

6/15/2023



Universidad de las Ciencias
Informáticas

FACULTAD DE CIENCIAS Y TECNOLOGÍAS COMPUTACIONALES

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

**"Sistema Web para la gestión y control de los medios
informáticos en la Empresa de Acueducto y
Alcantarillado de Matanzas (UEB) Calimete"**

Autor: Raúl Noda Baró

Tutores:

Ing. C Dariel Lazo Tamayo

Ing. C Kimandy Díaz Hernández

La Habana, junio 2023.

"Año 65 de la Revolución"



“Los seres humanos sin conocimientos, se asemejan a un meteoro viajando sin rumbo en el espacio sideral.”

Sidhu

Declaración de autoría

Declaro ser el único autor del trabajo de diploma “Sistema Web para la gestión y control de los medios informáticos en la Empresa de Acueducto y Alcantarillado de Matanzas (UEB) Calimete”, concedo a la Universidad de las Ciencias Informáticas y en especial a la Dirección de Educación de Posgrado los derechos patrimoniales, con carácter exclusivo.

Para que conste firmo el presente documento a los ____ días del mes de _____ del año 20____.

Firma del autor

Tutor

Tutor

Firma del tutor

Firma del tutor

Datos de contacto

Autores:

Autor: Raúl Noda Baró

Correo: rnoda@estudiantes.uci.cu

Tutores:

Tutor: Ing. C Dariel Lazo Tamayo

Correo: dlazo@xetid.cu

Tutor: Ing. C Kimandy Diaz Hernandez

Correo: kdhernandez@uci.cu

DEDICATORIA

Le dedico mi tesis a mi familia por siempre creer en mí y siempre estar para mí cuando los necesite, a mis amigos que siempre me apoyaron.

Resumen

El proyecto titulado "Sistema Web para la gestión y control de los medios informáticos en la Empresa de Acueducto y Alcantarillado de Matanzas (UEB) Calimete" tiene como objetivo desarrollar una aplicación que permita administrar y controlar eficientemente los medios informáticos en la UEB Calimete. Se presentan los resultados de la investigación, que incluye un exhaustivo estudio de los conceptos fundamentales relacionados con la gestión de la información. Se realiza un análisis de diversas herramientas, tecnologías y lenguajes que son necesarios para el desarrollo del sistema, tales como son HTML, JavaScript, CSS y herramientas necesarias para su modelado y diseño como Visual Code Studio, XAMPP, MySQL, entre otros. Además, se implementa una base de datos y una plataforma web acorde con los requisitos del Sistema Web para la administración y control de los medios informáticos en la UEB Calimete.

Se llevan a cabo pruebas exhaustivas para verificar el cumplimiento de los requisitos establecidos, y se documentan los resultados que validan el correcto funcionamiento del sistema. Por último, se brindan recomendaciones para mejorar el sistema y aprovechar de manera óptima el trabajo realizado.

La propuesta de solución proporciona a la UEB Calimete un sistema de gestión que permitirá un control eficaz de la información relacionada con los medios informáticos, mejorando así la gestión de los medios en la institución.

Palabras clave: **sistema web, administración, control, medios informáticos, UEB Calimete.**

Abstract

The project titled "Web System for the management and control of computer media in the Water and Sewerage Company of Matanzas (UEB) Calimete" aims to develop an application that allows for the efficient administration and control of computer media in the UEB Calimete. The research presents the results, which include an exhaustive study of the fundamental concepts related to information management. An analysis of various tools, technologies, and languages necessary for system development is conducted, such as HTML, JavaScript, CSS, and tools for modeling and design such as Visual Studio Code, XAMPP, MySQL, among others. Additionally, a database and a web platform in line with the requirements of the Web System for the administration and control of computer media in the UEB Calimete are implemented.

Comprehensive tests are carried out to verify compliance with the established requirements, and the results documenting the proper functioning of the system are documented. Finally, recommendations are provided to enhance the system and make optimal use of the work performed.

The proposed solution provides the UEB Calimete with a management system that will allow for effective control of information related to computer media, thus improving media management in the institution.

Keywords: web system, administration, control, computer media, UEB Calimete.

Índice

Introducción.....	9
Capítulo 1: Fundamentación Teórica.....	15

1.1	Introducción al capítulo 1.....	15
1.2	Conceptos asociados a la gestión de la información.....	15
1.3	Tendencias actuales de los sistemas de gestión de la información.....	16
1.4	Aplicaciones Web.....	18
1.5	Base de datos.....	19
1.5.1	Base de datos relacional.....	20
1.5.2	Sistemas Gestores de Base de Datos.....	21
1.6	Herramientas y lenguajes de programación utilizados.....	23
1.6.1	Herramientas seleccionadas.....	23
1.6.2	Lenguajes de programación utilizados.....	28
1.7.1	Metodologías tradicionales y metodologías ágiles.....	32
1.7.2	Diferencias entre modelos de proceso convencional y ágiles.....	33
1.7.3	Metodología seleccionada.....	35
	Conclusiones del capítulo.....	37
	Capítulo 2: Diseño e implementación del sistema.....	38
2.1	Introducción del capítulo.....	38
2.2	Proceso de gestión de la información de los medios informáticos en la entidad.....	38
2.3	Requisitos Funcionales.....	39
2.3.1	Requisitos no funcionales.....	43
2.3.2	Validación de Requisitos de Software.....	45
2.4	Fase de Planeación.....	45
2.4.1	Historias de usuarios, tareas de ingeniería y tarjetas CRC.....	45
2.4.2	Historias de usuarios.....	46
2.4.3	Plan de Iteraciones (PI).....	49
2.4.4	Tarjetas CRC.....	49
2.4.5	Tareas de ingeniería.....	51
2.5	Arquitectura del software y patrones de diseño.....	53
2.5.1	Arquitectura de software.....	53
2.5.2	Patrones de diseño.....	54
	Conclusiones del capítulo.....	57
	Capítulo 3: Implementación y validación del Sistema Web para la gestión y control de los medios informáticos en la Empresa de Acueducto y Alcantarillado de Matanzas (UEB) Calimete.....	58
3.1	Tareas de Ingeniería.....	58
3.2	Estilos de programación.....	61

3.3 Validación del diseño.....	62
3.3.1 Métrica relaciones entre clases.....	62
3.3.2. Métrica tamaño operacional de clases.....	65
3.4 Pruebas de software.....	67
3.4.1 Pruebas unitarias.....	67
3.4.2 Pruebas de aceptación.....	71
3.4.3 Prueba de usabilidad.....	72
3.4.4 Pruebas carga y estrés.....	73
Conclusiones parciales.....	74
Conclusiones Generales.....	75
Bibliografía.....	77
Anexos.....	80
Anexo 1.....	80
Anexo 2.....	81

Introducción

A lo largo de los siglos, la humanidad ha enfrentado el desafío de gestionar la información, una tarea complicada en épocas pasadas debido a la falta de herramientas adecuadas. Con el objetivo de preservar el conocimiento acumulado por generaciones, se buscó controlar y transmitir la información construida a través de la experiencia. A medida que se adquirían nuevos conocimientos, surgió la necesidad de organizar y almacenar esta creciente cantidad de información.

La política de informatización de la sociedad cubana constituye un proceso de utilización ordenada y masiva de las TIC en la vida cotidiana. Tiene como objetivo lograr cada vez más eficacia y eficiencia en todos los procesos llevados a cabo en el país y por consiguiente mayor generación de riqueza y aumento en la calidad de vida de los ciudadanos. Su importancia radica en que posibilita el desarrollo coherentemente de todos los campos de la sociedad en apoyo a las prioridades del país y en correspondencia con el ritmo de crecimiento de nuestra economía. Además, permite la integración de la ciencia, la tecnología y la innovación con la producción y comercialización de productos y servicios [CITATION Com17 \l 23562].

En la actualidad, la necesidad de analizar y procesar información ha llevado al desarrollo de líneas de investigación y métodos de análisis que impulsan constantemente el campo de las tecnologías de la información. La interpretación de los datos almacenados requiere técnicas avanzadas, ya que el almacenamiento tradicional se ha vuelto cada vez más insuficiente. La gestión de la información ha adquirido una importancia significativa a nivel mundial, impactando la economía de los países. Existe una estrecha relación entre la gestión de la información y la calidad de las operaciones en una organización, reconociendo las tecnologías como medios esenciales para transmitir y gestionar el conocimiento y la información, fundamentales para el desarrollo empresarial.

En Cuba, el uso de tecnologías para la gestión de procesos, resolución de problemas y manejo de datos ha ido en aumento. La Empresa de Acueducto y

Alcantarillado de Matanzas (UEB) Calimete reconoce la importancia de emplear estas tecnologías para garantizar la rapidez, flujo y procesamiento constante de la información. La situación actual impone la necesidad urgente de transformar y mejorar la organización de manera eficiente y efectiva, adaptándose a los cambios del entorno. Para lograr este propósito, el gobierno impulsa y desarrolla proyectos de informatización en diversas áreas de la sociedad mediante la implementación de sistemas informáticos, redes y equipos de cómputo.

Uno de estos proyectos se dirige a la Empresa de Acueducto y Alcantarillado de Matanzas UEB Calimete, la cual maneja una gran cantidad de información que necesita ser informatizada para gestionar y controlar los activos informáticos. La entidad está compuesta por varios departamentos, cada uno con su propio conjunto de computadoras y dispositivos, lo que permite la formación de una red para el control de los servicios informáticos, como servicios de directorio de usuarios, chat, correo e Internet. Sin embargo, existen dificultades en el manejo de la información de los medios básicos informáticos, ya que los datos se encuentran en fuentes no seguras, como hojas de papel o en libros de cálculo digitales en Microsoft Excel.

El incremento en el número de equipos de cómputo ha llevado a un mayor nivel de responsabilidad para el departamento de Informática, surgiendo nuevas necesidades, como la captura de información relevante para el departamento de Informática, pero no para el departamento de Economía. Las operaciones llevadas a cabo por Copextel, así como las auditorías realizadas por los departamentos de Economía e Informática en la Empresa de Acueducto y Alcantarillado de Matanzas (UEB) Calimete, se enfrentan a inconvenientes debido a que la información de los medios informáticos no está centralizada. Cada departamento tiene los expedientes de sus propios equipos, y solo los departamentos de Economía e Informática cuentan con una copia de todos los documentos. Esto implica que, si se desea revisar la información de un equipo en particular, es necesario llevar consigo todos los expedientes. Además, existen casos de duplicación de información y falta de actualización, lo que no refleja correctamente la existencia y características reales de los medios.

La falta de centralización de la información tiene un impacto negativo, ralentizando y complicando el proceso de actualización. Se requieren largas jornadas de trabajo para tareas como generar y actualizar informes o reportes de los medios básicos informáticos, así como para realizar consultas o verificaciones. Cualquier actualización implica la creación de un nuevo expediente por equipo para el departamento de Economía, el departamento de Informática y el departamento al que pertenece el equipo. El desorden existente propicia errores humanos, demoras en la gestión de la información, robos no autorizados de componentes informáticos y modificaciones indebidas por parte de personal no autorizado. Esta desorganización y falta de control compromete la integridad y confiabilidad de los datos, facilita la eliminación accidental de información, errores humanos o acciones malintencionadas, y descentraliza aún más la información.

Dificultades en la gestión de los medios informáticos:

1. La información se encuentra almacenada en fuentes no seguras, ya sea en formato físico como hojas de papel o en formatos digitales como libros de Microsoft Excel.
2. La información está descentralizada, dificultando su acceso y actualización.
3. Existe un alto grado de desorden en la manipulación de la información, lo que resulta en duplicación de datos y falta de actualización, lo que no refleja correctamente la existencia y características de los activos.
4. Cada consulta o verificación requiere la generación de nuevos expedientes en el departamento de Economía, Informática y en el departamento correspondiente al equipo en cuestión.
5. Se experimentan demoras en la gestión de los medios informáticos.
6. Se han registrado modificaciones indebidas y delitos como el robo o cambios no autorizados en los componentes informáticos.

7. La eliminación de datos se facilita debido a descuidos, errores humanos o acciones poco éticas por parte de algunas personas.
8. Falta de documentación y registros actualizados: La falta de documentación detallada y registros actualizados dificulta la comprensión completa de los medios informáticos. Esto puede generar confusiones y dificultades al intentar realizar cambios, actualizaciones o mantenimiento de los sistemas y aplicaciones heredados.
9. Dependencia de personal especializado: El manejo y control efectivo de los medios informáticos requiere de personal capacitado y especializado en las tecnologías y sistemas utilizados. La falta de personal con conocimientos adecuados puede generar problemas en la administración y mantenimiento de los medios, así como en la toma de decisiones estratégicas relacionadas con su gestión.

Por todo lo anteriormente expresado, se plantea el siguiente **problema a resolver**: ¿Cómo mejorar la gestión de la información relacionada con los medios informáticos en la Empresa de Acueducto y Alcantarillado de Matanzas UEB Calimete?

En relación a lo anterior, se define el **objeto de estudio** como el proceso de gestión de la información de los medios informáticos.

El **campo de acción** se establece como el proceso de gestión de los medios informáticos en la Empresa de Acueducto y Alcantarillado de Matanzas UEB Calimete.

Con el fin de solucionar el problema planteado, se establece como **objetivo general**: Desarrollar un sistema web que permita gestionar y controlar de manera eficiente la información relacionada con los medios informáticos en la Empresa de Acueducto y Alcantarillado de Matanzas UEB Calimete.

Para darle cumplimiento al objetivo general de la investigación se trazan los siguientes **objetivos específicos**:

1. Analizar los referentes teóricos metodológicos que sustentan el

desarrollo de sitios web enfocados a la gestión de la información de los medios informáticos.

2. Analizar las tendencias actuales de los sitios web enfocados a la gestión de la información en la empresa UEB Calimete.
3. Seleccionar las herramientas, tecnologías necesarias, así como la metodología para la implementación del sitio web.
4. Implementar las funcionalidades del sitio web, a partir del levantamiento de requisitos, análisis y diseño del mismo.
5. Validar el funcionamiento de la solución propuesta y rendimiento del sitio web a través de las pruebas de *software*.

Con el fin de resolver el problema de investigación y lograr el objetivo planteado de manera sistemática y progresiva, se proponen las siguientes **tareas de investigación**:

1. Elaboración del marco teórico de la investigación, identificando las principales tendencias y soluciones de software existentes para la gestión de la información relacionada con los medios informáticos.
2. Definición de los procesos fundamentales relacionados con la gestión de la información de los medios informáticos en la Empresa de Acueducto y Alcantarillado de Matanzas UEB Calimete.
3. Aplicación de instrumentos de investigación para recopilar información acerca del sistema de gestión actual de la empresa.
4. Obtención de la lista de los activos de los medios informáticos presentes en la Empresa.
5. Selección de las herramientas, tecnologías, lenguajes de programación y metodologías más adecuadas para el desarrollo del sistema web de gestión y control de los medios informáticos en la Empresa de Acueducto y Alcantarillado de Matanzas UEB Calimete.
6. Implementación de las funcionalidades del sistema web, adaptándolas a las necesidades específicas de la empresa.

7. Validación de los resultados obtenidos mediante pruebas exhaustivas del sistema web en un entorno controlado, evaluando su eficiencia y efectividad en la gestión y control de los medios informáticos en la Empresa de Acueducto y Alcantarillado de Matanzas UEB Calimete.

En la presente investigación se destaca la utilización de **métodos científicos**, que se clasifican en métodos teóricos y métodos empíricos.

Los métodos teóricos permiten estudiar las características del problema que no son observables directamente, o sea involucran la recolección de datos utilizando técnicas que no pretenden medir, ni asociar las mediciones con números. [CITATION MarcadorDePosición5 \l 3082]. De estos métodos se utilizaron los siguientes:

Métodos Teóricos:

1. **Método Histórico-Lógico:** Este enfoque permitió examinar la evolución del desarrollo de los sistemas de gestión de la información, lo que facilitó el análisis de su trayectoria y comprensión lógica de las tendencias actuales. Se llevó a cabo una comparación entre las diversas plataformas de aplicaciones disponibles, tanto en línea como de escritorio, con el propósito de seleccionar la más adecuada para resolver el problema de investigación planteado[CITATION Dan19 \l 3082].

2. **Método Analítico-Sintético:** Este enfoque permitió comprender los conceptos fundamentales relacionados con la gestión de los medios informáticos. Además, se analizó la documentación disponible para identificar los elementos esenciales del proceso de gestión de los medios informáticos. Esto posibilitó realizar la búsqueda de herramientas y metodologías adecuadas para el desarrollo de la aplicación web propuesta en la investigación[CITATION Dan19 \l 3082].

Métodos Empíricos:

1. **Entrevistas:** Se llevaron a cabo entrevistas con el objetivo de recopilar información relevante para el desarrollo del sistema web y el control de los medios informáticos en la Empresa de Acueducto y Alcantarillado de

Matanzas(UEB)Calimete.

1.1 Entrevista estructurada: Se utilizó este enfoque de entrevista, en el cual se diseñó un cuestionario específico para obtener los datos necesarios sobre las operaciones y los procesos relacionados con la gestión y control de los medios informáticos en la empresa mencionada. Mediante esta entrevista, se buscó identificar las prácticas actuales y los problemas existentes en el manejo de los medios informáticos [CITATION Dar18 \l 3082].

Capítulo 1: Fundamentación Teórica

1.1 Introducción al capítulo 1

En el mundo de la informática, específicamente en la producción de software, es común observar como lo que hasta ayer parecía la mejor solución hoy resulta obsoleto o atrasado, y mañana surge otra mejor. Esta idea lejos de agobiar a los desarrolladores les incita a adentrarse, cada vez más, dentro del excitante entorno que rodea a la mundialmente denominada “Industria del Software”. En el presente capítulo, se exponen conceptos esenciales asociados a las diversas y novedosas tecnologías de desarrollo existentes para dar solución al problema planteado.

1.2 Conceptos asociados a la gestión de la información

La gestión de la información es un campo multidisciplinario que involucra la adquisición, organización, almacenamiento, recuperación y uso efectivo de la información dentro de una organización o contexto específico. Para comprender mejor este concepto, es importante analizar y estudiar otros términos relacionados que son fundamentales en el ámbito de la gestión de la información, como proceso, sistema, gestión e información[CITATION Leo19 \l 3082].

- **Proceso**

Un proceso se refiere a una serie de actividades o pasos interrelacionados que se realizan para lograr un objetivo específico. En el contexto de la gestión de la información, los procesos están relacionados con las actividades involucradas en la recopilación, procesamiento, almacenamiento y distribución de la información. Estos procesos pueden ser manuales o automatizados, y su eficiencia y efectividad son fundamentales para una gestión de la información exitosa[CITATION Gut02 \l 3082].

- **Sistema**

Un sistema se define como un conjunto de componentes interrelacionados que trabajan juntos para lograr un objetivo común. En el contexto de la gestión de la información, un sistema de gestión de la información se refiere a la infraestructura y las herramientas utilizadas para recopilar, almacenar, procesar y distribuir la información de manera eficiente. Esto puede incluir bases de datos, software, hardware y redes de comunicación que facilitan la gestión y el flujo de información dentro de una organización[CITATION Mon19 \l 3082].

- **Gestión**

La gestión implica la coordinación y supervisión de actividades para lograr los objetivos establecidos. En el contexto de la gestión de la información, la gestión se relaciona con la planificación, organización, dirección y control de los procesos y sistemas utilizados para administrar la información. Esto implica tomar decisiones estratégicas, asignar recursos adecuados y establecer políticas y procedimientos para garantizar una gestión efectiva de la información en una organización.

- **Información**

La información se refiere a los datos procesados y organizados que tienen valor y significado para un receptor o usuario específico. La información es el resultado de la transformación de datos en conocimiento útil para la toma de decisiones. En el contexto de la gestión de la información, se busca garantizar la disponibilidad, integridad, confidencialidad y calidad de la información, así como su relevancia y pertinencia para los usuarios finales.

1.3 Tendencias actuales de los sistemas de gestión de la información

En la actualidad, los sistemas de gestión de la información están evolucionando para adaptarse a las necesidades cambiantes de las organizaciones y aprovechar el avance tecnológico. Aquí se presentan algunas tendencias y

características esenciales que se observan en los sistemas de gestión de la información:

- **Base de Datos:** Las bases de datos siguen siendo una parte fundamental de los sistemas de gestión de la información. Sin embargo, las tendencias actuales se centran en la utilización de bases de datos más eficientes y escalables, como las bases de datos en la nube y las bases de datos NoSQL. Estas tecnologías permiten el almacenamiento y procesamiento de grandes volúmenes de datos de manera más rápida y flexible[CITATION 23Mo \l 3082].
- **Informes:** Los sistemas de gestión de la información actuales buscan proporcionar informes y análisis más avanzados y personalizados. Las herramientas de generación de informes se están mejorando para permitir la visualización de datos en tiempo real, la generación automática de informes interactivos y la capacidad de personalizar los informes según las necesidades de los usuarios. Además, se están utilizando técnicas de inteligencia artificial y aprendizaje automático para analizar y extraer información útil de los datos.
- **Integración y acceso abierto:** Los sistemas de gestión de la información están adoptando enfoques de integración más sólidos para permitir la interoperabilidad y el intercambio de datos entre diferentes sistemas y aplicaciones. La integración con otras plataformas y sistemas externos se ha vuelto cada vez más importante para facilitar la recopilación y el análisis de datos de diversas fuentes. Además, se fomenta el acceso abierto a la información, permitiendo que los datos sean accesibles y compartidos de forma segura entre diferentes usuarios y organizaciones[CITATION Int22 \t \l 3082].

Ejemplo de sistema de gestión de la información nacional en Cuba:

1. Sistema de Gestión de la Información de Salud (SGIS): El **SGIS** en Cuba es un sistema integral que recopila, almacena y gestiona datos relacionados con la salud de la población cubana. Permite el

seguimiento de casos médicos, control de epidemias, estadísticas de salud y facilita el acceso a la información médica para profesionales de la salud en todo el país[CITATION Fre16 \l 3082].

2. Sistema de Gestión de la Información Turística (SGIT): El **SGIT** en Cuba se enfoca en la recopilación y gestión de datos relacionados con el turismo en la isla. Proporciona información actualizada sobre destinos turísticos, alojamientos, actividades y eventos para ayudar a los turistas y a la industria turística a tomar decisiones informadas y promover el desarrollo turístico sostenible[CITATION And18 \l 3082].

Ejemplos de sistemas de gestión de la información internacionales:

1. Sistema de Gestión de la Información Geoespacial (SIG): Los **SIG** internacionales se utilizan para recopilar, analizar y visualizar datos geoespaciales a nivel global. Estos sistemas permiten la integración de datos de mapas, imágenes satelitales, datos demográficos y otras fuentes para brindar información valiosa en campos como la planificación urbana, gestión de desastres y toma de decisiones en política ambiental[CITATION Pac18 \l 3082].
2. Sistema de Gestión de la Información Financiera (SGIF): Los **SGIF** internacionales son utilizados por organizaciones y gobiernos a nivel mundial para gestionar y controlar la información financiera. Estos sistemas integran datos de contabilidad, presupuesto, auditoría y reportes financieros para garantizar la transparencia, la eficiencia y el cumplimiento de las regulaciones financieras en diferentes sectores y países[CITATION Yes22 \l 3082].

1.4 Aplicaciones Web

Una aplicación web o llamadas online que se conectan a un servidor ofrece ventajas significativas. En primer lugar, brindan accesibilidad desde cualquier dispositivo con conexión a internet, lo que permite a los usuarios utilizar la aplicación o realizar llamadas desde cualquier ubicación. Además, las actualizaciones del software se realizan de

manera centralizada en el servidor, garantizando que todos los usuarios tengan acceso a las últimas mejoras y funcionalidades[CITATION Luj02 \l 3082].

Una aplicación web o llamadas online que se conectan a un servidor presenta las siguientes **ventajas**:

1. Accesibilidad: Pueden ser utilizadas desde cualquier dispositivo con acceso a internet.
2. Actualizaciones centralizadas: Las actualizaciones del software se realizan en el servidor, lo que garantiza que todos los usuarios tengan acceso a la última versión.
3. Compatibilidad multiplataforma: Funcionan en diferentes sistemas operativos y dispositivos, como computadoras, tablets y smartphones.
4. Colaboración en tiempo real: Permiten la colaboración simultánea entre usuarios, facilitando la comunicación y el trabajo en equipo.
5. Almacenamiento centralizado: Los datos se almacenan en el servidor, lo que proporciona una mayor seguridad y respaldo de la información.

Sin embargo, también presentan algunas **desventajas**:

1. Dependencia de la conexión a internet: La calidad y disponibilidad de la conexión a internet afecta la experiencia de uso.
2. Posible vulnerabilidad a ataques: Al estar conectadas a internet, pueden estar expuestas a riesgos de seguridad y ataques cibernéticos si no se implementan medidas adecuadas.
3. Limitaciones de rendimiento: El rendimiento de la aplicación puede verse afectado por la capacidad del servidor y la carga de usuarios simultáneos.

4. Privacidad de los datos: Existe la preocupación de la privacidad y el manejo de los datos personales almacenados en el servidor.
5. Dependencia del proveedor del servidor: La disponibilidad y el rendimiento de la aplicación están sujetos al proveedor del servidor, lo que puede generar limitaciones y restricciones.

1.5 Base de datos

Una base de datos es un sistema de almacenamiento y organización de información estructurada de manera que se pueda acceder, gestionar y actualizar de manera eficiente. Las bases de datos proporcionan una estructura y un conjunto de reglas para almacenar datos de forma coherente, lo que permite la gestión y manipulación de grandes volúmenes de información de manera eficaz que se almacenan en un equipo o dispositivo de cómputo [CITATION Leo19 \l 3082].

1.5.1 Base de datos relacional

Las bases de datos relacionales utilizan tablas con filas y columnas para almacenar y organizar datos de manera estructurada. Por esto fue seleccionada para la realización de este proyecto.

Ventajas

1. Estructura organizada: Las bases de datos relacionales ofrecen una estructura clara y organizada mediante tablas y relaciones, lo que facilita la gestión y el análisis de los datos.
2. Integridad de los datos: Las bases de datos relacionales permiten establecer restricciones de integridad, como claves primarias y claves externas, para garantizar la consistencia y la precisión de los datos almacenados.
3. Consultas flexibles: Utilizando el lenguaje estructurado de consultas (SQL), es posible realizar consultas complejas y extraer información específica de manera eficiente y precisa.

4. Escalabilidad: Las bases de datos relacionales son escalables, lo que significa que se pueden adaptar y crecer a medida que aumentan los datos y las necesidades del proyecto.
5. Amplio soporte y herramientas: Existen múltiples sistemas de gestión de bases de datos relacionales (por ejemplo, MySQL, PostgreSQL, Oracle) con una amplia comunidad de usuarios y una gran cantidad de herramientas disponibles, lo que facilita el desarrollo y la administración de la base de datos.

Desventajas

1. Rigidez en la estructura: Las bases de datos relacionales tienen una estructura fija y rígida. Si los requisitos del proyecto cambian y es necesario modificar la estructura de la base de datos, puede requerir un proceso complejo y costoso.
2. Rendimiento en escenarios complejos: En situaciones con grandes volúmenes de datos y consultas complejas, las bases de datos relacionales pueden experimentar un rendimiento más lento en comparación con otros tipos de bases de datos no relacionales, como bases de datos basadas en documentos o gráficas.
3. Dificultad en la escalabilidad horizontal: A medida que crece la base de datos y se requiere una escalabilidad horizontal (distribución en varios servidores), las bases de datos relacionales pueden enfrentar desafíos adicionales en términos de rendimiento y mantenimiento.

1.5.2 Sistemas Gestores de Base de Datos

Los Sistemas Gestores de Bases de Datos (SGBD) son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. Aun cuando en la actualidad existe una gran variedad de SGBD, las principales funciones que deben cumplir se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, así como evitar

redundancias e inconsistencias y mantener la integridad. A través de él se maneja todo acceso a la base de datos con el objetivo de servir de interfaz entre ésta, el usuario y las aplicaciones [CITATION Rou191 \l 3082]. Para el desarrollo de la solución propuesta se hace necesario utilizar el MySQL como sistema gestor de bases de datos. A continuación, se realiza un análisis, desde la perspectiva del software libre, de los SGBD MySQL y PostgreSQL como las herramientas para administrar y desarrollar sistemas de información más comunes dentro del desarrollo Web[CITATION Leo19 \l 3082].

MySQL13.3

Se trata de un sistema de gestión de base de datos relacional, multihilo y multiusuario. Constituye el servidor de bases de datos relacionales más popular y es desarrollado por la empresa MySQL AB, la cual desde enero del 2008 pertenece a Sun Microsystems. MySQL es desarrollado como software libre siguiendo un esquema de licenciamiento dual, o sea, se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia. Sin embargo, si no se está de acuerdo con la misma o se necesita incorporar MySQL en una aplicación comercial es posible comprar una licencia específica que permita este uso. Contrario a otros proyectos de software libre, como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código[CITATION Leo19 \l 3082].

PostgreSQL

Es un sistema de gestión de bases de datos (SGBD) de software libre y avanzado, publicado bajo la licencia BSD. Se destaca por proporcionar numerosas características que antes solo estaban disponibles en SGBD comerciales de alto nivel, como Oracle. Es un SGBD objeto-relacional que maneja rutinas y reglas complejas.

Algunas de sus características fundamentales son:

1. Gran escalabilidad: PostgreSQL es altamente ajustable al número de procesadores y cantidad de memoria del sistema, lo que le permite soportar eficientemente una mayor cantidad de peticiones simultáneas. Esto lo hace adecuado para entornos universitarios sin requerir un sistema de cómputo avanzado.
2. Almacenamiento de procedimientos almacenados: Tiene la capacidad de almacenar procedimientos almacenados directamente en la base de datos, lo que mejora el rendimiento y la eficiencia de las operaciones.
3. Multiusuario, cliente-servidor y control de privilegios de acceso: PostgreSQL es compatible con múltiples usuarios y sigue una arquitectura cliente-servidor. Además, ofrece un control de privilegios de acceso que permite gestionar y restringir el acceso a los datos según los roles y permisos asignados.
4. Mejoras en los tipos de datos internos: PostgreSQL ha mejorado los tipos de datos internos, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales. Esto proporciona una mayor flexibilidad y precisión en el manejo de diferentes tipos de datos.
5. PostgreSQL también es compatible con una amplia gama de lenguajes de programación populares y es capaz de soportar un número ilimitado de bases de datos.

En resumen, PostgreSQL es un SGBD de software libre que ofrece gran escalabilidad, capacidad de almacenar procedimientos almacenados, soporte multiusuario con arquitectura cliente-servidor y control de privilegios de acceso, así como mejoras en los tipos de datos internos.

1.6 Herramientas y lenguajes de programación utilizados

En el proceso de desarrollo de software, se emplean diversas herramientas y lenguajes de programación para asegurar un manejo eficiente y adecuado de la información.

1.6.1 Herramientas seleccionadas

MySQL 13.3

MySQL es un sistema de gestión de bases de datos relacionales de código abierto con un modelo cliente-servidor. Los procesos principales que tienen lugar en un entorno MySQL son (Gustavo B,2019):

- Crea una base de datos para almacenar y manipular datos, definiendo la relación de cada tabla.
- Los clientes pueden realizar solicitudes escribiendo instrucciones SQL. MySQL tiene ventajas al ser un software con Licencia GPL de bajo costo en requerimientos para la elaboración y ejecución del programa. No se necesita disponer de hardware o software de alto rendimiento para la ejecución del programa, logra velocidad al realizar las operaciones y buen rendimiento. Es fácil su instalación y configuración. Soporte en casi el 100% de los sistemas operativos actuales y baja probabilidad de corrupción de datos en entorno con seguridad y encriptación. Aunque presenta una desventaja que afecta su utilización que es que, a pesar de ser software libre, no presenta documentación oficial y muchas de las soluciones para las deficiencias no están documentados, ni muchas de sus utilidades.

Framework para el desarrollo.

Un framework simplifica el desarrollo de las aplicaciones, ya que automatiza muchos de los patrones utilizados para resolver las tareas comunes. Además, proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

Fundamentación del IDE de desarrollo seleccionado

Teniendo en cuenta los elementos antes citados se propone la utilización de MySQL 13.3

Es un entorno de desarrollo gratuito y de código abierto. Permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como para aplicaciones web, o para dispositivos móviles. Da soporte a las siguientes tecnologías, entre otras: Java, PHP, Groovy, C/C++ y HTML5. Además, posee una muy importante característica, es multiplataforma. (Goovy, 2014)

Características principales

- Suele dar soporte a casi todas las novedades en el lenguaje Java.
- Asistentes para la creación y configuración de distintos proyectos, incluida la elección de algunos

marcos de trabajo.

- Buen editor de código multilenguaje, con habitual coloreado y sugerencias de código, acceso a

clases pinchando en el código, control de versiones, localización de la ubicación de la clase actual.

- Simplifica la gestión de grandes proyectos con el uso de diferentes vistas, asistente de ayuda, y

estructurando la visualización de manera ordenada, lo que ayuda en el trabajo diario.

- Herramienta para el depurado de errores.
- Optimización de código.
- Acceso a bases de datos.

Geany

Es un editor de texto usando el kit de herramientas GTK2 con las características básicas de un entorno de desarrollo integrado. Fue desarrollado para proporcionar un pequeño y rápido IDE, que tiene sólo unas pocas dependencias de otros paquetes. Es compatible con muchos tipos de archivos. Está disponible para distintos sistemas operativos, como GNU/Linux, Mac OS X, BSD, Solaris y Microsoft Windows. Es distribuido como software libre bajo la Licencia Pública General de GNU. Algunas de sus principales funcionalidades son:

- Autocompletado.
- Código de navegación.
- Cierre automático de etiquetas XML y HTML.
- Muchos tipos de archivos soportados tales como C, Java, PHP, Python, Perl y Pascal.
- Fácil gestión de proyectos.
- Soporte para plugins.

Notepad ++

Notepad++ es un editor de texto de código fuente libre con soporte para varios lenguajes de programación. Se distribuye bajo los términos de la Licencia Pública General de GNU. Aunque su soporte nativo es Microsoft Windows puede utilizarse en GNU/Linux mediante Wine. Basado en un potente componente de edición llamado Scintilla, Notepad++ está escrito en C++ y usa directamente la API de Win32 y STL, lo que asegura una gran velocidad de ejecución y un menor tamaño del programa. Mediante la optimización de tantas rutinas como sea posible sin perder la interfaz amigable con el usuario. Algunas de sus principales características son:

- Resaltado de sintaxis y envoltura de sintaxis
- WYSIWYG
- Resaltado de sintaxis definido por el usuario
- Auto-completado
- Multi-documento
- Multi-vista
- Soporte para la búsqueda y substitución de expresiones regulares
- Soporte completo para arrastrar y soltar
- Posición dinámica de vistas
- Auto-detección de estado del fichero
- Aumentar y reducir el zoom
- Entorno multi-lenguaje soportado
- Favoritos

Visual Studio Code 1.71.0

Visual Studio Code es un editor de código en términos sencillos. Visual Studio Code es “un editor libre que ayuda al programador a escribir código, ayuda a depurar y corrige el código utilizando el método intelli-sense”. En términos normales, facilita a los usuarios a escribir el código de una manera fácil. Mucha gente dice que es la mitad de un IDE y un editor, pero la decisión depende de los codificadores. Cualquier programa/software que veamos o utilicemos, trabaja sobre el código que se ejecuta en segundo plano. Tradicionalmente la codificación solía hacerse en los editores tradicionales o incluso en los editores básicos como el bloc de notas. Estos editores solían proporcionar un apoyo

básico a los codificadores. Entre las principales características de Visual Studio Code podemos mencionar:

- Tiene una amplia variedad de themes para configurar el entorno de trabajo[CITATION htt \l 3082].
- Soporte para múltiples lenguajes de programación: Visual Studio Code cuenta con una gran compatibilidad con la mayoría de Lenguajes de Programación que se usan a diario. Así que antes, los programadores necesitaban un editor diferente para diferentes lenguajes, pero este tiene soporte multilingüe incorporado. Esto también significa que detecta fácilmente, si hay algún fallo o referencia entre lenguajes, será capaz de detectarlo fácilmente. Ver en[CITATION vis \l 3082]
- Soporte multiplataforma: Tradicionalmente, los editores solían ser compatibles con los sistemas Windows, Linux o Mac. Pero Visual Studio Code es multiplataforma. Así que puede funcionar en las tres plataformas. Además, el código funciona en las tres plataformas, de lo contrario, los códigos de software propietario y de código abierto solían ser diferentes.
- Cuenta con extensiones que te ayudaran a trabajar aún más rápido en el lenguaje de programación de tu preferencia[CITATION vis \l 3082].
- Multiproyectos: Se pueden abrir simultáneamente varios proyectos que contengan múltiples archivos/carpetas.

Estos proyectos/carpetas pueden o no estar relacionados entre sí.

Fundamentación del IDE de desarrollo seleccionado

Teniendo en cuenta los elementos antes citados se propone la utilización de la herramienta Visual Studio Code.

Apache

Se optó por utilizar el servidor web Apache debido a su capacidad para funcionar en una amplia gama de sistemas operativos. Además, es gratuito, de código abierto y cuenta con una gran facilidad para expandir sus capacidades. Una ventaja adicional es que permite la creación de archivos de registro, lo que brinda un mayor control sobre las actividades que ocurren en el servidor [CITATION Fri20 \l 3082].

XAMPP V7.2.10

Se hará uso del servidor XAMPP, puesto que es una plataforma de código libre. Permitirá instalar de forma sencilla Apache en tu propio ordenador, sin importar tu sistema operativo y su uso es gratuito. Es una herramienta de desarrollo que permite probar el sistema en el propio ordenador sin necesidad de tener que acceder a internet. También se utiliza actualmente como servidor de sitios web, ya que, con algunas modificaciones, es generalmente lo suficientemente seguro [CITATION Placeholder26 \l 23562].

1.6.2 Lenguajes de programación utilizados

Lenguajes del lado del cliente

HTML5

HTML (Hyper Text Markup Lenguaje) es el lenguaje que describe la estructura y el contenido de un sitio web en forma de texto, permitiendo que los navegadores lean e interpreten el código y presenten al usuario un sitio web de la forma en que comúnmente es visualizado [CITATION W3C15 \l 3082].

HTML5 introdujo varias características nuevas y mejoradas en comparación con las versiones anteriores de HTML. Algunas de estas características incluyen:

1. Semántica mejorada: HTML5 ofrece una amplia gama de elementos semánticos que permiten una estructura más clara y significativa del contenido, como <header>, <nav>, <article>, <section>, <footer>, entre otros.

2. Etiquetas de audio y video: HTML5 proporciona las etiquetas <audio> y <video> que permiten la reproducción de contenido multimedia sin necesidad de plugins o complementos externos.
3. Canvas: El elemento <canvas> permite la creación de gráficos y animaciones interactivas utilizando JavaScript. Es útil para aplicaciones y juegos web.

CSS 3

Hoja de estilo en cascada o CSS (Cascading Style Sheets) es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. El código de estilo puede ser definido en un documento separado o en el mismo documento HTML. CSS permite el control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo. Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario[CITATION W3C152 \l 3082]

JavaScript 1.9

Es un lenguaje de programación interpretado y orientado a objetos que se utiliza principalmente en el desarrollo web. Es ampliamente conocido por su capacidad de interactuar con elementos HTML y CSS, lo que permite crear sitios web dinámicos e interactivos.

Una de las principales ventajas de JavaScript es su ejecución en el navegador del cliente, lo que significa que se puede utilizar directamente en el navegador sin necesidad de plugins o programas externos. Esto lo convierte en una opción popular para agregar interactividad y funcionalidad a las páginas web [CITATION Placeholder21 \l 23562].

SQL Lenguaje de Consultas Estructurado (SQL por sus siglas en inglés). Es el lenguaje por el cual se puede acceder a las bases de datos relacionales. Con este lenguaje se construirán estructuras y objetos de datos, se insertarán: datos, se modificarán y se realizarán consultas básicas. Proporciona la manera de acceder directamente a los datos no sólo a través de los lenguajes de programación como PHP, sino desde el mismo Sistema Gestor de Bases de Datos (SGBD). El SQL se comporta como un lenguaje de “alto nivel”, con una estructura declarativa de sentencias que posee una sintaxis particular y estándar.

Lenguaje del lado del servidor

PHP v8.1

PHP es un lenguaje de 'scripting' de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser embebido en páginas HTML.[CITATION The19 \l 3082]

Algunas ventajas que presenta:

- Muy sencillo de aprender.
- Permite las técnicas de Programación Orientada a Objeto (POO).
- Es *software* libre.
- Se puede obtener en la Web y su código está disponible bajo la Licencia Pública General (del inglés *General Public License*, GPL).
- Cuenta con una excelente biblioteca de funciones que permite realizar cualquier labor (acceso a base de datos, encriptación y creación de PDF).
- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad. Puede interactuar con muchos motores de base de datos tales como MySQL, Oracle y PostgreSQL.
- Permite leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.

Una de las principales ventajas de PHP v8.1 es su notable aumento en el rendimiento. Gracias a la introducción del nuevo motor de ejecución Zend Engine 3.0, PHP v8.1 es considerablemente más rápido y consume menos recursos en comparación con versiones anteriores. Esto se traduce en una mayor capacidad de respuesta y tiempos de carga más rápidos para las aplicaciones web[CITATION Ioa23 \l 3082].

Framework para el desarrollo

Un framework simplifica el desarrollo de las aplicaciones, ya que automatiza muchos de los patrones utilizados para resolver las tareas comunes. Además, proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

Drupal 9.0

Drupal es un sistema de gestión de contenidos (CMS proviene del inglés Content Management System) multipropósito, modular, libre y con una amplia capacidad de personalización. Permite publicar archivos, imágenes, artículos, al igual que crear y administrar todo tipo de contenidos como votaciones, encuestas, foros, entre otros. A pesar de no ser tan popular como WordPress, Drupal es uno de los CMS más completos para grandes portales corporativos. Además, se trata de una plataforma flexible, siendo ideal para usuarios con conocimientos avanzados y fácilmente integrable con otras soluciones de negocio. Ayuda a crear proyectos complejos, generar una cantidad de tráfico elevada, procesar datos y desarrollar funcionalidades específicas. En vista de lo efectivo que es Drupal como sistema de gestión de contenidos, cada vez hay más usuarios que lo eligen como plataforma para sus proyectos.

Sus ventajas más importantes son las siguientes:

- Muchas posibilidades de configuración y flexibilidad.
- Interfaz de usuario intuitiva y administrable.

- Permite crear contenido optimizado para el Search Engine Optimization (SEO optimización en motores de búsqueda), además de ser reutilizable.
- Está disponible en varios idiomas, lo que es primordial para dar soporte técnico.
- Múltiples opciones de diseño.
- Amplio repertorio de extensiones y módulos.
- Posibilidad de crear grandes desarrollos, como multisites.
- Soporta una gran cantidad de tráfico. Por supuesto, no es un software perfecto, y existen algunos inconvenientes:
 1. No hay muchos plugins gratuitos disponibles si lo comparamos con otros CMS, pero con un buen equipo experto podrás desarrollar un módulo que necesites para la funcionalidad que busques.
 2. No recomendable para hacer una primera página web básica, si no que Drupal es para desarrollos más amplios y exigentes tecnológicamente.
 3. La curva de aprendizaje inicial es elevada [CITATION Placeholder24 \l 23562].

1.7.1 Metodologías tradicionales y metodologías ágiles

Metodologías Ágiles

Las metodologías ágiles son enfoques de desarrollo de software que se centran en la flexibilidad, la colaboración y la entrega incremental. Estas metodologías priorizan la adaptabilidad a los cambios, la participación activa de los clientes y la entrega temprana de incrementos de software funcionales. Algunas de las metodologías ágiles más conocidas son:

1. **Scrum:** Es un marco de trabajo que se basa en iteraciones cortas llamadas "sprints". El equipo de desarrollo trabaja en colaboración con el

cliente para entregar incrementos de software en cada sprint y realizar revisiones periódicas.

2. **Kanban:** Se centra en la visualización del flujo de trabajo y la limitación de la cantidad de trabajo en progreso. Los elementos de trabajo se mueven a través de columnas de un tablero, lo que permite un control visual del progreso y una gestión eficiente de las tareas.
3. **XP (Programación Extrema):** Se enfoca en la calidad del software y la mejora continua. Promueve la comunicación constante entre los miembros del equipo, pruebas exhaustivas, integración continua y desarrollo orientado a las pruebas.
4. **Lean Software Development:** Se inspira en los principios de la filosofía Lean para reducir el desperdicio y maximizar el valor para el cliente. Se enfoca en la entrega temprana, la optimización del flujo de trabajo y la mejora continua.
5. **Crystal:** Propone una serie de metodologías ligeras y adaptables a diferentes contextos. Se centra en la comunicación efectiva, la colaboración y la reflexión constante para mejorar el proceso de desarrollo.

Metodologías Tradicionales

Las metodologías tradicionales, también conocidas como metodologías en cascada o enfoques de desarrollo secuencial, se caracterizan por seguir un proceso lineal y secuencial en el desarrollo de software. A diferencia de las metodologías ágiles, las metodologías tradicionales tienden a ser más estructuradas y predecibles. reciben el apelativo de Metodologías Tradicionales (o también denominadas Metodologías Pesadas, o Peso Pesado).

1.7.2 Diferencias entre modelos de proceso convencional y ágiles

Las diferencias entre los modelos de proceso convencional (tradicional) y los modelos ágiles son significativas y afectan la forma en que se planifica, se

desarrolla y se entrega el software. A continuación, se presentan algunas diferencias clave[CITATION Yag09 \l 3082]:

1. Enfoque secuencial vs. iterativo: Los modelos de proceso convencional, como el modelo en cascada, siguen un enfoque secuencial y lineal, donde cada etapa (requisitos, diseño, implementación, pruebas, etc.) se completa antes de pasar a la siguiente. En cambio, los modelos ágiles, como Scrum o Kanban, se basan en un enfoque iterativo e incremental, donde se realizan entregas frecuentes de incrementos funcionales del software y se permiten ajustes y mejoras continuas a medida que se avanza.
2. Documentación exhaustiva vs. documentación justa: Los modelos convencionales suelen requerir una documentación detallada y exhaustiva en cada etapa del desarrollo. Esto puede incluir documentos como especificaciones de requisitos, diseños detallados, planes de pruebas extensos, entre otros. En los modelos ágiles, la documentación se simplifica y se centra en lo esencial, fomentando una comunicación más directa y efectiva entre los miembros del equipo de desarrollo.
3. Gestión centralizada vs. auto organización: Los modelos convencionales suelen tener una estructura de gestión jerárquica y centralizada, donde las decisiones son tomadas por la alta dirección y se asignan tareas específicas a los miembros del equipo. En los modelos ágiles, se fomenta la auto organización y la toma de decisiones colaborativa. Los equipos ágiles tienen mayor autonomía y responsabilidad para planificar y ejecutar su trabajo.
4. Cambios inflexibles vs. adaptabilidad: Los modelos convencionales suelen resistir los cambios en los requisitos una vez que se establecen. Los cambios pueden requerir una reevaluación y ajuste de todo el plan de proyecto, lo que puede ser costoso y lento. Los modelos ágiles, en cambio, se adaptan fácilmente a los cambios. Se espera que los requisitos evolucionen durante el proyecto y se incorporan ajustes continuos para satisfacer las necesidades cambiantes.

5. Enfoque en el producto vs. enfoque en el cliente: Los modelos convencionales tienden a centrarse en la entrega del producto final según los requisitos iniciales establecidos. Los modelos ágiles ponen un mayor énfasis en la satisfacción del cliente y la entrega de valor continuo. Se busca una colaboración estrecha con el cliente, se recibe retroalimentación constante y se priorizan las características y funcionalidades más importantes para el cliente[CITATION Vel19 \l 3082].

1.7.3 Metodología seleccionada

Luego de analizadas las variantes de metodologías existentes, así como sus ventajas y desventajas, se seleccionó la metodología XP pues la misma es la que se adapta de mejor forma al desarrollo de la solución.

XP (eXtreme Programming) Actualmente con el creciente desarrollo tecnológico y la aparición de nuevos modelos de producción, el uso de las metodologías ágiles gana seguidores. Entre ellas destaca XP, clasificada de metodología ligera por cuanto trata de reducir la complejidad de software orientando el trabajo directamente al objetivo, centrándose tanto en potenciar las relaciones interpersonales de los desarrolladores como en mantener un ambiente de desarrollo colaborativo y agradable, donde la superación continua de los integrantes del equipo se convierta en una práctica cotidiana. Entre sus tendencias resalta la presencia casi a tiempo completo del cliente dentro del proceso de desarrollo, garantizando que sus opiniones contribuyan a una constante retroalimentación desarrolladores-clientes. Quizás dos de los más excitantes valores que posee son: la búsqueda de simplicidad en las soluciones y el coraje requerido para enfrentar los posibles cambios.

XP se basa en UserStories (historias de usuarios), estas historias las escribe el cliente o su representante dentro del equipo y describen los escenarios clave del funcionamiento del software. A partir de estas se planifican las entregas entre el equipo y el cliente. Las entregas son frecuentes, lo que permite mejorar el diseño cada vez que se le añada una nueva funcionalidad. Estas entregas a su vez permiten definir las iteraciones necesarias para cumplir con los

objetivos, de manera que cada resultado de la iteración sea un programa aprobado por el cliente de quien depende la definición de las siguientes iteraciones.

Una característica distintiva de XP es la programación en parejas, con el objetivo de que el código sea revisado y validado antes de ser escrito; la refactorización de código está presente durante todo el desarrollo, lo cual permite reescribir el código fuente buscando claridad, pero sin cambiar la funcionalidad resultante. Las parejas no serán siempre las mismas, sino que se pretende que cada desarrollador haya formado dupla al menos una vez con todos los demás, de donde se desprende que el código es de propiedad colectiva y cada uno es responsable por todo el proyecto[CITATION Gar17 \l 3082].

Entre sus múltiples características, se destacan algunas relevantes que lo hace efectivo para proyectos de corto tiempo y un reducido grupo de personas como:

- XP está concebida para ser utilizada dentro de proyectos pequeños.
- Los requisitos del sistema cambian frecuentemente: Con la aceptación de nuevos requerimientos, el sistema debe cambiar y ampliar sus funcionalidades de forma que sea capaz de adaptarse a cada nueva situación. Uno de los principios básicos de XP es que el cambio frecuente de los requerimientos es algo normal en el proceso de desarrollo. Esta metodología se adapta perfectamente a los proyectos cuyos requerimientos cambian a menudo.
- El cliente forma parte del equipo de desarrollo: Mediante la aplicación de XP se puede lograr una retroalimentación mayor y lograr un producto que satisfaga sus necesidades.
- Poca disponibilidad de personal: El sistema debe ser realizado por dos personas solamente, no siendo posible la existencia de muchos roles ni la especialización en un rol específico por parte de los miembros. Uno de los principios básicos de XP es la programación en equipos

pequeños (2 a 12 personas) con pocos roles, pudiendo los miembros del equipo intercambiar responsabilidades en un momento determinado.

- Propiedad colectiva del código: XP plantea que todos los programadores pueden realizar cambios en cualquier parte del código en cualquier momento.

XP es una metodología incluida dentro de las denominadas ágiles, se encamina más a lograr entregas desde el principio basadas en la comunicación e interacción directa que debe existir entre el equipo de desarrollo y el cliente, siendo este último parte del mismo, además, existe poca disponibilidad de personal dentro del grupo de trabajo. Esta metodología se adapta a los proyectos donde los requisitos cambian frecuentemente, característica que no es ajena a la presente investigación. Analizadas las características de las anteriores metodologías de desarrollo de software, es selecciona XP como la candidata para el trabajo de investigación, la cual se ajusta perfectamente al mismo.

Conclusiones del capítulo

Se realizó una investigación exhaustiva previa al diseño e implementación del sistema, que incluyó un estudio de los conceptos relacionados con la gestión de información y las tendencias actuales en sistemas similares al que se desea desarrollar. Tras analizar la situación problemática, se determinó la necesidad de crear un sistema compuesto por una aplicación web y una base de datos correspondiente. El objetivo era asegurar un almacenamiento adecuado de la información y permitir el acceso a la aplicación desde cualquier equipo conectado a la red de la institución para la cual se llevó a cabo la investigación.

A partir del estudio de las tecnologías para el desarrollo del sistema web y la política de desarrollo definida en la UCI se determinó el uso de XP como metodología de desarrollo, Drupal como sistema de gestión de contenidos, MySQL como sistema gestor de base de datos y phpMyAdmin como herramienta para la administración de bases de datos en MySQL. Se decidió el

uso de los lenguajes del lado del cliente: HTML, JavaScript y CSS 3, y el lenguaje del lado del servidor PHP.

Capítulo 2: Diseño e implementación del sistema

2.1 Introducción del capítulo

El presente capítulo aborda las principales características del sistema, los distintos elementos de la planificación; diseño e implementación relacionados con la metodología utilizada. Como parte del proceso ingenieril, se detallan los principales artefactos generados por la metodología seleccionada como los requisitos de software identificados, las historias de usuario, Tarjetas Clase-Responsabilidad-Colaborador. Por último, se especifica la arquitectura empleada.

2.2 Proceso de gestión de la información de los medios informáticos en la entidad

Se plantea el desarrollo de un Sistema Web para la gestión y control de los medios informáticos en la Empresa de Acueducto y Alcantarillado de Matanzas (UEB) Calimete esta se encuentra trabajando en la digitalización de la información de los medios informáticos. La entidad está compuesta por 14 departamentos, que incluyen Dirección, Puesto de Mando, Revisión, Fiscalización, Economía, Recepción, Contabilidad, Servicios, Comprobación, Subdirección, Caja, Recursos Humanos, Almacén e Informática.

Cada departamento cuenta con un conjunto de estaciones de trabajo, que incluyen computadoras y dispositivos asociados. Estos equipos están

interconectados en una red donde se brindan y controlan los servicios informáticos, como directorio de usuarios, chat, correo e Internet.

Cabe destacar que cada estación de trabajo tiene un expediente que contiene información detallada sobre las computadoras y los dispositivos asociados, como UPS, monitores, ratones, teclados, impresoras y escáneres, así como sus componentes, como memoria RAM, microprocesador, discos duros, lectores de discos y dispositivos de audio, video y red. Estos expedientes se almacenan en los respectivos departamentos donde se encuentran las estaciones de trabajo, y también se mantienen copias en los departamentos de Economía e Informática. En particular, en el departamento de Informática, las copias de los expedientes incluyen planillas con información adicional de interés exclusivamente para el personal informático de la entidad.

Cada vez que se realiza mantenimiento a un equipo, se da de baja, se cambian componentes o se realiza un movimiento entre departamentos, se actualizan los expedientes de las estaciones de trabajo, incluyendo las copias que poseen los departamentos de Informática y Economía. Estos expedientes contienen información sobre los medios básicos informáticos y se organizan en carpetas que contienen planillas en papel.

Es importante destacar que el departamento Económico y el departamento Informático poseen parte de la información en formato digital, utilizando hojas de Microsoft Word y libros de Microsoft Excel. Cualquier verificación, cambio o consulta de información se realiza directamente en el departamento al que pertenece el equipo, y se garantiza la correspondencia de los datos en los expedientes y sus dos copias.

2.3 Requisitos Funcionales

Para la creación del sistema se tomaron un conjunto de funcionalidades obtenidas como resultado del levantamiento de información. Cada funcionalidad constituye una característica que el sistema debe tener o una restricción que el sistema debe satisfacer en respuesta de las necesidades del

cliente. En la siguiente tabla se especifican los requisitos funcionales que se definieron para la realización de este proyecto.

Tabla 1. Descripción de los Requisitos Funcionales.

(Fuente: Elaboración propia)

No.	Nombre	Descripción
RF1.	Iniciar sesión	Permitirá al usuario ser autenticado en el sitio web permitiéndole o denegando su entrada
RF2.	Cerrar sesión	Permitirá al usuario salir del sitio web
RF3.	Insertar información de las computadoras	Permite crear una computadora en el sitio web
RF4.	Eliminar información de las computadoras	Permite eliminar una computadora en el sitio web
RF5.	Modificar información de las computadoras	Permite modificar una computadora en el sitio web
RF6.	Listar información de las computadoras	Permite listar todas las computadoras en el sitio web por fecha, autor y tema.
RF7.	Ver detalles de las computadoras	Permite visualizar todas las computadoras en el sitio web por fecha ,tipo de computadora y autor
RF8.	Filtrar información de las computadoras	Permite filtrar todas las computadoras en el sitio web por fecha, autor y tema.
RF9.	Generar reporte de las computadoras	Permite crear un reporte de las computadoras
RF10.	Insertar información de los responsables de las computadoras	Permite crear un responsable de computadoras en el sitio web
RF11.	Listar información de los responsables de las computadoras	Permite listar toda la información de los responsables de las computadoras por fecha autor y tema
RF12.	Ver detalles de los responsables de las computadoras	Permite visualizar todos los responsables de las computadoras en el sitio web por fecha, autor
RF13.	Modificar información de los responsables de las computadoras	Permite editar la información de los responsables de las computadoras
RF14.	Eliminar información	Permite eliminar un responsable de

	de los responsables de las computadoras	computadoras en el sitio web
RF15.	Filtrar información de los responsables de las computadoras	Permite filtrar la información de los responsables de las computadoras en el sitio web por fecha, autor y tema.
RF16.	Generar reporte de los responsables de las computadoras	Permite crear un reporte de los responsables de las computadoras
RF17.	Insertar información de los discos duros	Permite crear un disco duro en el sitio web
RF18.	Modificar información de los discos duros	Permite editar la información de discos duros en el sitio web
RF19.	Eliminar información de los discos duros	Permite eliminar la información de los discos duros en el sitio web
RF20.	Ver detalles de los Discos duros	Permite visualizar todos los discos duros en el sitio web por fecha ,tipo de disco duro y autor
RF21.	Insertar información de los mouses	Permite crear un mouse en el sitio web
RF22.	Modificar información de los mouses	Permite editar la información de los mouses en el sitio web
RF23.	Eliminar información de los mouses	Permite eliminar un mouse en el sitio web
RF24.	Ver detalles de los mouses	Permite visualizar todos los mouses en el sitio web por fecha ,tipo de mouse y autor
RF25.	Insertar información de los monitores	Permite crear un monitor en el sitio web
RF26.	Modificar información de los monitores	Permite editar un monitor en el sitio web
RF27.	Eliminar información de los monitores	Permite eliminar un monitor en el sitio web
RF28.	Ver detalles de los monitores	Permite visualizar todos los monitores en el sitio web por fecha ,tipo de monitor y autor
RF29.	Insertar información de los dispositivos de audio	Permite crear un dispositivo de audio en el sitio web
RF30.	Modificar información de los dispositivos de audio	Permite editar la información de los dispositivos de audio en el sitio web

RF31.	Eliminar información de los dispositivos de audio	Permite eliminar un dispositivo de audio en el sitio web
RF32.	Ver detalles de los dispositivos de audio	Permite visualizar todas los monitores en el sitio web por fecha ,tipo de monitor y autor
RF33.	Insertar información de los dispositivos de video	Permite crear un dispositivo de video en el sitio web
RF34.	Modificar información de los dispositivos de video	Permite editar la información de los dispositivos de video en el sitio web
RF35.	Eliminar información de los dispositivos de video	Permite eliminar un dispositivo de video en el sitio web
RF36.	Ver detalles de los dispositivos de video	Permite visualizar todos los dispositivos de video en el sitio web por fecha ,tipo de dispositivo de video y autor
RF37.	Insertar información de los dispositivos de red	Permite crear un dispositivo de red en el sitio web
RF38.	Modificar información de los dispositivos de red	Permite editar la información de los dispositivos de red en el sitio web
RF39.	Eliminar información de los dispositivos de red	Permite eliminar un dispositivo de red en el sitio web
RF40.	Ver detalles de los dispositivos de red	Permite visualizar todos los dispositivos de red en el sitio web por fecha ,tipo de dispositivo de red y autor
RF41.	Insertar información de los lectores de discos	Permite crear la información de los lectores de discos
RF42.	Modificar información de los lectores de discos	Permite editar la información de los lectores de discos en el sitio web
RF43.	Eliminar información de los lectores de discos	Permite eliminar información de un lector de disco en el sitio web
RF44.	Ver detalles de los lectores de discos	Permite visualizar todos los lectores de discos en el sitio web por fecha ,tipo de lector de disco y autor
RF45.	Insertar información de los microprocesadores	Permite crear un microprocesador en el sitio web

RF46.	Modificar información de los microprocesadores	Permite editar la información de los microprocesadores en el sitio web
RF47.	Eliminar información de los microprocesadores	Permite eliminar un microprocesador en el sitio web
RF48.	Ver detalles de los microprocesadores	Permite visualizar todos los microprocesadores en el sitio web por fecha ,tipo de microprocesador y autor
RF49.	Insertar información de las impresoras	Permite crear una impresora en el sitio web
RF50.	Modificar información de las impresoras	Permite editar una impresora en el sitio web
RF51.	Eliminar información de la impresoras	Permite eliminar una impresora en el sitio web
RF52.	Ver detalles de las impresoras	Permite visualizar todas las impresoras en el sitio web por fecha ,tipo de impresora y autor

2.3.1 Requisitos no funcionales

Los requisitos no funcionales (RNF) son propiedades o cualidades que el producto debe tener. Son esas características que posibilitan que el producto sea atractivo, usable, rápido, confiable. Normalmente están vinculados a requisitos funcionales, es decir, una vez que se conozca lo que el sistema debe hacer, se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido debe ser [CITATION Pre10 \l 3082]. Se asumirán requisitos de calidad teniendo en cuenta la ISO/IEC 25030. A continuación, en la Tabla 4 se listan los RNF.

TABLA: LISTADO DE REQUISITOS NO FUNCIONALES. FUENTE: ELABORACIÓN PROPIA.

Categoría	Número	Descripción
Restricciones de diseño	RNF1	La aplicación se desarrollará en Drupal como lenguaje de programación PHP, el servidor web Apache y se utilizará como gestor de bases de datos MySQL.
Software	RNF2	El sistema deberá poder ser ejecutado desde cualquier sistema operativo con navegador web

		Mozilla Firefox o Internet Explorer
Hardware	RNF3	La computadora que hará uso del sistema a desarrollar tendrá como CPU, i3, similar o superior
	RNF4	La computadora que hará uso del sistema a desarrollar debe tener la memoria RAM como mínimo 4 Gb o superior y disco duro con capacidad de 40 GB o superior
Usabilidad	RNF5	La solución debe tener una interfaz gráfica atractiva permitiendo el uso del sistema sin necesidad de mucho conocimiento informático y el acceso a todas sus funcionalidades de manera sencilla y directa.
	RNF6	Se utilizarán colores frescos, como el color azul, que no carguen el diseño y con buen contraste entre los mismos, niveles de profundidad en cuanto a clics serán lo menos posible, el aspecto de los botones muy bien definidos (son rectangulares con relleno de color azul) y sin sobrecarga de colores, ni letras.
Confiabilidad	RNF7	Comprobar que los datos proporcionados por los usuarios sean valores válidos.
	RNF8	Garantizar que el sistema esté disponible las 24 horas del día para garantizar el acceso en todo momento.
Eficiencia:	RNF9	Atender las peticiones en un tiempo estimado de un 1s.

2.3.2 Validación de Requisitos de Software

La validación de requisitos examina las especificaciones para asegurar que todas las especificidades del sistema han sido establecidas sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos, y que el resultado del trabajo se ajusta a los estándares establecidos para el proceso, el proyecto y el producto [CITATION Pre106 \l 3082]. El proceso de validación se considera de vital importancia dentro de la disciplina de requisitos, ya que:

- Asegura que los ingenieros de *software* entiendan el marco de trabajo y los objetivos.
- Verifica que los requerimientos sean comprensibles, constantes y finitos.
- Reduce los costos, al detectar errores antes del desarrollo o después que el sistema esté en uso.

En la presente investigación para la validación se adoptó la técnica de realización de prototipos, la cual permitió representar visualmente las condiciones solicitadas por el cliente. En los anexos se hace referencia a las Historias de Usuario, donde se detallan los prototipos obtenidos en la propuesta de solución.

2.4 Fase de Planeación

La metodología XP plantea la planeación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores. El plan de iteraciones, las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido.

2.4.1 Historias de usuarios, tareas de ingeniería y tarjetas CRC

2.4.2 Historias de usuarios

En las metodologías de desarrollo ágiles las historias de usuario se emplean para la definición inicial de los requisitos del sistema. Ofrecen una visión general de lo que se desea crear y describen funcionalidades deseadas desde la perspectiva del cliente, quién la necesita, cómo y porqué se va a utilizar.

A continuación, se presenta el listado de Historias de Usuario definido en el proyecto:

Historias de Usuario HU.

1. Iniciar sesión.
2. Gestionar información referente a las computadoras.

3. Gestionar información referente a los responsables de las computadoras.
4. Gestionar información referente a los discos duros.
5. Gestionar información referente a los mouses.
6. Gestionar información referente a los monitores.
7. Gestionar información referente a los dispositivos de audio.
8. Gestionar información referente a los dispositivos de video.
9. Gestionar información referente a los dispositivos de red.
10. Gestionar información referente a los lectores de discos.
11. Gestionar información referente a los microprocesadores.
12. Gestionar información referente a las impresoras.
13. Gestionar información referente a las particiones de discos
14. Gestionar información referente a los teclados.
15. Gestionar información referente a los scanners.
16. Gestionar información referente a las memorias.
17. Gestionar información referente a las UPS.

Algunas descripciones Historias de Usuario definidas:

Tabla 2: Iniciar sesión.

Historia de Usuario	
No: 1	Nombre: Iniciar sesión.
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):	
Usuario: Informático y Económico.	
Prioridad en el Negocio: Alta	Riesgo de desarrollo: Medio
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: Para que el usuario interactúe con la aplicación se solicita su	

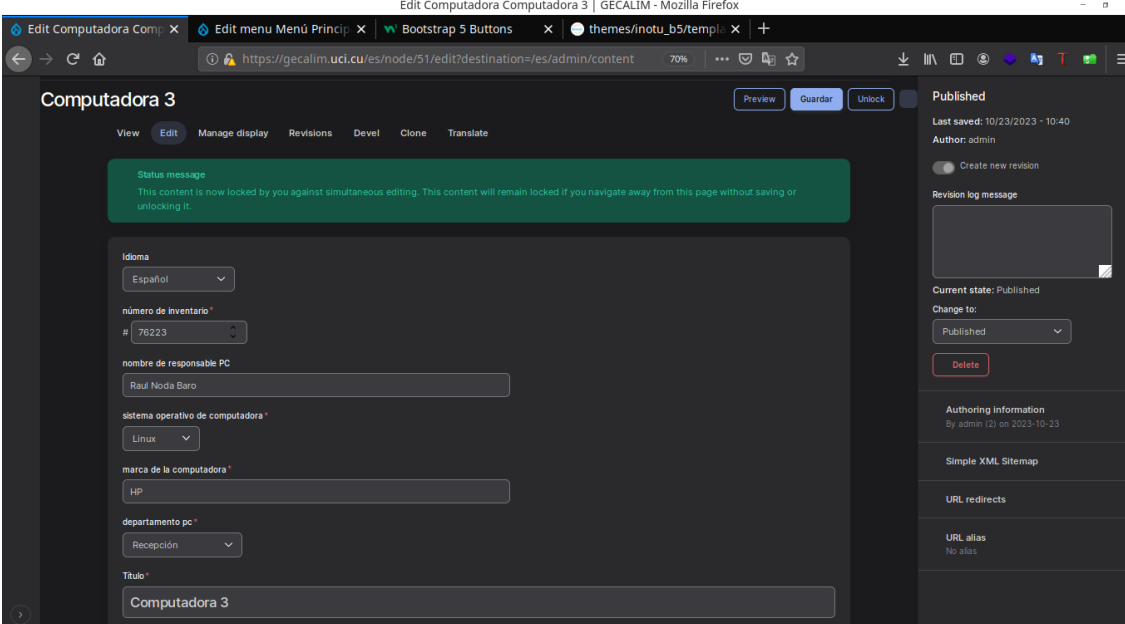
nombre de usuario y su clave para tener acceso a los contenidos que corresponden al rol que ocupa en el sistema.

Observación: Los usuarios deben tener sus datos de inicio de sesión en la respectiva base de datos de la aplicación.

Tabla 3: Gestionar información referente a las computadoras.

Historia de Usuario	
No:2	Nombre: Gestionar información referente a las computadoras.
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):	
Usuario: Informático.	
Prioridad en el Negocio: Alta	Riesgo de desarrollo: Medio
Puntos de Estimación: 2	Iteración Asignada: 1
Descripción: El sistema debe permitir a los usuarios mencionados realizar las siguientes funcionalidades respecto a las computadoras: crear, listar, eliminar, modificar, ver detalles, filtrar por campos y generar reportes.	
Observación: Los usuarios deben estar autenticados.	

Prototipo



Edit Computadora Computadora 3 | GECALIM - Mozilla Firefox
 https://gecalim.uci.cu/es/node/51/edit?destination=/es/admin/content

Computadora 3 [Preview] [Guardar] [Unlock]

View Edit Manage display Revisions Devel Clone Translate

Status message
This content is now locked by you against simultaneous editing. This content will remain locked if you navigate away from this page without saving or unlocking it.

Idioma: Español

número de inventario*: # 76223

nombre de responsable PC: Raul Noda Baro

sistema operativo de computadora*: Linux

marca de la computadora*: HP

departamento pc*: Recepción

Título*: Computadora 3

Published
Last saved: 10/23/2023 - 10:40
Author: admin
Create new revision

Revision log message

Current state: Published
Change to: Published
Delete

Authoring information
By admin (2) on 2023-10-23

Simple XML Sitemap

URL redirects

URL alias
No alias

Nota: Los otros prototipos de importancia se encuentran en los anexos.

Tabla 4: Gestionar información referente a los responsables de las computadoras.

Historia de Usuario	
No:3	Nombre: Gestionar información referente a los responsables de las computadoras.
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre): 2.Computadoras	
Usuario: Informático.	
Prioridad en el Negocio: Alta	Riesgo de desarrollo: Medio
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: El sistema debe permitir a los usuarios mencionados realizar las siguientes funcionalidades respecto a los responsables de las computadoras: crear, listar, eliminar, modificar, ver detalles, filtrar por campos y generar reportes.	
Observación: Los usuarios deben estar autenticados.	

2.4.3 Plan de Iteraciones (PI)

La planificación de la iteración es el proceso de discutir y planificar el siguiente ciclo, fase o iteración de una aplicación de *software* que está en desarrollo. Se lleva a cabo mediante una reunión de todo el equipo de desarrollo de *software* en el punto de partida de cada iteración para planificar formalmente los procesos técnicos y no técnicos [CITATION Jen22 \l 23562].

Tabla: Plan de iteraciones por Historia de Usuario.

Iteración	Descripción de la iteración.	Orden de la HU a implementar.	Duración total.
1	En esta iteración se implementarán las HU con prioridad Alta.	HU-1, HU-2, HU-3, HU-4	4 semanas
2	En esta iteración se implementarán las HU con prioridad Media.	HU-5, HU-6, HU-7, HU-8, HU-9	4 semanas
3	En esta iteración se implementarán las HU con prioridad Baja.	HU-10, HU-11, HU-12	3semanas

2.4.4 Tarjetas CRC

Las tarjetas CRC (Cargo o Clase, Responsabilidad y Colaboración) representan objetos. En el desarrollo del diseño de la aplicación se utilizaron dichas tarjetas, las cuales permiten el desarrollo orientado a objetos. Cada tarjeta refleja: nombre de la clase, las responsabilidades que posee y las clases que colaboran con la misma para cumplir dichas responsabilidades.

Tabla 5: Tarjeta CRC Gestionar Usuario.

Gestionar Usuario	
Responsabilidades	Colaboradores
Iniciar sesión	Rol Usuario

Tabla 6: Tarjeta CRC Gestionar Responsable.

Gestionar Responsable	
Responsabilidades	Colaboradores
Insertar información de responsable. Modificar información de responsable. Eliminar información de responsable. Listar información de responsables. Filtrar información de responsables. Ver detalles de responsable. Generar reporte de responsables.	Iniciar sesión. Responsable. Computadora.

Tabla 7: Tarjeta CRC Gestionar Computadora.

Gestionar Computadora	
Responsabilidades	Colaboradores
Insertar información de computadora. Modificar información de computadora. Eliminar información de computadora. Listar información de computadoras. Filtrar información de computadoras. Ver detalles de computadora.	Iniciar sesión. Computadora. Departamento. Tipo de equipo. Tipo de Sistema Operativo.

Generar reporte de computadoras.

Tabla 8: Tarjeta CRC Gestionar Impresora.

Gestionar Impresora	
Responsabilidades Insertar información de impresora. Modificar información de impresora. Eliminar información de impresora. Listar información de impresoras. Filtrar información de impresora. Ver detalles de impresora. Generar reporte de impresoras.	Colaboradores Iniciar sesión. Impresora. Computadora. Estado.

2.4.5 Tareas de ingeniería

Las tareas de ingeniería describen las tareas que se realizan sobre el proyecto. Las tareas tienen relación con una historia de usuario y pueden ser de varios tipos: desarrollo, corrección y mejora.

Representación detallada de algunas tareas de ingeniería definidas:

Tabla 9: Insertar información de computadora.

Tarea de Ingeniería	
No:1	Historia de Usuario (Nro. y Nombre): 2
Nombre Tarea: Insertar información de computadora	
Tipo de Tarea : Desarrollo	Puntos Estimados:1
Fecha Inicio:20/9/2023	Fecha Fin: 20/9/2023
Programador Responsable: Raúl Noda Baró.	
Descripción: Se implementa la funcionalidad que permita agregar los datos de una nueva computadora al sistema.	

Tabla 10: Modificar información de computadora.

Tarea de Ingeniería	
No:2	Historia de Usuario (Nro. y Nombre): 2
Nombre Tarea: Modificar información de computadora.	
Tipo de Tarea : Desarrollo	Puntos Estimados:1

Fecha Inicio: 20/9/2023	Fecha Fin: 20/9/2023
Programador Responsable: Raúl Noda Baró.	
Descripción: Se implementa la funcionalidad que permita modificar los datos de una computadora existente en el sistema.	

Tabla 11: Eliminar información de computadora.

Tarea de Ingeniería	
No: 3	Historia de Usuario (Nro. y Nombre): 2
Nombre Tarea: Eliminar información de computadora.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 20/9/2023	Fecha Fin: 20/9/2023
Programador Responsable: Raúl Noda Baró.	
Descripción: Se implementa la funcionalidad que permita eliminar del sistema de una computadora determinada.	

Tabla 12: Listar información de computadoras.

Tarea de Ingeniería	
No: 4	Historia de Usuario (Nro. y Nombre): 2
Nombre Tarea: Listar información de computadoras.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 20/9/2023	Fecha Fin: 20/9/2023
Programador Responsable: Raúl Noda Baró.	
Descripción: Se implementa la funcionalidad que genera el listado de registros de las computadoras existentes en el sistema.	

2.5 Arquitectura del software y patrones de diseño

2.5.1 Arquitectura de software

El diseño de la arquitectura es el proceso por el cual se define una solución para los requisitos técnicos y operacionales del mismo. Este proceso define qué componentes forman el *software*, cómo se relacionan entre ellos, y cómo mediante su interacción llevan a cabo la funcionalidad especificada, cumpliendo con los criterios previamente establecidos; como seguridad, disponibilidad, eficiencia o usabilidad [CITATION Alm22 \l 23562].

Al utilizar el CMS Drupal para el desarrollo del portal web, la arquitectura y los patrones son heredados de este. En lugar de considerar el sitio web como un conjunto de páginas interrelacionadas, Drupal estructura los contenidos en una

serie de elementos básicos. No posee una arquitectura lineal, sino que se estructura en diferentes capas, por lo que para el desarrollo del sistema se propone el trabajo con la arquitectura n-capas. Estas capas son los datos, módulos, bloques y menús, permisos de usuario y plantillas. Seguidamente en la Ilustración 1 se representa la arquitectura utilizada.

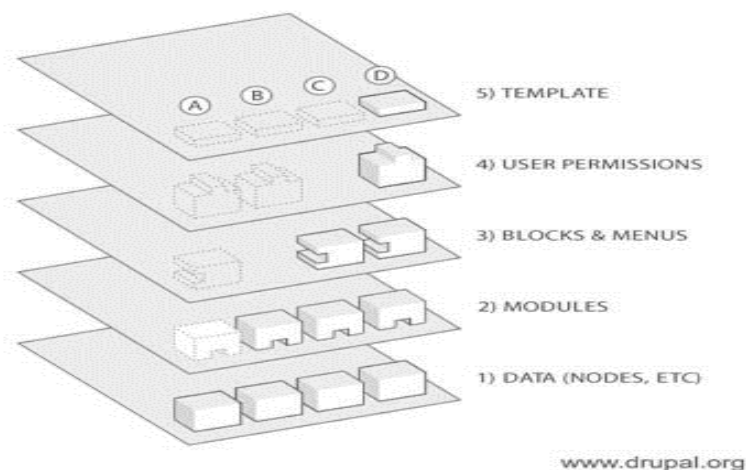


Ilustración 1: N-capas en Drupal. Fuente: Google

La primera capa representa la información almacenada en los Nodos (tipos de contenido a publicar ejemplo noticias, avisos, adquisiciones). En la segunda se encuentran los Módulos que son utilizados en la aplicación (ejemplos los módulos ctool, views, jcarousel), que son los elementos que operan sobre los Nodos y otorgan funcionalidad a Drupal permitiendo incrementar sus capacidades o adaptarlas a las necesidades de cada sitio web. La tercera esta descrita por los Bloques y Menús que permiten estructurar y organizar los contenidos en la página web (ejemplo: los menús ¿Quiénes Somos?, Enlaces y bloque de adquisiciones, Video). La cuarta capa recoge los Permisos de Usuarios donde Drupal dispone de un registro de usuarios y de roles que permiten especificar qué tareas pueden realizar y a qué contenidos puede acceder cada tipo de usuario (ejemplos: roles de administrador, editor y usuario). La última capa maneja las Plantillas y es la que establece la apariencia gráfica o estilo de la información que se le muestra al usuario.

2.5.2 Patrones de diseño

Un patrón de diseño permite describir un problema que ocurre reiteradas veces en nuestro entorno y el núcleo de la solución del problema, de tal forma que puede reutilizarse continuamente. El framework de desarrollo Drupal emplea patrones en su diseño, a continuación, se describen.

Patrones GRASP

Patrones GRASP significan según sus siglas en inglés General Responsibility Assignment Software Patterns, es decir, patrones generales de asignación de responsabilidades. Se utilizan en la programación orientada a objetos (POO), y el cómo definir estos objetos/clases será la aplicación de cada patrón[CITATION 22FE \l 3082].

Creador: Este patrón implica que un objeto debe responsabilizarse de crear otros: si contiene o agrega varios objetos del tipo de los creados, si se encarga de registrar objetos del tipo de los creados, si utiliza mucho los objetos creados o si contiene los datos para crear los del tipo creado. En el sistema, la clase Manager.php es la encargada de crear instancias (Ver Ilustración 2).

```
public static function initialize(  
    Behavior $behaviour,  
    Resolvable $resolver = null,  
    Collectable $collection = null  
) : self {  
    if (self::$instance === null) {  
        self::$instance = new self($behaviour, $resolver, $collection);  
        return self::$instance;  
    }  
    throw new \LogicException(  
        'Manager can only be initialized once',  
        1535189871  
    );  
}
```

ILUSTRACIÓN 2: PATRÓN CREADOR. FUENTE: DRUPAL

Alta cohesión: La alta cohesión la tiene un objeto si todo lo que hace está bien delimitado dentro del mismo objeto. Además, para tener alta cohesión, debe de ser un objeto cuanto más pequeño mejor (Ver Ilustración 3). Este concepto está relacionado inversamente con el acoplamiento, es decir, si aumenta la cohesión, suele disminuir el acoplamiento, y viceversa[CITATION 223 \l 3082].


```
!function(e,t){"object"==typeof exports&&"undefined"!=typeof module?t(exports):  
"function"==typeof define&&define.amd?define(["exports"],t):t((e="undefined"!=typeof globalThis?globalThis:e||  
self).Popper=({}))(this,(function(e){"use strict";function t(e){if(null==e)return window;if("[object Window]"!==e.  
toString()){var t=e.ownerDocument;return t&&t.defaultView||window}return e}function n(e){return e instanceof t(e).  
Element||e instanceof Element}
```

Ilustración 3: Alta cohesión. Fuente: Drupal

Bajo acoplamiento: el acoplamiento mide la fuerza con que una clase está conectada a otra, de esta forma una clase con bajo acoplamiento debe tener un número mínimo de dependencia con otras clases. Este patrón se tuvo presente debido a la importancia que se le atribuye a realizar un diseño de clases independientes que puedan soportar los cambios de una manera fácil y que a su vez permitan la reutilización. El patrón se evidencia en cada una de las clases diseñadas para la herramienta.

Patrones (GoF)

Estos patrones son conocidos como los patrones de la pandilla de los cuatro (GoF, gang of four), deben su nombre al grupo compuesto por sus creadores: Erich Gamma, Richard Helm, Ralph Jhonson y John Vlissides. Estos patrones se destacan por proporcionar elementos reusables en el diseño de sistemas de *software* y por su amplia estandarización de diseño. En el desarrollo de la presente investigación, se utilizaron los siguientes patrones:

Instancia única (Singleton): Presenta un mecanismo para limitar el número de instancias de una clase.

Decorator: Drupal hace un amplio uso del patrón decorador. El polimorfismo de los objetos del nodo es solo una pequeña parte del poder del sistema de nodos. Más interesante es el uso de varios ganchos de nodo (`hook_node_load()`, `hook_node_view()`, etc.), que permiten a los módulos arbitrarios extender el comportamiento de todos los nodos[CITATION Dru19 \l 2057].

Esta característica permite agregar una amplia variedad de comportamientos a los nodos sin la necesidad de subclasificar. Por ejemplo, un nodo de historia básica tiene solo unos pocos datos asociados: título, autor, cuerpo, avance y un puñado de metadatos. Una necesidad común es que los archivos se carguen y

se adjunten a un nodo, por lo que se podría diseñar un nuevo tipo de nodo que tuviera las características del nodo de la historia más la capacidad de adjuntar archivos. El módulo de carga de Drupal satisface esta necesidad de una manera mucho más modular mediante el uso de la API de nodo para otorgar a cada nodo que le solicite la capacidad de tener archivos adjuntos.

Este comportamiento podría ser imitado por el uso de decoradores, envolviéndolos alrededor de cada objeto de nodo. Más simplemente, los lenguajes que admiten categorías, como Objective-C, podrían aumentar la clase base común de todos los objetos de nodo para agregar el nuevo comportamiento. La implementación de Drupal es una simple ramificación del sistema de enlace y la presencia de `node_invoke ()`

Observer: Este patrón de Observador es generalizado en todo Drupal, ya que muchos de los ganchos de Drupal esencialmente permiten que los módulos se registren como observadores de los objetos de Drupal. Por ejemplo, cuando se realiza una modificación a un vocabulario en el sistema de taxonomía de Drupal, se llama a un gancho de taxonomía como `hook_taxonomy_vocabulary_update ()` en todos los módulos que lo implementan. Al implementar el gancho, los módulos se han registrado como observadores del objeto de vocabulario; cualquier cambio en él se puede actuar según corresponda [CITATION CAG13 \l 3082].

Conclusiones del capítulo

A partir del análisis y diseño del sistema se arribó a las siguientes consideraciones:

- La correcta identificación de los requisitos del sistema, la caracterización del problema, el uso del patrón de arquitectura de software n-capas, unido al empleo de los patrones de diseño GRASP y GoF, permitieron obtener un diseño robusto de la aplicación.
- Las definiciones de las Tarjetas CRC aportaron en la identificación de las responsabilidades de las clases y las colaboraciones de las mismas.

Capítulo 3: Implementación y validación del Sistema Web para la gestión y control de los medios informáticos en la Empresa de Acueducto y Alcantarillado de Matanzas (UEB) Calimete.

El presente capítulo tiene como objetivo esencial presentar los resultados de la implementación y pruebas realizadas a la aplicación como herramienta de validación de la solución. Se describen las tareas de ingenierías, los estilos de programación utilizados, las pruebas unitarias realizadas al código y las pruebas de aceptación realizadas en conjunto con el cliente.

3.1 Tareas de Ingeniería

La Metodología XP define como tareas de ingeniería a todas las tareas que se realizan en cada iteración. A cada una de las tareas se le asigna un programador, el cual, será responsable de ejecutarla.

A continuación, en la Tabla 13 se muestra las tareas de ingeniería definidas para la realización de las historias de usuario.

Tabla 13: Tareas de ingeniería. Fuente: Elaboración propia.

Iteración	Historias de usuario	Tareas
-----------	----------------------	--------

1	Iniciar sesión (HU-1)	• Realizar la autenticación de usuarios.
	Gestionar información referente a las computadoras (HU-2)	• Crear prueba unitaria • Adiciona el contenido de la computadora • Crea la vista de las computadoras • Ejecutar prueba unitaria
	Gestionar información referente a los responsables de las computadoras (HU-3)	• Crear prueba unitaria • Adicionar el contenido de responsable • Crea la vista de los responsables • Ejecutar prueba unitaria
	Gestionar información referente a los discos duros (HU-4)	• Crear prueba unitaria • Adicionar el contenido de dispositivos asociados de tipo discos duros • Crea la vista de discos duros • Ejecutar prueba unitaria
2	Gestionar información referente a los mouses (HU-5)	• Crear prueba unitaria • Adicionar el contenido de dispositivos asociados de tipo mouses • Crea la vista de mouses • Ejecutar prueba unitaria

	Gestionar información referente a los monitores (HU-6)	• Crear prueba unitaria • Adiciona el contenido de dispositivos asociados de tipo monitores • Crea la vista de particiones de discos • Ejecutar prueba unitaria
	Gestionar información referente a los dispositivos de audio (HU-7)	• Crear prueba unitaria • Adiciona el contenido de dispositivos asociados de tipo dispositivos de audio • Crea la vista de dispositivos de audio • Ejecutar prueba unitaria
	Gestionar información referente a los dispositivos de video (HU-8)	• Crear prueba unitaria • Adiciona el contenido de dispositivos asociados de tipo dispositivos de video • Crea la vista de dispositivos de video • Ejecutar prueba unitaria
	Gestionar información referente a los dispositivos de red (HU-9)	• Crear prueba unitaria • Adiciona el contenido de dispositivos asociados de tipo dispositivos de red • Crea la vista de

		dispositivos de red
		• Ejecutar prueba unitaria
3	Gestionar información referente a los lectores de discos (HU-10)	• Crear prueba unitaria • Adiciona el contenido de dispositivos asociados de tipo lectores de discos • Crea la vista de lectores de discos Ejecutar prueba unitaria
	Gestionar información referente a los microprocesadores (HU-11)	• Crear prueba unitaria • Adiciona el contenido de dispositivos asociados de tipo microprocesadores • Crea la vista de microprocesadores • Ejecutar prueba unitaria
	Gestionar información referente a las impresoras (HU-12)	• Crear prueba unitaria • Adicionar el contenido de dispositivos asociados de tipo impresoras • Crea la vista de impresoras • Ejecutar prueba unitaria

3.2 Estilos de programación

Los estilos de programación son reglas que se utilizan para estructurar el código fuente de una aplicación en desarrollo. En este epígrafe se describirán

los estándares y reglas de programación utilizadas en la implementación del sistema.

Nombre de las clases: El estilo de capitalización utilizado para la notación de las clases es propio, cada subpalabra que conforme el nombre de una función comenzará con una letra minúscula.

Se definieron un grupo de reglas con el fin de lograr una mayor claridad en el código:

- Utilizar nombres descriptivos para los nombres de las estructuras, atributos y parámetros.
- Los comentarios deben estar en el mismo nivel del código.
- Se deberá realizar una sola declaración por línea con el fin de facilitar los comentarios.
- No debe haber espacios en blanco entre los nombres de los métodos y el paréntesis (que abre su lista de parámetros).
- Cuando una expresión no entra en una línea, se rompe de acuerdo con estos principios:
- Romper después de una coma.
- Romper antes de un operador.

3.3 Validación del diseño

La validación del diseño mediante las métricas de diseño, permite medir de forma cuantitativa la calidad de los atributos internos del software, de forma tal que pueden ayudar al desarrollador a juzgar la calidad de un diseño a nivel de componente.

Para la validación del sistema en cuestión se utilizaron las métricas siguientes:

- Métrica de relaciones entre clases (RC).
- Métrica de tamaño operacional de las clases (TOC)

3.3.1 Métrica relaciones entre clases

Utilizando la métrica RC el resultado es dado por el número de relaciones de uso que se establecen entre una clase y las demás clases existentes y se

evalúa a partir de atributos de calidad; acoplamiento, complejidad de mantenimiento, reutilización y cantidad de pruebas. [CITATION Pre10 \l 3082]

Cada atributo de calidad se describe a continuación:

- **Acoplamiento:** un aumento del RC implica un aumento del acoplamiento de la clase.
 - **Complejidad de mantenimiento:** un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
 - **Reutilización:** un aumento del RC implica una disminución en el grado de reutilización de la clase.
 - **Cantidad de pruebas:** un aumento del RC implica un aumento de la cantidad de pruebas necesarias para probar una clase.
- Para aplicar la métrica RC es necesario categorizar cada una de las clases según la cantidad de relaciones que esta contenga. Para obtener un nivel óptimo de relación entre clases, el valor obtenido al aplicar dicha métrica debe ser directamente proporcional al acoplamiento y a la complejidad de mantenimiento; además debe ser inversamente proporcional al nivel de reutilización del código. A continuación, en la Tabla 14 se describe el atributo de calidad con sus categorías y criterios de evaluación.

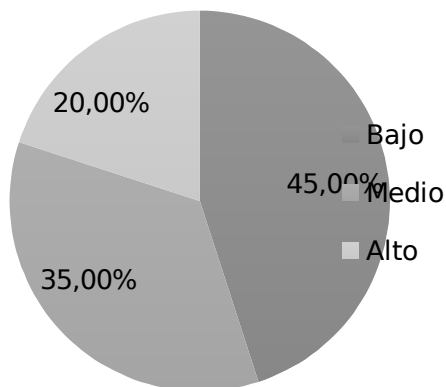
Tabla 14: Atributos de calidad de la métrica RC. Fuente: Elaboración propia

Atributo	Categoría	Criterio
Acoplamiento	Ninguno	RC = 0
	Bajo	RC = 1
	Medio	RC = 2
	Alto	RC > 2
Complejidad de Mantenimiento	Baja	RC Promedio
	Media	Promedio $\leq RC \leq 2$ *
	Alto	RC > 2 * Promedio
Reutilización	Baja	RC > 2 * Promedio

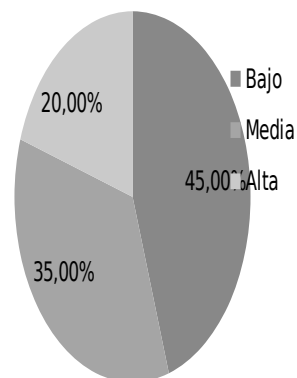
	Media	$\text{Promedio} < \text{RC} \leq 2 * \text{Promedio}$
	Alto	$\text{RC} \leq \text{Promedio}$
Cantidad de pruebas	Baja	$\text{RC} \leq \text{Promedio}$
	Media	$\text{Promedio} \leq \text{RC} < 2 * \text{Promedio}$
	Alto	$\text{RC} \geq 2 * \text{Promedio}$

La aplicación del instrumento de evaluación de la métrica RC en el diseño del sistema propuesto, arrojó los resultados expuestos en la Ilustración 4.

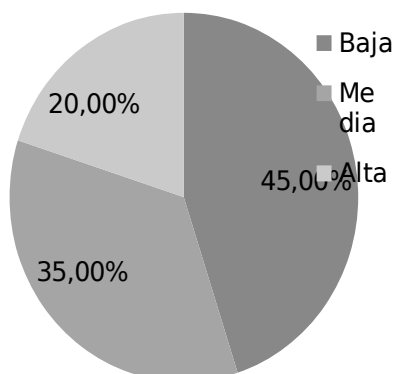
Acoplamiento



Complejidad de mantenimiento



Cantidad de Pruebas



Reutilización

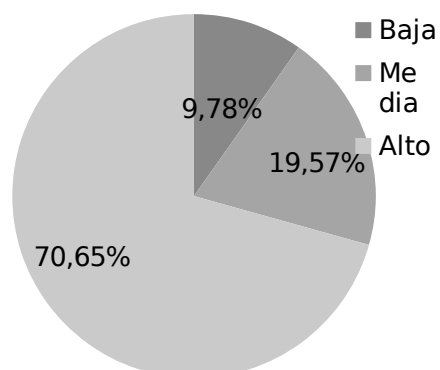


ILUSTRACIÓN 4: RESULTADOS DE LA MÉTRICA RC. FUENTE: ELABORACIÓN PROPIA

Luego de evaluar los resultados obtenidos de cada uno de los atributos de calidad, se obtuvieron los siguientes resultados:

- El 45% de las clases tienen un bajo acoplamiento.
- El 45% de las clases poseen una baja complejidad de mantenimiento
- El 45% de las clases tienen una baja de cantidad de pruebas
- El 71% de las clases poseen una alta reutilización.

Como se puede observar en la figura el acoplamiento posee un nivel bajo por lo que existe poca dependencia entre las clases, lo que posibilita una alta probabilidad de reutilización. También se puede apreciar que existe un medio nivel de complejidad de mantenimiento por lo que a la hora de mejorar los métodos y demás operaciones no es necesario realizar una gran cantidad de pruebas.

3.3.2. Métrica tamaño operacional de clases.

Las métricas basadas en el Tamaño Operacional de Clases (TOC) están dadas por el número de métodos asignados a una clase, así como la cantidad de atributos q esta posea y el promedio que presenta el sistema en su totalidad. Se evalúa a partir de los siguientes atributos de calidad. [CITATION Pre10 \l 3082]

Responsabilidad: un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.

Complejidad de implementación: un aumento del TOC implica un aumento de la complejidad de implementación de la clase.

Reutilización: un aumento del TOC implica una disminución del grado de reutilización de la clase.

Una vez analizado el indicador tamaño de clase, si el valor resultante tiende al crecimiento, es probable que la clase posea un alto grado de Responsabilidad; en consecuencia, el nivel de Reutilización sería mínimo y la implementación altamente compleja. A continuación, en la Tabla 15 se describe el atributo de calidad con sus categorías y criterios de evaluación:

TABLA 15: ATRIBUTOS DE CALIDAD DE LA MÉTRICA TOC. FUENTE: ELABORACIÓN PROPIA

Atributo de calidad	Categoría	Criterio
Responsabilidad	Bajo	$TOC \leq \text{Promedio}$ (Promedio)
	Medio	$\text{Promedio} \leq TOC \leq 2 * \text{Promedio}$
	Alto	$TOC > 2 * \text{Promedio}$
Complejidad de implementación	Baja	$TOC \leq \text{Promedio}$
	Media	$\text{Promedio} \leq TOC \leq 2 * \text{Promedio}$
	Alto	$TOC > 2 * \text{Promedio}$
Reutilización	Baja	$TOC > 2 * \text{Promedio}$
	Media	$\text{Promedio} \leq TOC \leq 2 * \text{Promedio}$
	Alto	$TOC \leq \text{Promedio}$

La aplicación del instrumento de evaluación de la métrica RC en el diseño del sistema propuesto, arrojó los resultados expuestos en la Ilustración 5.

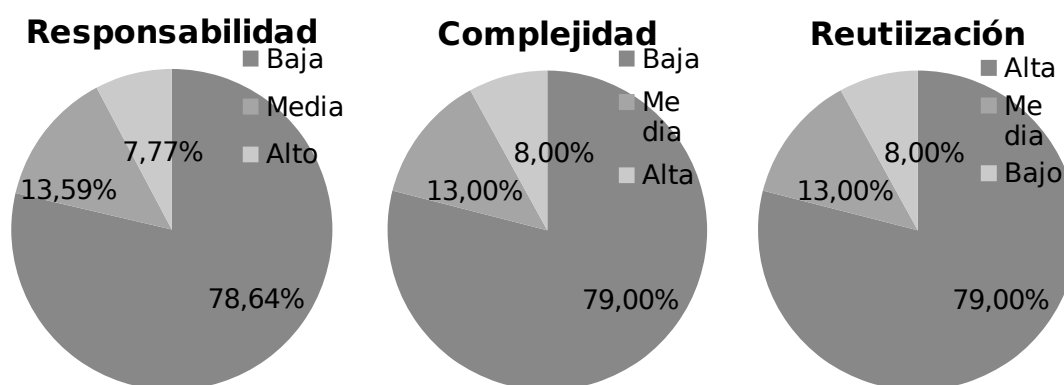


ILUSTRACIÓN 5: RESULTADOS DE LA MÉTRICA TOC. FUENTE: ELABORACIÓN PROPIA

Una vez evaluados los resultados obtenidos de cada uno de los atributos de calidad, se obtuvieron los siguientes resultados:

- El 79% de las clases poseen una baja responsabilidad de prueba.
- El 79% de las clases poseen una baja complejidad de prueba.
- El 79 % de las clases poseen una alta reutilización.

Obteniendo un bajo porcentaje de responsabilidad y complejidad de las clases se demuestra la calidad del diseño. Unidos a un alto grado de reutilización de las clases se facilita en gran medida la implementación de las mismas

3.4 Pruebas de software

En el proceso de desarrollo de *software* una de las fases más importante es la fase de pruebas, debido a que a través de ella se validan que los requisitos del *software* han sido cumplidos y garantiza que el sistema posee la calidad requerida.

3.4.1 Pruebas unitarias

Las pruebas unitarias son diseñadas por los desarrolladores con el objetivo de verificar el código y detectar posibles errores durante la ejecución del *software* a las funcionalidades de las clases. Todos los módulos deben pasar este tipo de pruebas antes de ser liberados o publicados. Por otra parte, en XP las pruebas deben ser definidas antes de realizar el código. Que todo código liberado pase correctamente las pruebas unitarias, es lo que habilita que funcione la propiedad colectiva del código. Cuando se encuentra un error éste debe ser corregido inmediatamente, y se deben tener precauciones para que errores similares no vuelvan a ocurrir. Asimismo, se generan nuevas pruebas para verificar que el error haya sido resuelto.

La técnica de ruta básica tiene como objetivo comprobar que cada camino se ejecute independiente de un componente o programa, obteniéndose una medida de la complejidad lógica del diseño. Esta técnica de prueba debe ser utilizada para evaluar la efectividad de los métodos asociados a una clase, con el objetivo de asegurar que cada camino independiente sea ejecutado por lo menos una vez en el sistema. [CITATION Pre10 \l 3082]

Pressman propone como estrategia para aplicar la ruta básica, realizar un análisis de la complejidad ciclomática de cada procedimiento que componen las clases del sistema; una vez concluido este paso se selecciona el método con valor de contener errores, además de que ofrece una medida del número

de pruebas que deben diseñarse para validar la correcta implementación de una determinada función. Esta métrica se calcula sobre un grafo y se puede realizar mediante tres formas distintas:

1. $V(G) = R$
2. $V(G) = E - N + 2$
3. $V(G) = P + 1$

Conociendo que:

- G: Grafo de flujo (grafo).
- R: El número de regiones contribuye a estimar el valor de la complejidad ciclomática.
- E: Número de aristas.
- $V(G)$: Complejidad ciclomática.
- N: Número de nodos del grafo.
- P: Número de nodos predicados¹ incluidos en el grafo.

El siguiente fragmento de código muestra el resultado de camino básico (ver ilustraciones 6 y 7):

```

;(function(root, factory) {
  // https://github.com/umdjs/umd/blob/master/returnExports.js
  if (typeof exports == 'object') {
    // For Node.js.
    module.exports = factory(root);
  } else if (typeof define == 'function' && define.amd) {
    // For AMD. Register as an anonymous module.
    define([], factory.bind(root, root));
  } else {
    // For browser globals (not exposing the function separately).
    factory(root);
  }
})(typeof global != 'undefined' ? global : this, function(root) {

  if (root.CSS && root.CSS.escape) {
    return root.CSS.escape;
  }
});
  
```

ILUSTRACIÓN 6: RESULTADO DE CAMINO BÁSICO. FUENTE. DRUPAL.

$V(G) = E - N + 2$	$V(G) = P + 1$	$V(G) = R = 4$	N: Total de nodos
$V(G) = 12 - 10 + 2$	$V(G) = 3 + 1$		E: Total de aristas
$V(G) = 2 + 2$	$V(G) = 4$		P: Nodos
			predicados
			R: Regiones

 $V(G) = 4$

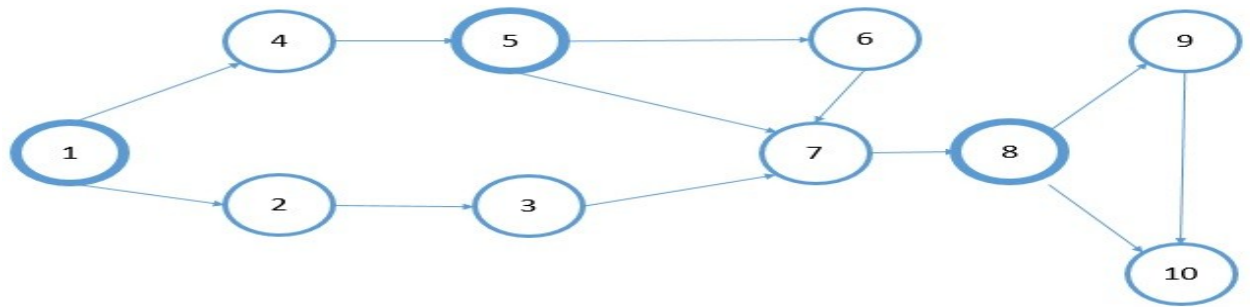


ILUSTRACIÓN 7: GRAFO DE FLUJO. FUENTE. ELABORACIÓN PROPIA.

TABLA 16: CAMINOS IDENTIFICADOS. FUENTE: ELABORACIÓN PROPIA.

Caminos identificados
1-2-3-7-8-9-10
1-2-4-6-7-8-10
1-2-4-5-7-8-9-10
1-2-4-5-7-8-10

El cálculo efectuado mediante las fórmulas antes presentadas muestra una complejidad ciclomática de valor cuatro, de manera que existen cuatro posibles caminos por donde el flujo puede circular, este valor representa el número máximo de casos de pruebas para el procedimiento tratado (Ver Tabla 16).

El método de caja negra se centra en los requisitos funcionales del *software*. O sea, la prueba de caja negra permite al ingeniero del *software* obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La prueba de caja negra no es una alternativa a las técnicas de prueba de caja blanca. Más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los detectados por los métodos de caja blanca. La prueba de caja negra intenta encontrar errores de las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y de terminación. [CITATION Pre10 \l 3082]

Para la aplicación de esta técnica a la solución obtenida, se confeccionaron casos de prueba por cada HU. A continuación, en la Tabla 17 se representa el caso de prueba para la HU 2 Escenario Crear Computadora.

Tabla 17: Caso de prueba del escenario Crear Computadora. Fuente: Elaboración propia

Descripción	Variable 1	Respuesta del sistema	Flujo central
Permite crear una computadora en el sistema web	<i>V (Introducir valores en los campos.)</i>	El sistema muestra el mensaje "Computadora ".	1. Seleccionar en el menú Computadora 2. Seleccionar la opción Crear computadora. 3. <i>Introducir valores en el campo.</i> 4. Seleccionar la opción Enviar. 5. <i>El sistema agrega la inversión en mantenimiento a la base de datos.</i>
	<i>I (introducir un String o carácter extraño en los campos.)</i>	El sistema muestra el mensaje "Existen valores incorrectos. Verifíquelos".	
	<i>I(Dejar campos vacíos)</i>	El sistema muestra el mensaje "Los campos no pueden estar vacíos".	

Análisis de las pruebas unitarias

Luego de haber implementado las historias de usuario planificadas en la primera iteración del desarrollo del sistema se realizó la primera iteración de pruebas unitarias detectándose nueve no conformidades (NC). La mayor parte de estas, estaban asociadas a las de tipo interfaz y a la entrada de datos al sistema, de las cuales fueron analizadas y corregidas seis de estas. Luego de haber implementado las historias de usuario planificadas en la segunda iteración del desarrollo del sistema, corregidas ya todas las NC encontradas, se realizó la segunda iteración de pruebas unitarias detectándose tres NC, dos de las cuales estaban asociadas al manejo de las clases y a la visualización de los datos de salidas que muestra el sistema. Las no conformidades de la segunda iteración fueron corregidas. En la tercera no se obtuvo no conformidades dándose por concluidas las pruebas. La Ilustración 8 muestra los resultados de

aplicar el método de caja negra, teniendo en cuenta los tipos de NC identificados (interfaz, funcionalidad, redacción).

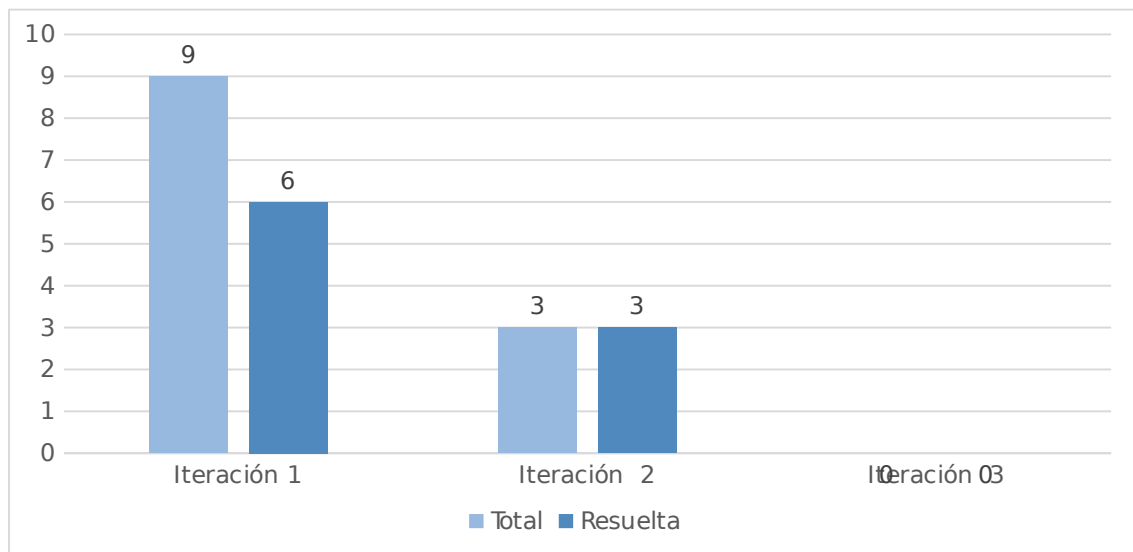


ILUSTRACIÓN 8: ANÁLISIS DE LAS PRUEBAS DE CAJA NEGRA. FUENTE: ELABORACIÓN PROPIA.

3.4.2 Pruebas de aceptación.

Las pruebas de aceptación tienen como objetivo fundamental validar el nivel de satisfacción del cliente con respecto al *software* desarrollado. Estas son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo del *software*. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta que pase correctamente todas las pruebas de aceptación.

A continuación, en la Tabla 18 se muestra un ejemplo de prueba de aceptación realizada al sistema. En la primera iteración resultó Fallida por validación de un dato. Se rectificó la NC y resultó Satisfactoria en la segunda iteración.

TABLA 18: CASO DE PRUEBA PARA EL REQUISITO "CREAR COMPUTADORA". FUENTE: ELABORACIÓN PROPIA

Caso de prueba de aceptación	
Código de caso de prueba: 3	Nombre historia de usuario: Crear computadora
Nombre de la persona que realiza el caso de prueba: Raúl Noda Baró.	

Descripción de la prueba: El sistema debe tener una opción crear computadora donde se introducirá la información sobre esta. La información de esta opción se podrá guardar siempre que tengan valores todos los atributos
Precondiciones: El usuario debe estar previamente registrado y autenticado en el sistema
Pasos: • El usuario acceder al sistema. • Ir a la pestaña agregar. • Llenar los campos correspondientes.
Resultados esperados: se agrega la computadora satisfactoriamente a la lista de computadoras
Evaluación:

Análisis de las pruebas de aceptación

En las pruebas de aceptación se realizaron tres iteraciones. Una vez implementadas las historias de usuario de la primera iteración de la implementación del sistema, se realizó la primera iteración de las pruebas de aceptación, en ella se detectó cinco no conformidades, cuatro de interfaz y una de validación, los cuales fueron corregidas. Una vez implementadas las historias de usuario de la segunda iteración, se desarrolló la segunda iteración de pruebas de aceptación detectándose tres no conformidades. Las no conformidades de la segunda iteración fueron corregidas. En la tercera iteración no se hallaron NC,

3.4.3 Prueba de usabilidad

Las pruebas de usabilidad aplicadas a la propuesta de solución se realizaron a través del documento “Lista de chequeo de usabilidad de sitios web”. Esta cuenta con 144 indicadores separados por 17 elementos de usabilidad.

Se efectuaron dos iteraciones. En la primera iteración quedó evaluada de regular la prueba, debido a que se encontraron 2 no conformidades críticas y 1 no conformidades no críticas, para un total de 3 no conformidades, con 3 elementos afectados: Visibilidad del sistema; Consistencia y estándares; Prevención de errores (Ver Ilustración 9). En la segunda iteración se solucionaron todas las no conformidades, resultando satisfactorio la prueba.

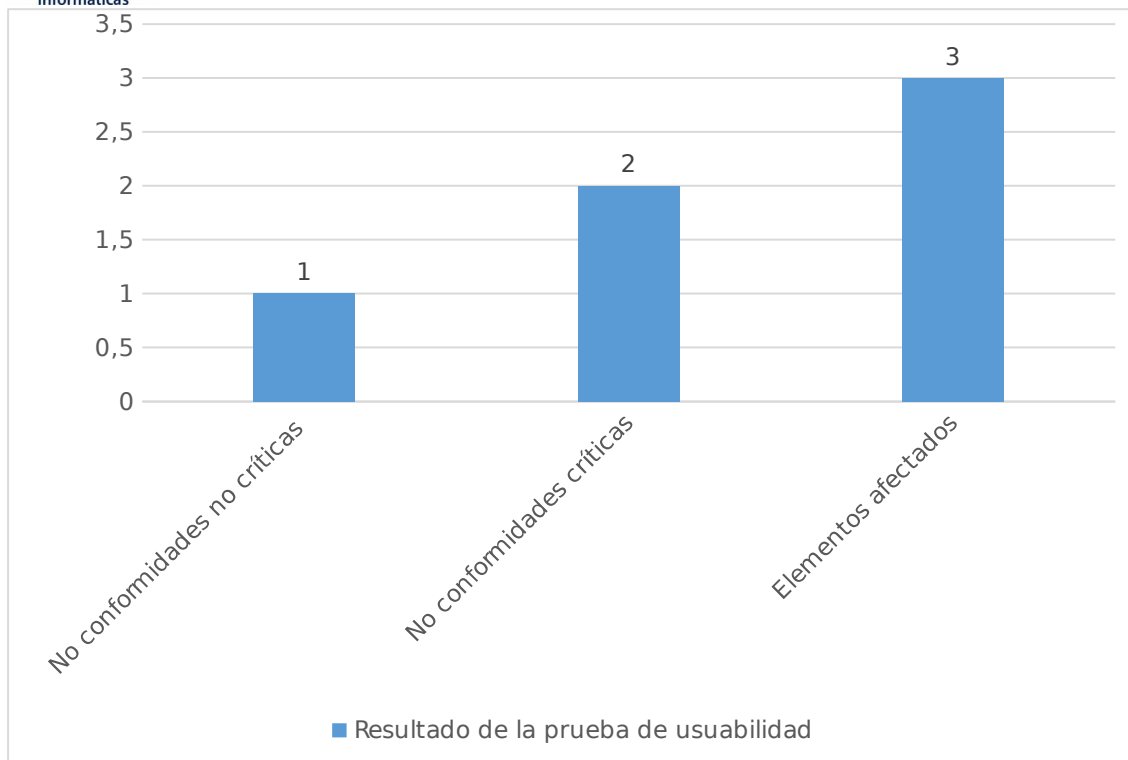


ILUSTRACIÓN 9: RESULTADO DE LA PRUEBA DE USABILIDAD. FUENTE: ELABORACIÓN PROPIA.

3.4.4 Pruebas carga y estrés

Las pruebas de carga y estrés se desarrollaron con la ayuda de la herramienta Apache JMeter 5.5, la cual permite medir el rendimiento de recursos como servicios web, lenguajes dinámicos y bases de datos, entre otros.

Estas se llevaron a cabo en un ambiente seleccionado utilizando un ordenador con las siguientes características: Sistema operativo Windows, Microprocesador Intel Core i3 2.40 GHz y memoria RAM 2GB.

Los resultados se muestran en la Ilustración 10.

Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
Pag Inicio	2563	3753	264	8189	584.81	0.00%	3.3/sec	100.33	0.43	30676.7
Pag Pregunta...	2541	3786	275	6869	534.47	0.00%	3.3/sec	105.66	0.47	32566.1
Pag login	2525	3809	243	6786	532.87	0.00%	3.3/sec	87.13	0.42	26970.5
Pag Cuestion...	2513	3822	582	7871	535.07	0.00%	3.3/sec	64.95	0.45	20160.2
Pag Mi cuenta	2505	3819	1415	8347	562.88	0.00%	3.3/sec	65.07	0.44	20171.0
Pag Aviso	2499	3831	1411	7719	588.57	0.00%	3.3/sec	70.63	0.57	21849.4
Pag Artículo	2496	3798	279	8832	565.28	0.00%	3.3/sec	81.57	0.45	25167.9
Pag Materiales	2485	3776	251	8181	551.03	0.00%	3.3/sec	68.83	0.52	21230.7
Total	20127	3799	243	8832	557.82	0.00%	26.3/sec	639.09	3.72	24884.3

Ilustración 10: Reporte resumen de las pruebas de carga y estrés. Fuente: JMeter.

Se espera que una vez instalado el sitio web en los servidores apropiados el rendimiento del mismo sea mejor porque estos servidores cuentan con mayores prestaciones. Con las condiciones de la PC cliente anteriormente descrito se obtiene un rendimiento de 3.3/sec por página visitada, donde el usuario ve la demora en la respuesta, pero el sistema responde correctamente.

Conclusiones parciales

En el presente capítulo se describen los elementos necesarios para completar el proceso de implementación y prueba del Sistema Web para la gestión y control de los medios informáticos en la Empresa de Acueducto y Alcantarillado de Matanzas (UEB) Calimete. Se describen las tareas de Ingeniería, así como los estilos de programación utilizados en la implementación del sistema. También se realiza el análisis de los resultados arrojados por las pruebas de *software* realizadas a la aplicación llegando a las siguientes consideraciones:

- Con la definición de las tareas de ingeniería se evitó la sobrecarga de trabajo y se agilizó el proceso.
- La aplicación de las pruebas unitarias permitió validar las funcionalidades críticas del sistema, mientras que las pruebas de aceptación en conjunto con el cliente posibilitaron comprobar la correcta implementación de las historias de usuarios definidas con anterioridad.
- Con la realización de las pruebas de *software* se llegó a la conclusión que la implementación satisface los requerimientos definidos por el cliente.

Conclusiones Generales

- El estudio de los referentes teóricos metodológicos que sustentan el desarrollo de Sitios web institucionales enfocados a la gestión de la información ratifica la pertinencia y necesidad del desarrollo de un sistema web para la empresa UEB Calimete.
- El análisis de las tendencias actuales de los Sitios web institucionales enfocados a la gestión de la información permitió determinar los elementos esenciales que debe tener el Sitio en cuanto a contenido, ergonomía, navegabilidad y servicios.
- El uso de un enfoque de desarrollo ágil con XP, así como de herramientas y tecnologías tales como el framework Drupal, el lenguaje de programación PHP y la herramienta Visual Studio Code facilitó el correcto desarrollo del sitio web.
- Los requisitos del sistema, la arquitectura n-capa y el uso de los patrones GRASP y GoF, permitieron obtener un diseño robusto, flexible y óptimo del sitio web.
- Con la realización de las pruebas de software se demostró que la implementación satisface los requisitos definidos por el cliente.

Recomendaciones

Para una futura versión del sitio web para la gestión y control de los medios informáticos en la Empresa de Acueducto y Alcantarillado de Matanzas (UEB) Calimete se recomienda:

- Actualizar los módulos de seguridad cada vez que la comunidad de desarrolladores de Drupal publiquen una nueva versión.
- Desarrollar y emplear una técnica de procesamiento de datos para la realización de estudios estadísticos respecto a los medios informáticos.
- Implementar las funcionalidades de la gestión de la información referente a las particiones de discos, memorias y UPS.

Bibliografía

- (s.f.). Obtenido de <https://code.visualstudio.com/docs/getstarted/themes>
- (2023). *MongoDB*, <https://www.mongodb.com>.
- Almeyda, Miguel. (5 de septiembre de 2022). Obtenido de <https://blog.continental.edu.pe/sistemas-informatica/2013/01/10/arquitectura-de-software-que-es-y-como-funciona/>
- Andrade Loor, M. J. (2018). <http://repositorio.espam.edu.ec/handle/42000/938>.
- Apache Friends. (2022). Obtenido de <https://www.apachefriends.org/es/index.html>
- Arzoo, M. K. (2017). NetBeans IDE 8.2. https://d1wqtxts1xzle7.cloudfront.net/53539590/IRJET-V4I4484-libre.pdf?1497610755=&response-content-disposition=inline%3B+filename%3DK_Means_algorithm_with_different_distanc.pdf&Expires=1695703211&Signature=So5AbRSqan46AmiE4mcQ85RKIOVUhk8Ekwru08ycFTugDNuo.
- CA Guerrero, J. S. (2013). Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. *Información tecnológica, - SciELO Chile*.
- Catallops, L. J. (2019). Sistema de gestión de información tecnológica para la Facultad 1 de la Universidad de las Ciencias Informáticas. https://repositorio.uci.cu/jspui/bitstream/123456789/10067/1/TD_09436_19.pdf.
- Comunicaciones, M. d. (julio de 2017). <https://www.mincom.gob.cu/es/ode/2469>. Obtenido de https://siteal.iiep.unesco.org/sites/default/files/sit_accion_files/11171.pdf
- Drupal. (2019). Obtenido de <https://www.drupal.org/about>
- Eguiluz, J. (2011). *Introducción a JavaScript*. España.
- Fresno Chávez, C. (2016). *Revista Cubana de Informática Médica*, <http://www.revinformatica.sld.cu/index.php/rcim/index>.
- Frisancho Jibaja, J. (2020). Diseño e implementación de un sistema de información de mercados para los productores rurales de granos andinos. <http://hdl.handle.net/20.500.12404/16699>.
- García O'reilly, A. (2017). Componente para el cálculo de la relevancia de información en el buscador Orión. <https://repositorio.uci.cu/jspui/handle/123456789/7954>.

- Gimenez, M. (9 de septiembre de 2021). Obtenido de <https://www.hiberus.com/crecemos-contigo/que-es-drupal/>
- González, D. d. (2018). API RESTasociadaal repositorio de Nova. https://repositorio.uci.cu/jspui/bitstream/123456789/9923/1/TD_09236_18.pdf.
- Gutiérrez Prado, F. (2023). <https://repositorio.lamolina.edu.pe>.
- Hardt, P. D. (2019). *Technische Universität Chemnitz, Chemnitz*, <https://nbn-resolving.org/urn:nbn:de:bsz:ch1-qucosa2-329499>.
- Hernandez, R. y., Fernandez, C., & Baptista, P. (2010). *Metodología de la Investigación*. Mexico: McGraw-Hill Interamericana.
- Integración y acceso abierto. (2022). *OpenAPI Initiative*, <https://www.openapis.org/>.
- Jenna. (2022). Obtenido de iteraciones<https://techinfo.wiki/planificacion-de-iteraciones/>
- JnjSite. (3 de FEBRERO de 2022). *JnjSite*. Obtenido de JnjSite: <https://jnjsite.com/los-patrones-del-diseno-software-grasp/>
- JnjSite. (3 de febrero de 2022). *JnjSite*. Obtenido de JnjSite: <https://jnjsite.com/los-patrones-del-diseno-software-grasp/>
- Luján-Mora, S. (2002). Programación de aplicaciones web: historia, principios básicos y clientes web. <http://hdl.handle.net/10045/16995>.
- Luna, F. (2019). *JavaScript - Aprende a programar en el lenguaje de la web*. RedUsers: https://books.google.es/books?id=SqikDwAAQBAJ&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false.
- Mihai, I.-C. (2023). *Proceedings of the International Conference on Cybersecurity and Cybercrime*. Asociatia Romana pentru Asigurarea Securitatii Informatiei.
- Montalvo González, N. (2019). *tesis*, <https://tesis.ipn.mx/bitstream/handle/123456789/31344/Nancy%20Montalvo%20Gonz%c3%a1lez%20-%20Tesis%20-%20P.pdf?sequence=2&isAllowed=y>.
- Pacino, M. C. (2018). La ciencia geomática presente y futuro. <http://hdl.handle.net/11336/101397>.
- Portilla, D. d. (2019). Licenciador por tiempo para el sistema Gestor de Recursos de Hardware y Software. https://repositorio.uci.cu/bitstream/123456789/10259/1/TD_09485_19.pdf
- Pressman. (2010).

Pressman, R. S. (2010). *Ingeniería de Software. Un enfoque Práctico. Séptima edición*. MCGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V.

Quiñonez, Y. A. (2022). <https://deliverypdf.ssrn.com/delivery.php?ID=199020086001028016116010114126122099104015006077091033071072020106070095067001067103097114000125006036111069121076070124064123062015046052031094101123020015069119110095033053093091114069087071005102083108076006>.

Rouse. (2019).

The PHP Group. (2019). Recuperado el 16 de Enero de 2019, de <http://us3.php.net/manual/es/preface.php>

Velásquez, S. M. (2019). Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software. <https://doi.org/10.33131/24222208.334>.

Visual Paradigm. (2019). *Visual Paradigm*. Recuperado el 25 de octubre de 2016, de Visual Paradigm: <https://www.visual-paradigm.com/>

visualstudio. (s.f.). Obtenido de <https://code.visualstudio.com/docs>

W3C. (2019). *World Wide Web Consortium*. Recuperado el 16 de Enero de 2019, de <http://www.w3.org/html>

W3C. (2019). *World Wide Web Consortium*. Recuperado el 16 de Enero de 2019, de <http://www.w3.org/TR/CSS/>

Yagüe, A., & Garbajosa, J. (2009). Comparativa práctica de las pruebas en entornos tradicionales y ágiles. *Revista Española de Innovación*, <https://www.redalyc.org/pdf/922/92217159004.pdf>.

Anexos

Anexo 1

Estructura de la entrevista con el cliente.

A continuación, se enumeran las preguntas realizadas durante la entrevista con el cliente.

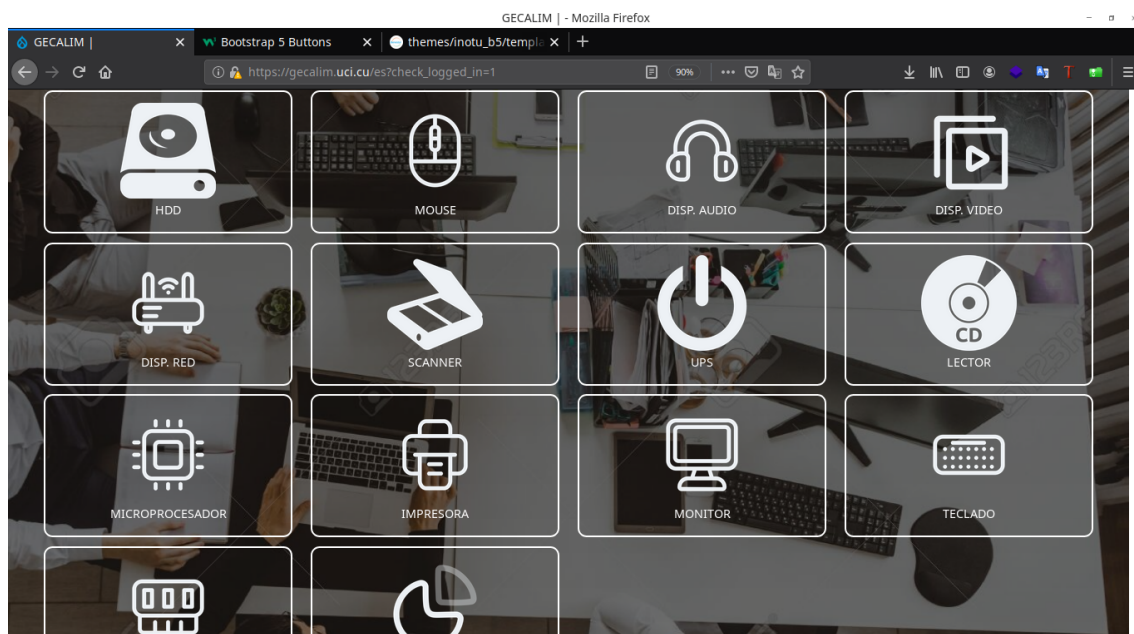
1. ¿Cuáles son las principales necesidades de la institución que la lleva a crear este sitio web?
2. ¿Qué tipo y formato de contenido debe permitir compartir el sitio?
3. ¿El usuario debe autenticarse en el sitio web para acceder a la información?
4. ¿Quién o quiénes serán el webmaster del sitio web?
5. ¿Puede el usuario consultar la información de un determinado autor que haya creado una computadora?
6. ¿Debe el usuario poder crear, eliminar, editar las publicaciones?
7. ¿Cuáles servicios de interacción debe permitir el sitio web?
8. ¿Desea agregar alguna otra funcionalidad al sitio web?

Anexo 2

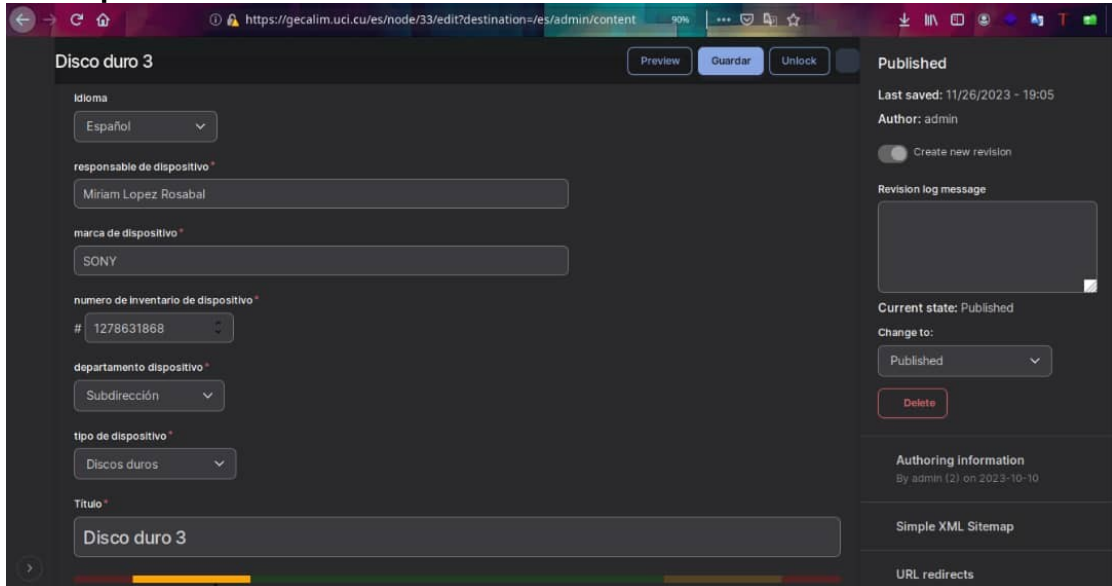
Página principal del sitio Web.



Medios informáticos que se gestionan en el Sistema.



Historia de usuario HU-4. Fuente: elaboración propia.

Historia de Usuario			
Código:	HU-4	Nombre:	Gestionar información referente a los discos duros.
Programador:	Raúl Noda Baró	Iteración Asignada:	1
Prioridad:	Alto	Tiempo estimado:	8h
Riesgo en desarrollo:	Alto	Tiempo real:	8h
Descripción: El sistema debe permitir a los usuarios mencionados realizar las siguientes funcionalidades respecto a los discos duros. Campos: • Insertar información de los discos duros • Modificar información de los discos duros • Eliminar información de los discos duros • Ver detalles de los discos duros			
Prototipo de interfaz: 			
Referencias:		Requisitos Funcionales	RF17, RF18, RF19, RF20

Historia de usuario HU-5. Fuente: elaboración propia.

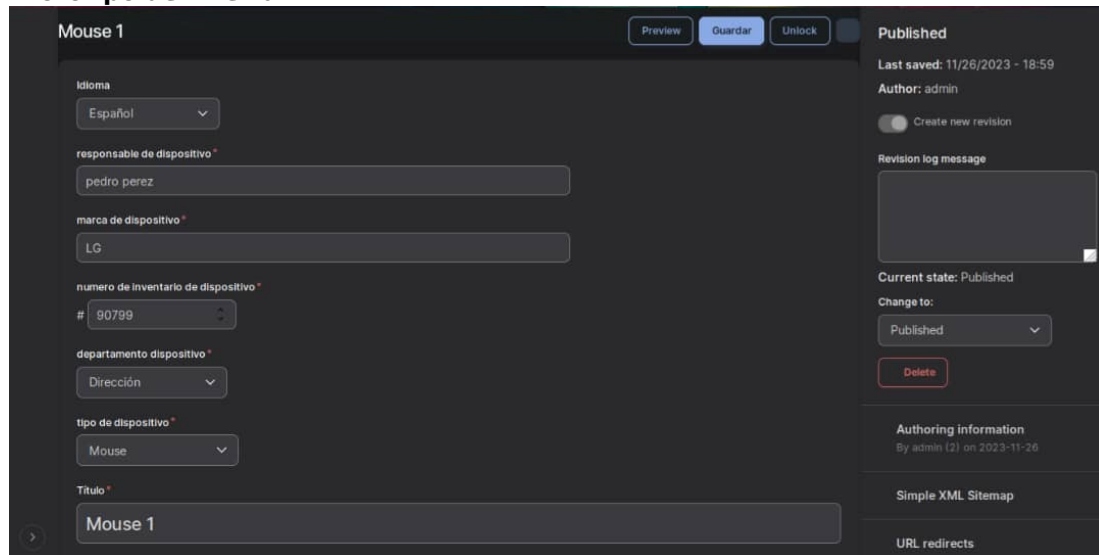
Historia de Usuario			
Código:	HU-5	Nombre:	Gestionar información referente a los mouse.
Programador:	Raúl Noda Baró	Iteración Asignada:	2
Prioridad:	Media	Tiempo estimado:	6h
Riesgo en desarrollo:	Medio	Tiempo real:	6h

Descripción: El sistema debe permitir a los usuarios mencionados realizar las siguientes funcionalidades respecto a los mouses.

Campos:

- Insertar información de los mouses
- Modificar información de los mouses
- Eliminar información de los mouses
- Ver detalles de los mouses

Prototipo de interfaz:



The screenshot shows a web form titled 'Mouse 1'. The form has several fields: 'Idioma' (dropdown menu), 'responsable de dispositivo' (text input), 'marca de dispositivo' (text input), 'numero de inventario de dispositivo' (text input), 'departamento dispositivo' (dropdown menu), 'tipo de dispositivo' (dropdown menu), and 'Título' (text input). The form is currently in 'Published' state. The sidebar on the right shows 'Last saved: 11/26/2023 - 18:59', 'Author: admin', 'Create new revision' button, 'Revision log message' (text area), 'Current state: Published', 'Change to:' (dropdown menu), 'Delete' button, 'Authoring information' (By admin (2) on 2023-11-26), 'Simple XML Sitemap', and 'URL redirects'.

Referencias:

Requisitos Funcionales

RF21, RF22, RF23, RF24

Historia de usuario HU-6. Fuente: elaboración propia.

Historia de Usuario			
Código:	HU-6	Nombre:	Gestionar información referente a los monitores.
Programador:	Raúl Noda Baró	Iteración Asignada:	2
Prioridad:	Media	Tiempo estimado:	6h
Riesgo en desarrollo:	Medio	Tiempo real:	6h
Descripción: El sistema debe permitir a los usuarios mencionados realizar las siguientes funcionalidades respecto a los monitores. Campos: • Insertar información de los monitores • Modificar información de los monitores • Eliminar información de los monitores • Ver detalles de los monitores			
Prototipo de interfaz:			

Monitor 7		Published
idioma	Español	Last saved: 10/22/2023 - 21:13
responsable de dispositivo *	Daniel Lazo Tamayo	Author: admin
marca de dispositivo *	HP	☐ Create new revision
numero de inventario de dispositivo *	8975463	Revision log message
departamento dispositivo *	Almacén	
tipo de dispositivo *	Monitores	Current state: Published
Título *	Monitor 7	Change to:
		Published
		Delete
		Authoring information
		By admin (2) on 2023-10-22
		Simple XML Sitemap
		URL redirects

Referencias:	Requisitos Funcionales	RF25, RF26, RF27, RF28
--------------	------------------------	------------------------

Historia de usuario HU-8. Fuente: elaboración propia.

Historia de Usuario			
Código:	HU-8	Nombre:	Gestionar información referente a los dispositivos de video.
Programador:	Raúl Noda Baró	Iteración Asignada:	2
Prioridad:	Media	Tiempo estimado:	6h
Riesgo en desarrollo:	Medio	Tiempo real:	6h
Descripción: El sistema debe permitir a los usuarios mencionados realizar las siguientes funcionalidades respecto a los dispositivos de video.			
Campos: - Insertar información de los dispositivos de video - Modificar información de los dispositivos de video - Eliminar información de los dispositivos de video - Ver detalles de los dispositivos de video			
Prototipo de interfaz:			

Dispositivo de video 1		Published
Preview Guardar Unlock		Last saved: 11/26/2023 - 19:03 Author: admin Create new revision
Idioma ▼ Español		Revision log message
responsable de dispositivo * Cristiano Ronaldo		Current state: Published
marca de dispositivo * HUAWEI		Change to: Published
numero de inventario de dispositivo * # 203529		Delete
departamento dispositivo * Recursos Humanos		Authoring information By admin (2) on 2023-10-10
tipo de dispositivo * Dispositivos de video		Simple XML Sitemap
Título * Dispositivo de video 1		URL redirects

Referencias:

Requisitos
Funcionales

RF33, RF34, RF35, RF36

Historia de usuario HU-9. Fuente: elaboración propia.

Historia de Usuario			
Código:	HU-9	Nombre:	Gestionar información referente a los dispositivos de red.
Programador:	Raúl Noda Baró	Iteración Asignada:	2
Prioridad:	Media	Tiempo estimado:	6h
Riesgo en desarrollo:	Medio	Tiempo real:	6h
Descripción: El sistema debe permitir a los usuarios mencionados realizar las siguientes funcionalidades respecto a los dispositivos de red.			
Campos: - Insertar información de los dispositivos de red - Modificar información de los dispositivos de red - Eliminar información de los dispositivos de red - Ver detalles de los dispositivos de red			
Prototipo de interfaz:			

Dispositivo de red 1

Preview

Guardar

Unlock

Published

Idioma

Español

responsable de dispositivo *

Luis Gustavo Flores

marca de dispositivo *

ACER

numero de inventario de dispositivo *

1243434

departamento dispositivo *

Caja

tipo de dispositivo *

Dispositivos de red

Título *

Dispositivo de red 1

Last saved: 11/26/2023 - 19:04

Author: admin

Create new revision

Revision log message

Current state: Published

Change to:

Published

Delete

Authoring information

By admin (2) on 2023-10-10

Simple XML Sitemap

URL redirects

Historia de usuario HU-10. Fuente: elaboración propia.

Historia de Usuario			
Código:	HU-10	Nombre:	Gestionar información referente a los lectores de discos.
Programador:	Raúl Noda Baró	Iteración Asignada:	3
Prioridad:	Baja	Tiempo estimado:	4h
Riesgo en desarrollo:	Bajo	Tiempo real:	4h
Descripción: El sistema debe permitir a los usuarios mencionados realizar las siguientes funcionalidades respecto a los lectores de discos.			
Campos: - Insertar información de los lectores de discos - Modificar información de los lectores de discos - Eliminar información de los lectores de discos - Ver detalles de los lectores de discos			
Prototipo de interfaz:			

Lector de disco 3		Published
Preview Guardar Unlock		Last saved: 11/26/2023 - 19:14 Author: admin ☐ Create new revision Revision log message Current state: Published Change to: Published Delete Authoring information By admin (2) on 2023-11-26 Simple XML Sitemap URL redirects
Idioma Español		
responsable de dispositivo * Cristiano Ronaldo		
marca de dispositivo * BENQ		
numero de inventario de dispositivo * # 23423		
departamento dispositivo * Caja		
tipo de dispositivo * Dispositivos de audio		
Título * Lector de disco 3		
Referencias:	Requisitos Funcionales	RF41, RF42, RF43, RF44