se ajuste a las expectativas del cliente.

## **Conclusiones Generales**

De manera general la presente investigación concluyó con la creación del módulo de Gestión de partes y piezas para el Local de Piezas Recuperadas de la Dirección de Gestión Tecnológica de la Universidad UCI integrado al Módulo Gestión de Configuración para el Sistema de Información de la Dirección de Gestión Tecnológica., sirviendo el mismo de apoyo al proceso de atención a las incidencias tecnológicas. Entre los principales aspectos significativos que se pueden recalcar del Módulo se encuentran:

- El estudio del estado del arte sobre el desarrollo de sistemas referentes al objeto de estudio permitió sentar las bases para el desarrollo de la investigación propuesta e identificar posibles puntos de reutilización con los sistemas semejantes estudiados.
- La utilización de la metodología AUP-UCI en su escenario número cuatro fue un factor de gran importancia para la organización del proceso de desarrollo del Módulo de Gestión de partes y piezas para el LPR de la DGT de la UCI
- La realización del análisis y diseño de la solución permitió realizar la implementación del sistema a partir de los artefactos ingenieriles obtenidos.
- Las estrategias de pruebas definidas, logró probar el correcto funcionamiento del Módulo, teniendo en cuenta para esto los requisitos definidos junto al cliente, además permitió concluir que el módulo desarrollado cumple 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- Agregar a la creación de los vales la funcionalidad de firma digital.
- 2- Desarrollar una Aplicación Android con funcionalidades equivalentes al módulo obtenido.

## 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](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). Ambyssoft.

<https://www.ambyssoft.com/unifiedprocess/agileUP.html>. [Online] 2013.

**Arismendiz, Cruz Delia Cabada. 2017.** Arquitectura Cliente - Servidor .

[https://www.academia.edu/35294361/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%2Ffcvsroot%2F eclipse%20CVS%20repository%3A%20Workbench%20User%20Guide%20-%20org.eclipse.platform.doc.user](https://wiki.eclipse.org/Eclipse_Documentation#:~:text=Eclipse%20Documentation%20This%20page%20is%20a%20hub%20for,dev.eclipse.org%3A%2Ffcvsroot%2F eclipse%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.

**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.

<http://www.it.uc3m.es/spickin/docencia/comsoft/presentations/spanish/doc/Python.pdf>.

[Online] julio 27, 2008.

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

**CLEMENTE, A. 2012.** *PROPUESTA DE ACCIONES PARA SOLUCIONAR LAS DEFICIENCIAS EN CUANTO AL CONTROL DE LOS ACTIVOS FIJOS TANGIBLES EN CUBA*. s.l. : Observatorio de la Economía Latinoamericana, no 169., 2012.

**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.

**Diaz, S. M. 2014.** *Pruebas de seguridad en aplicaciones web como imperativo en la calidad de desarrollo del software*. 2014.

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

**Figuerola, 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.

**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.* 2020.

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

[https://docs.oracle.com/cd/E18727\\_01/doc.121/e13671.pdf](https://docs.oracle.com/cd/E18727_01/doc.121/e13671.pdf). [Online]

<https://www.guiaserviciosproductos.com/analisis-productos-mercado/sap-ventajas-y-desventajas.php>. [Online]

<https://www.ibm.com/es-es/products/maximo>. [Online]

<https://www.oracle.com/a/ocom/docs/applications/ebusiness/oracle-enterprise-asset-management-data-sheet.pdf>. [Online]

*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. [En línea] . [Online] agosto 31, 2011.

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

**Kent Beck, Cynthia Andres. 2004.** *Extreme Programming Explained: Embrace Change (2nd ed.). Addison-Wesley.* 2004.

**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.

**Maximo, IBM. 2023.** <https://www.ibm.com/es-es/products/maximo/asset-management>. [Online] 2023.

**MDN.** [https://developer.mozilla.org/es/docs/Learn/CSS/First\\_steps/Why\\_use\\_CSS](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.*
- Nekaka, Pedro Massengo. 2020.** Sistema de Gestión de Estudiantes para el Sector de Apoyo Estudiantil de Angola en Cuba. [Online] 2020.
- 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.).
- ORACLE. 2023.** <https://www.oracle.com/utilities/work-asset-management/>. [Online] 2023.
- Pressman, Roger. 2015.** *Ingeniería del Software: Un enfoque práctico.* 2015.
- REST. 2011.** <https://www.django-rest-framework.org/>. [Online] 2011.
- . **2011.** <https://www.django-rest-framework.org/>. [Online] 2011.
- 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.
- SAP. 2023.** Gestión de activos empresariales (EAM). <https://www.sap.com/products/scm/asset-management-eam.html>. [Online] 2023.
- Simmons, Eary. 2019.** [https://docs.oracle.com/cd/E18727\\_01/doc.121/e13671.pdf](https://docs.oracle.com/cd/E18727_01/doc.121/e13671.pdf). [Online] 2019.
- 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](https://www.w3schools.com/html/html_intro.asp). [Online]

# Anexos

## Diagramas de clase

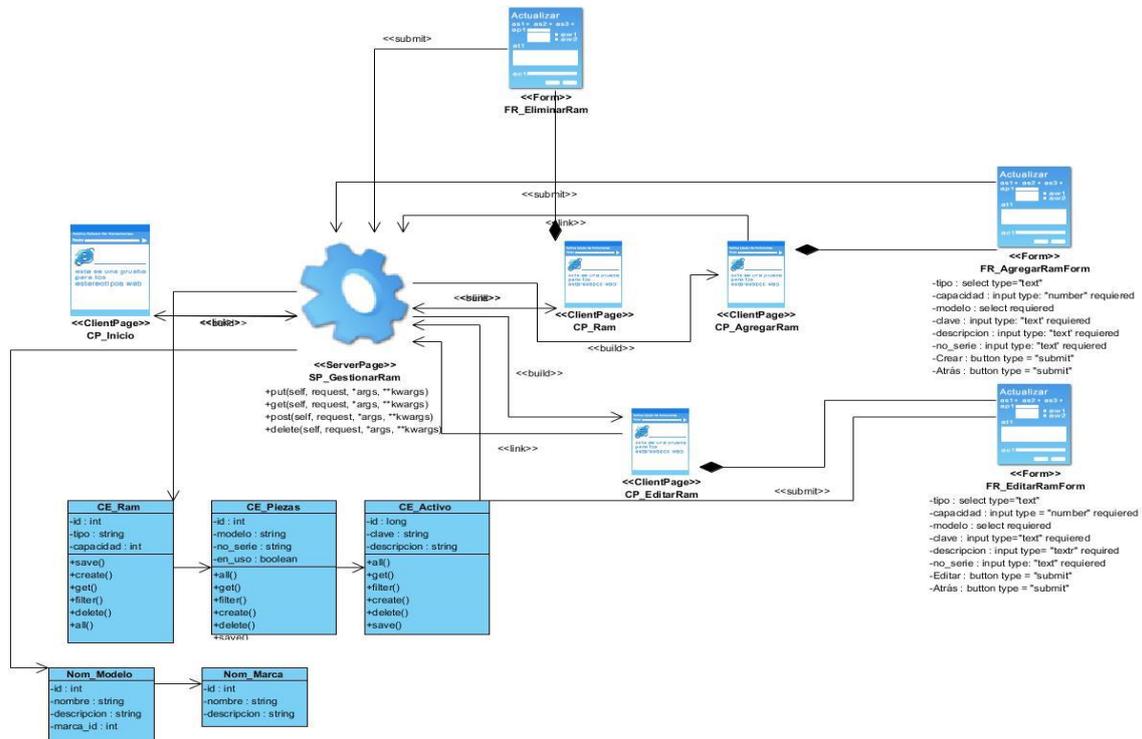


Figura 19 Gestionar Ram. Anexo

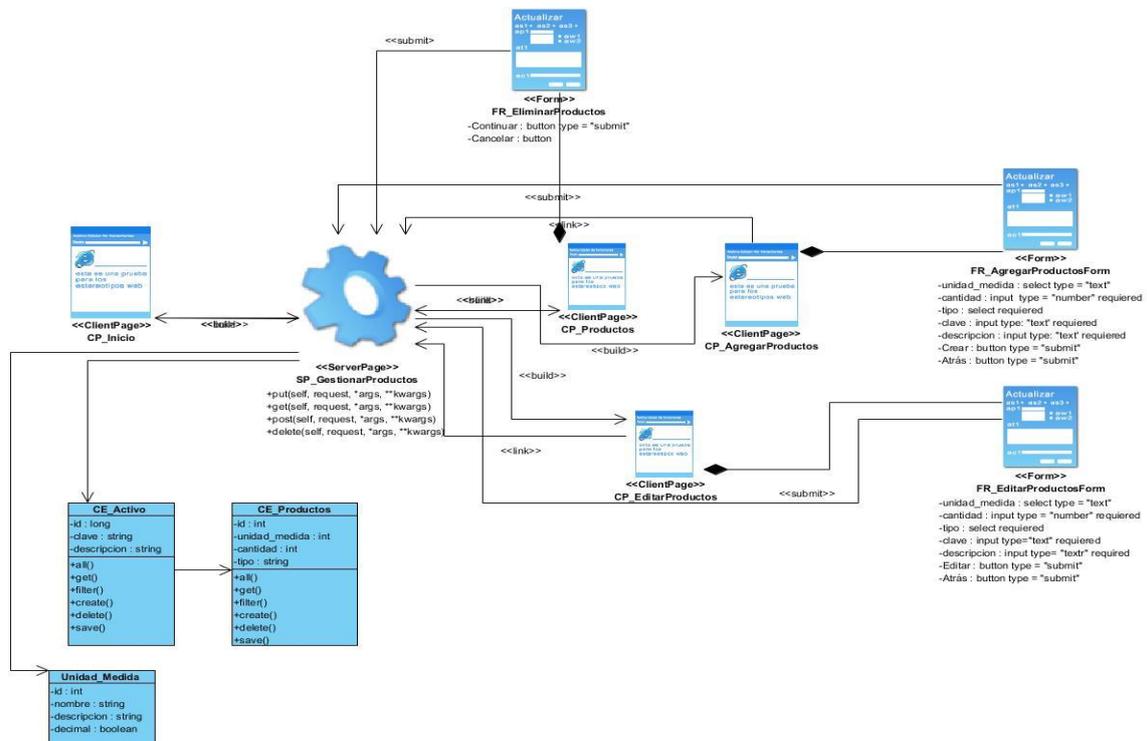


Figura 20 Gestionar Productos. Anexo

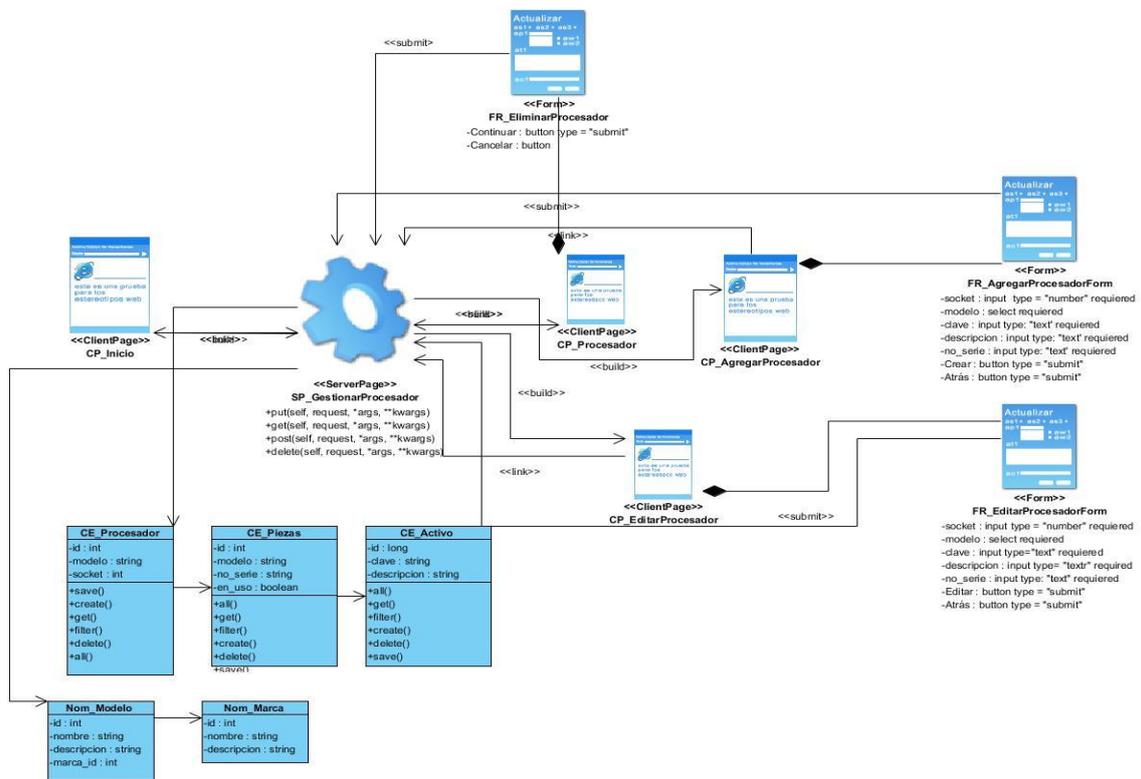


Figura 21 Gestionar Procesador. Anexo

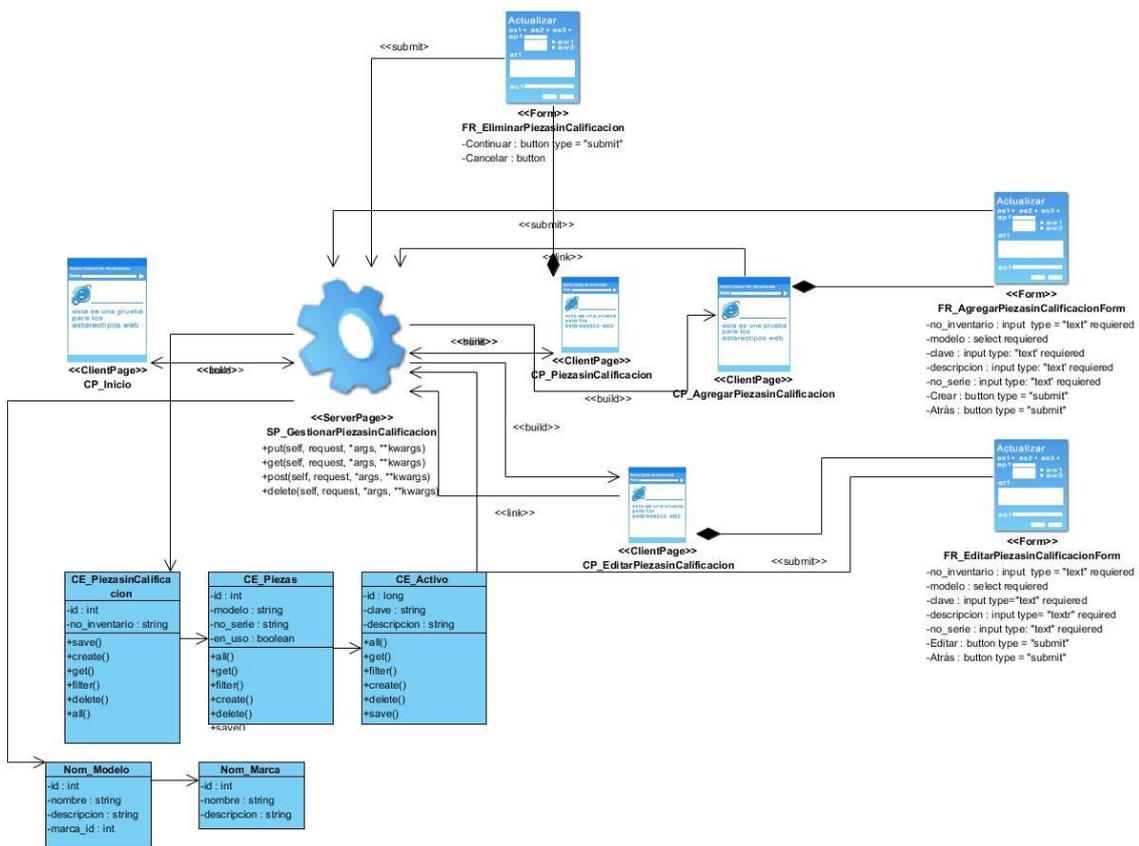


Figura 22 Gestionar Pieza sin Calificación. Anexo

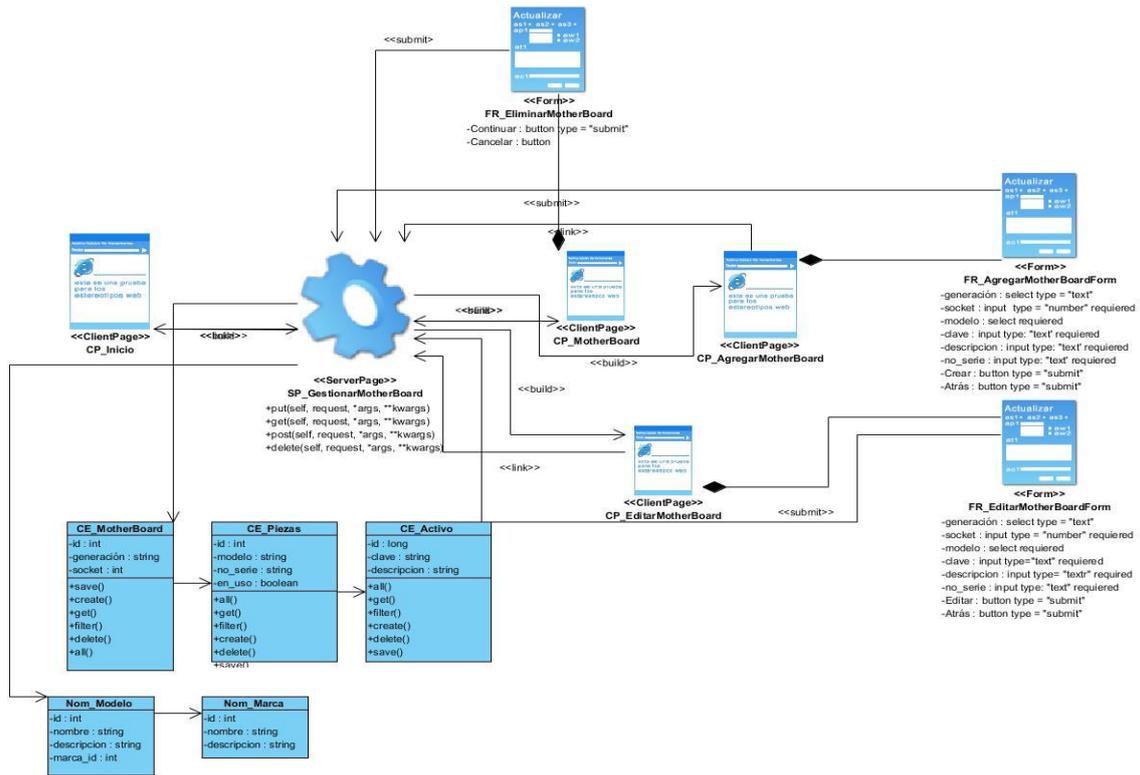


Figura 23 Gestionar MotherBoard. Anexo

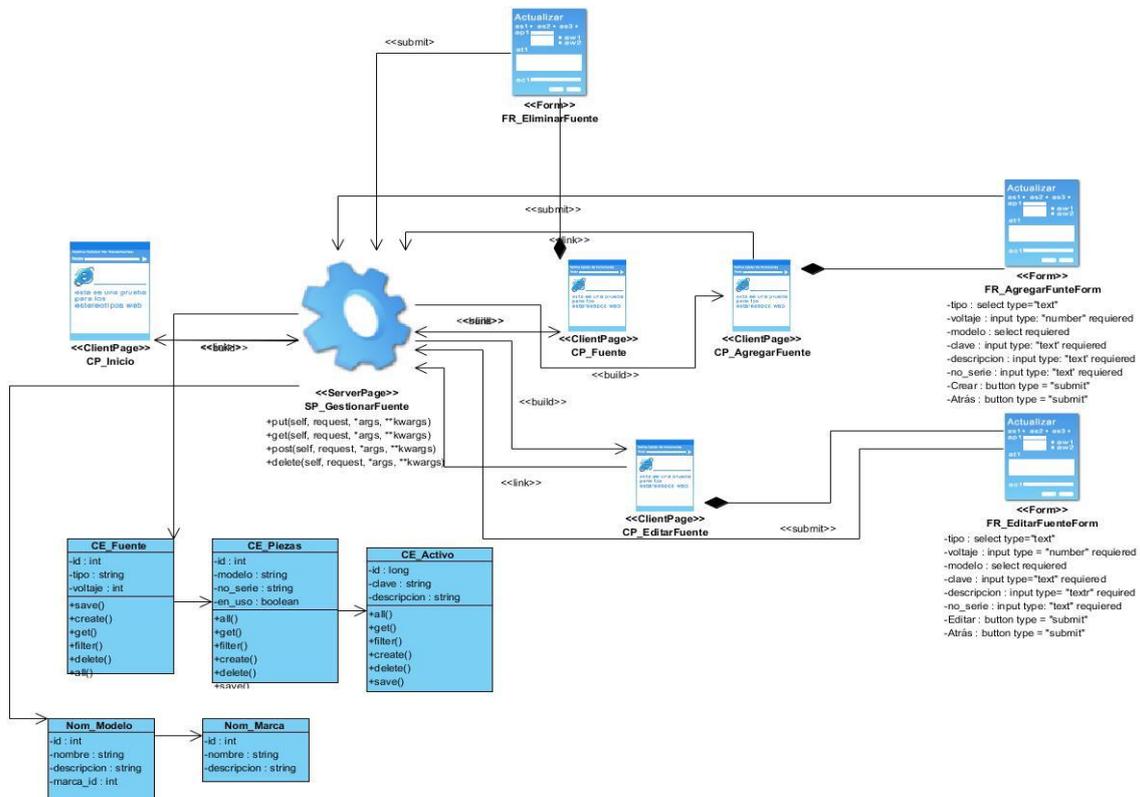


Figura 24 Gestionar Fuente. Anexo

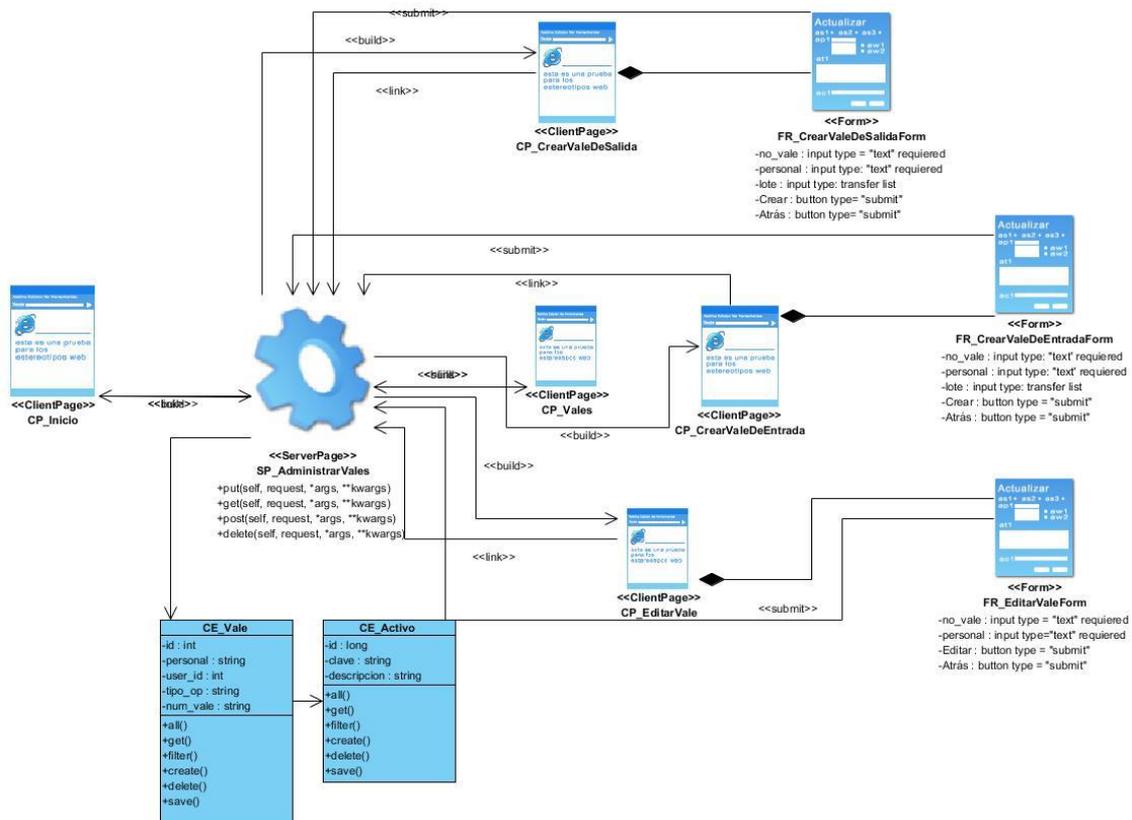


Figura 25 Administrar Vales. Anexo

## Historias de Usuario

| Historia de usuario  |  |
|--|--|
| Número:  | Nombre de la HU: Gestionar MotherBoard |
| Programador: Beatriz Pérez Moreno  | Iteración Asignada: 1era               |
| Prioridad: Alta  | Tiempo Estimado: 7 días                |
| Riesgo en Desarrollo: Alto   | Tiempo Real: 4 días                    |
| <p><b>1. Objetivo:</b></p> <p>- Ofrece al usuario una vista que permite agregar, eliminar, editar y listar las MotherBoard con su socket, generación, No_serie, marca, modelo, clave, descripción.</p> <p><b>2. Flujo de la acción a realizar:</b></p> <p>-El sistema debe permitir agregar, eliminar, editar y listar la/las MotherBoard con su socket, generación, No_serie, marca, modelo, clave, descripción.</p> <p><b>3. Flujo alternativo</b></p> |  |

- El sistema debe permitir ir “Atrás” o “Cancelar” en caso de que la opción seleccionada no sea la deseada.

Tabla 16 Historia de Usuario <Gestionar MotherBoard>. Fuente: Elaboración propia. Anexo

| Historia de usuario   |                                       |
|---|---------------------------------------|
| <b>Número:</b>  | <b>Nombre de la HU:</b> Gestionar Ram |
| <b>Programador:</b> Beatriz Pérez Moreno  | <b>Iteración Asignada:</b> 1era       |
| <b>Prioridad:</b> Alta  | <b>Tiempo Estimado:</b> 7 días        |
| <b>Riesgo en Desarrollo:</b> Alto   | <b>Tiempo Real:</b> 3 días            |
| <p><b>1. Objetivo:</b></p> <p>- Ofrece al usuario una vista que permite agregar, eliminar, editar y listar las Ram con su capacidad, tipo, pines, No_serie, marca, modelo, clave, descripción.</p> <p><b>2. Flujo de la acción a realizar:</b></p> <p>-El sistema debe permitir agregar, eliminar, editar y listar la/las Ram con su capacidad, tipo, pines, No_serie, marca, modelo, clave, descripción,</p> <p><b>3. Flujo alternativo</b></p> <p>- El sistema debe permitir ir “Atrás” o “Cancelar” en caso de que la opción seleccionada no sea la deseada.</p> |                                       |

Tabla 17 Historia de Usuario <Gestionar Ram>. Fuente: Elaboración propia. Anexo

| Historia de usuario                      |  |
|--|--|
| <b>Número:</b>                           | <b>Nombre de la HU:</b> Gestionar Procesador |
| <b>Programador:</b> Beatriz Pérez Moreno | <b>Iteración Asignada:</b> 1era              |
| <b>Prioridad:</b> Alta                   | <b>Tiempo Estimado:</b> 6 días               |
| <b>Riesgo en Desarrollo:</b> Alto        | <b>Tiempo Real:</b> 3 días                   |

### 1. Objetivo:

- Ofrece al usuario una vista que permite agregar, eliminar, editar y listar el/los Procesadores con su modelo, socket, No\_serie, marca, modelo, clave, descripción.

### 2. Flujo de la acción a realizar:

-El sistema debe permitir agregar, eliminar, editar y listar el/los Procesadores con su modelo, socket, No\_serie, marca, modelo, clave, descripción.

### 3. Flujo alternativo

- El sistema debe permitir ir "Atrás" o "Cancelar" en caso de que la opción seleccionada no sea la deseada.

Tabla 18 Historia de Usuario <Gestionar Procesador>. Fuente: Elaboración propia. Anexo

| Historia de usuario  |  |
|--|--|
| <b>Número:</b>   | <b>Nombre de la HU:</b> Gestionar Pieza sin Calificación |
| <b>Programador:</b> Beatriz Pérez Moreno   | <b>Iteración Asignada:</b> 1era                          |
| <b>Prioridad:</b> Alta   | <b>Tiempo Estimado:</b> 6 días                           |
| <b>Riesgo en Desarrollo:</b> Alto  | <b>Tiempo Real:</b> 5 días                               |
| <h3>1. Objetivo:</h3> <p>- Ofrece al usuario una vista que permite agregar, eliminar, editar y listar la/las Pieza sin Calificación con su No_serie, modelo, clave, descripción, no_inventario.</p>      |  |
| <h3>2. Flujo de la acción a realizar:</h3> <p>-El sistema debe permitir agregar, eliminar, editar y listar la/las Pieza sin Calificación con su No_serie, modelo, clave, descripción, no_inventario.</p> |  |
| <h3>3. Flujo alternativo</h3> <p>- El sistema debe permitir ir "Atrás" o "Cancelar" en caso de que la opción seleccionada no sea la deseada.</p>   |  |

Tabla 19 Historia de Usuario <Gestionar Pieza sin Calificación >. Fuente: Elaboración propia. Anexo

| Historia de usuario                      |   |
|--|---|
| <b>Número:</b>                           | <b>Nombre de la HU:</b> Administrar Vales |
| <b>Programador:</b> Beatriz Pérez Moreno | <b>Iteración Asignada:</b> 1era           |
| <b>Prioridad:</b> Alta                   | <b>Tiempo Estimado:</b> 10 días           |
| <b>Riesgo en Desarrollo:</b> Alto        | <b>Tiempo Real:</b> 13 días               |

### 1. Objetivo:

- Ofrece al usuario una vista que permite Crear, editar y listar el/los Vales tanto de entrada como de salida con su No\_vale y personal.

### 2. Flujo de la acción a realizar:

-El sistema debe permitir agregar, editar y listar l/los Vales tanto de entrada como de salida con su No\_vale y personal.

### 3. Flujo alterno

- El sistema debe permitir ir "Atrás" o "Cancelar" en caso de que la opción seleccionada no sea la deseada.

*Tabla 20 Historia de Usuario <Administrar Vales >. Fuente: Elaboración propia. Anexo*

## Pruebas de Aceptación

|  |
|--|
| Caso de Prueba de Aceptación   |
| Número de Historia de Usuario: 4   |
| Nombre: Gestionar MotherBoard  |
| Descripción: Ofrece al usuario una vista que permite agregar, eliminar, editar y listar las MotherBoard con su socket, generación, No_serie, marca, modelo, clave, descripción.  |
| Condiciones de Ejecución: <ul style="list-style-type: none"><li>• El usuario ha iniciado sesión en el sistema.</li><li>• El usuario tiene permisos para agregar, eliminar y editar las MotherBoard.</li></ul>  |
| Entrada/Pasos de Ejecución: <ul style="list-style-type: none"><li>• <b>Paso 1:</b> Acceder a la pantalla del listado de las MotherBoard<br/>Agregar una nueva MotherBoard.<br/>En la pantalla de listado de MotherBoard, hacer clic en el botón "Agregar MotherBoard".<br/>Completar los siguientes campos: socket, generación, No_serie, marca, modelo, clave, descripción.<br/>Hacer clic en el botón "Agregar" para confirmar la adición de la MotherBoard y verificar la adición.</li><li>• <b>Paso 2:</b> Editar una MotherBoard existente.<br/>En la vista de gestión de MotherBoard, identificar una MotherBoard de la lista que se desea editar.</li></ul> |

|  |
|--|
| <p>Hacer clic en el botón "Editar" junto a la MotherBoard seleccionada.</p> <p>Modificar los detalles de la MotherBoard.</p> <p>Hacer clic en el botón "Guardar" para confirmar los cambios.</p> <ul style="list-style-type: none"> <li>• <b>Paso 3:</b> Eliminar una MotherBoard existente.</li> </ul> <p>En la vista de gestión de MotherBoard, identificar una MotherBoard de la lista que se desea eliminar.</p> <p>Hacer clic en el botón "Eliminar" junto a la MotherBoard seleccionada.</p> <p>Confirmar la eliminación en el cuadro de diálogo de confirmación.</p> <ul style="list-style-type: none"> <li>• <b>Paso 4:</b> Listar las MotherBoard existentes.</li> </ul> <p>Verificar que la lista de las MotherBoard se muestra correctamente en la vista.</p> |
| <p>Resultado Esperado: Las MotherBoard se agregan, eliminan, editan y listan correctamente al sistema y los datos se muestren correctamente en la lista.</p>   |
| <p>Evaluación de la Prueba: Satisfactoria</p>  |

*Tabla 21 Prueba de aceptación <Gestionar MotherBoard>. Fuente: Elaboración propia. Anexo*

|   |
|---|
| <p>Caso de Prueba de Aceptación</p>   |
| <p>Número de Historia de Usuario: 5</p>   |
| <p>Nombre: Gestionar Ram</p>  |
| <p>Descripción: Ofrece al usuario una vista que permite agregar, eliminar, editar y listar las Ram con su capacidad, tipo, pines, No_serie, marca, modelo, clave, descripción.</p>  |
| <p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> <li>• El usuario ha iniciado sesión en el sistema.</li> <li>• El usuario tiene permisos para agregar, eliminar y editar las Ram.</li> </ul>   |
| <p>Entrada/Pasos de Ejecución:</p> <ul style="list-style-type: none"> <li>• <b>Paso 1:</b> Acceder a la pantalla del listado de las Ram.</li> <li>• Agregar una nueva Ram.</li> <li>• En la pantalla de listado de Ram., hacer clic en el botón "Agregar Ram ".<br/>Completar los siguientes campos: capacidad, tipo, pines, No_serie, marca, modelo, clave, descripción.<br/>Hacer clic en el botón "Agregar" para confirmar la adición de la Ram y verificar la adición.</li> <li>• <b>Paso 2:</b> Editar una Ram existente.</li> </ul> <p>En la vista de gestión de Ram, identificar una Ram de la lista que se desea editar.</p> <p>Hacer clic en el botón "Editar" junto a la Ram seleccionada.</p> <p>Modificar los detalles de la Ram.</p> <p>Hacer clic en el botón "Guardar" para confirmar los cambios.</p> |

- **Paso 3:** Eliminar una Ram existente.

En la vista de gestión de Ram, identificar una Ram de la lista que se desea eliminar.

Hacer clic en el botón "Eliminar" junto a la Ram seleccionada.

Confirmar la eliminación en el cuadro de diálogo de confirmación.

- **Paso 4:** Listar las Ram existentes.

Verificar que la lista de las Ram se muestra correctamente en la vista.

Resultado Esperado: Las Ram se agregan, eliminan, editan y listan correctamente al sistema y los datos se muestren correctamente en la lista.

Evaluación de la Prueba: Satisfactoria

*Tabla 22 Prueba de aceptación <Gestionar Ram>. Fuente: Elaboración propia. Anexo*

|  |
|--|
| Caso de Prueba de Aceptación   |
| Número de Historia de Usuario: 3   |
| Nombre: Gestionar Procesador   |
| Descripción: Ofrece al usuario una vista que permite agregar, eliminar, editar y listar los Procesadores con su modelo, socket, No_serie, marca, modelo, clave, descripción.   |
| Condiciones de Ejecución: <ul style="list-style-type: none"> <li>• El usuario ha iniciado sesión en el sistema.</li> <li>• El usuario tiene permisos para agregar, eliminar y editar los Procesadores.</li> </ul>  |
| Entrada/Pasos de Ejecución: <ul style="list-style-type: none"> <li>• <b>Paso 1:</b> Acceder a la pantalla del listado de los Procesadores<br/> Agregar un nuevo Procesadores.<br/> En la pantalla de listado de Procesadores, hacer clic en el botón "Agregar Procesador".<br/> Completar los siguientes campos: modelo, socket, No_serie, marca, modelo, clave, descripción.<br/> Hacer clic en el botón "Agregar" para confirmar la adición del procesador y verificar la adición.</li> <li>• <b>Paso 2:</b> Editar un Procesador existente.<br/> En la vista de gestión de Procesadores, identificar un Procesador de la lista que se desea editar.<br/> Hacer clic en el botón "Editar" junto al Procesador seleccionado.<br/> Modificar los detalles del Procesador.<br/> Hacer clic en el botón "Guardar" para confirmar los cambios.</li> </ul> |

- **Paso 3:** Eliminar un Procesador existente.

En la vista de gestión de Procesador, identificar un Procesador de la lista que se desea eliminar.

Hacer clic en el botón "Eliminar" junto al Procesador seleccionado.

Confirmar la eliminación en el cuadro de diálogo de confirmación.

- **Paso 4:** Listar los Procesadores existentes.

Verificar que la lista de Procesadores se muestra correctamente en la vista.

Resultado Esperado: El Procesador se agrega, elimina, edita y lista correctamente al sistema y los datos se muestren correctamente en la lista.

Evaluación de la Prueba: Satisfactoria

*Tabla 23 Prueba de aceptación <Gestionar Procesador>. Fuente: Elaboración propia. Anexo*

Caso de Prueba de Aceptación

Número de Historia de Usuario: 5

Nombre: Gestionar Pieza sin Calificación

Descripción: Ofrece al usuario una vista que permite agregar, eliminar, editar y listar las Piezas sin Calificación con su No\_serie, modelo, clave, descripción, no\_inventario.

Condiciones de Ejecución:

- El usuario ha iniciado sesión en el sistema.
- El usuario tiene permisos para agregar, eliminar y editar las Piezas sin Calificación.

Entrada/Pasos de Ejecución:

- **Paso 1:** Acceder a la pantalla del listado de las Piezas sin Calificación.
- Agregar una nueva Piezas sin Calificación.
- En la pantalla de listado de Piezas sin Calificación, hacer clic en el botón "Agregar Pieza sin Calificación".

Completar los siguientes campos: No\_serie, modelo, clave, descripción, no\_inventario.

Hacer clic en el botón "Agregar" para confirmar la adición de la Pieza sin Calificación y verificar la adición.

- **Paso 2:** Editar una Pieza sin Calificación existente.

En la vista de gestión de Piezas sin Calificación, identificar una pieza de la lista que se desea editar.

Hacer clic en el botón "Editar" junto a la Pieza sin Calificación seleccionada.

Modificar los detalles de la Pieza sin Calificación.

Hacer clic en el botón "Guardar" para confirmar los cambios.

- **Paso 3:** Eliminar una Pieza sin Calificación existente.

En la vista de gestión de Piezas sin Calificación, identificar una Pieza sin Calificación de la lista que se desea eliminar.

Hacer clic en el botón "Eliminar" junto a la Pieza sin Calificación seleccionada.

Confirmar la eliminación en el cuadro de diálogo de confirmación.

- **Paso 4:** Listar las Piezas sin Calificación existentes.

Verificar que la lista de las Piezas sin Calificación se muestra correctamente en la vista.

Resultado Esperado: Las Piezas sin Calificación se agregan, eliminan, editan y listan correctamente al sistema y los datos se muestren correctamente en la lista.

Evaluación de la Prueba: Satisfactoria

*Tabla 24 Prueba de aceptación <Gestionar Pieza sin Calificación>. Fuente: Elaboración propia. Anexo*

## Aval de aceptación del cliente



DIRECCIÓN DE GESTIÓN DE TECNOLOGÍA

La Habana, 01 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 la estudiante Beatriz Pérez Moreno.

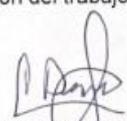
En la Dirección de Gestión Tecnológica se desarrolla un sistema informático para llevar a cabo la gestión de la información que generan los procesos del área. Hasta el momento el sistema no gestionaba la información que se maneja en los procesos llevados a cabo en el Local de Piezas Recuperadas (LPR) razón por la cuál fue necesario desarrollar el Módulo Gestión de partes y piezas para dicho local.

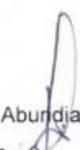
El Módulo de Gestión de partes y piezas desarrollado para el Local de Piezas Recuperadas permite un registro preciso y en tiempo real de las partes, piezas y herramientas disponibles, proporciona acceso en línea al inventario para que los trabajadores de la DGT puedan verificar la disponibilidad de manera remota. Este sistema permitió modernizar y optimizar el proceso de entrega y devolución de partes, piezas y herramientas, mejorando la eficiencia operativa y reduciendo los riesgos asociados con los procedimientos que ejecutamos de manera manual hasta la fecha.

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 en el LPR.

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

  
Lester Rodríguez Vallejo  
Director

  
José David Rodríguez Vargas  
Especialista Superior de  
control Interno

  
Miriam Abundia Palenzuela  
Martínez  
Responsable Local de  
Piezas Recuperadas