



Universidad de las Ciencias Informáticas

Facultad 1

**Sistema de gestión de información tecnológica para la Facultad 1 de la
Universidad de las Ciencias Informáticas**



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

Autor:

Leonardo José Santiesteban Catallops

Tutores:

MSc. Leiny Amel Pons Flores

Ing. Leovan Peña Serrano

Ing. Gustavo Rodríguez Méndez

"Estoy convencido que lo que separa a los emprendedores exitosos y los que no es la perseverancia. Es tan difícil, pones tanto de tu vida en esto, hay momentos tan duros en que la mayoría se da por vencido, no los culpo, es muy difícil y consume gran parte de tu vida. Al menos que tengas mucha pasión en lo que haces no vas a sobrevivir, vas a darte por vencido."

Steve Jobs



Declaración de autoría

Declaro por este medio que yo Leonardo José Santiesteban Cantallops, con carnet de identidad 95101842762, soy el autor principal del trabajo titulado "Sistema de Gestión de Tecnologías para la Facultad 1" y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Declaro que todo lo anteriormente expuesto se ajusta a la verdad, y asumo la responsabilidad moral y jurídica que se derive de este juramento profesional.

Y para que así conste, firmo la presente declaración de autoría en La Habana a los ____ días del mes de _____ del año 2019.

Autor:

Leonardo José Santiesteban Cantallops

Tutores:

MSc. Leiny Amel Pons Flores

Ing. Leovan Peña Serrano

Ing. Gustavo Rodríguez Méndez

Resumen

La presente investigación tiene como objetivo desarrollar un sistema web de gestión de la información de los medios tecnológicos de la Facultad 1 de la Universidad de las Ciencias Informáticas, para contribuir a informatizar el proceso de organización y procesamiento de la información de estos medios. Este proceso se realiza de forma manual, generando dificultades a las cuales se enfrentan los asesores de tecnologías de cada centro. El análisis del estado del arte permitió identificar las funcionalidades a desarrollar y la tecnología a utilizar en la propuesta de solución. La metodología AUP-UCI guió el desarrollo de la investigación, utilizando su variante 4. Entre las principales tecnologías a utilizar están el marco de trabajo *Symfony*, *Hypertext Preprocessor* (PHP) como lenguaje de programación y el entorno integrado de desarrollo *PhpStorm*. Se aplicó una estrategia de prueba con el objetivo de verificar el cumplimiento de los objetivos trazados y evaluar la calidad del sistema. Las pruebas de software aplicadas al sistema de gestión de tecnologías, demostraron que es una solución funcional, segura, con un rendimiento adecuado. Los resultados de la investigación evidenciaron que el sistema desarrollado posee un alto valor para facilitar el trabajo de la gestión de información de los medios tecnológicos.

Palabras claves: gestión, sistema, tecnologías, información.

ÍNDICE

Introducción	1
Capítulo 1: Fundamentación teórica sobre la gestión tecnológica	6
1.1 Conceptos asociados al dominio del problema	6
1.2 Sistemas homólogos	9
1.2.1 Proyecto TAU.....	10
1.2.2 OCS Inventory NG	10
1.2.3 Sistema Integral de Gestión Cedrux:.....	11
1.2.4 Sistema Gestor de Recursos de <i>Hardware</i> y <i>Software</i>	11
1.3 Herramientas, lenguajes y metodología de desarrollo	12
1.3.1 Lenguaje y herramientas de modelado	13
1.3.2 <i>Framework</i> de desarrollo.....	13
1.3.3 Lenguaje de programación.....	15
1.3.4 Entorno de desarrollo integrado	16
1.3.5 Servidor web	17
1.3.6 Sistemas gestores de base de datos	17
1.3.7 Metodología de desarrollo.....	19
1.3.8 Herramientas para las pruebas de seguridad.....	19
Conclusiones parciales.....	20
Capítulo 2: Análisis, planificación y diseño del sistema de gestión de tecnologías	21
2.1 Descripción de la propuesta de solución	21
2.2 Requisitos de la propuesta de solución	22
2.2.1 Requisitos funcionales	23

2.2.2 Requisitos no funcionales	25
2.3 Historia de usuario.....	27
2.4 Arquitectura de software.....	30
2.5 Patrones de diseño.....	31
2.5.1 Patrones Generales de Software para la Asignación de Responsabilidades (GRASP)	32
2.5.2 Patrones Gang of Four (GOF).....	33
2.6 Diagrama de clase del diseño	33
2.7 Modelo de datos.....	34
2.8 Modelo de despliegue	36
Conclusiones parciales.....	37
Capítulo 3: Implementación y validación del sistema de gestión de tecnologías.....	38
3.1 Diagrama de Componentes.....	38
3.2 Estándares de codificación.....	40
3.3 Validación de la propuesta de solución.....	41
3.3.1 Criterio de expertos	42
3.3.3 Pruebas funcionales.....	44
3.3.3 Pruebas de usabilidad.....	46
3.3.4 Pruebas de Seguridad	48
Conclusiones parciales	49
Conclusiones	50
Recomendaciones	51
Referencias bibliográficas	52
Anexos.....	57

ÍNDICE DE TABLAS

Tabla 1. Análisis de sistemas homólogos.	12
Tabla 2. Requisitos funcionales.	23
Tabla 3. HU_9 Listar Medios.....	28
Tabla 4. HU_20 Generar reporte estadístico de la propiedad y cantidad de medios por local.....	28
Tabla 5. HU_21 Generar reporte estadístico de la cantidad y el tipo de medios rotos por local.	29
Tabla 6. HU_22 Generar reporte estadístico del total de medios existentes en el área.....	29
Tabla 7. Descripción del Diagrama de Componentes.	38
Tabla 8: Estándares de codificación.....	40
Tabla 9. Selección de los expertos.....	42
Tabla 10. Resultado de las encuestas aplicada al grupo de experto.	43
Tabla 11. Caso de prueba Listar medio.	44
Tabla 12. Indicadores de la categoría. Visibilidad del sistema.....	46
Tabla 13. Indicadores de la categoría. Lenguaje común entre sistemas y usuario.	47
Tabla 14: Resultado de las pruebas de seguridad	48
Tabla 15. HU_8 Adicionar Medio.	57
Tabla 16. HU_10 Modificar Medio.	57
Tabla 17. HU_11 Eliminar Medio.	58
Tabla 18. HU_12 Adicionar Local.....	58
Tabla 19. HU_13 Listar Locales.	58
Tabla 20. HU_14 Modificar Local.	59
Tabla 21. HU_15 Eliminar Local.....	59

Tabla 22. HU_16 Adicionar Área.....	60
Tabla 23. HU_17 Listar Áreas.....	60
Tabla 24. HU_18 Modificar Área.....	60
Tabla 25. HU_19 Eliminar Área.....	61
Tabla 26. HU_1 Adicionar usuario.....	61
Tabla 27. HU_2 Listar usuarios.....	62
Tabla 28. HU_3 Modificar usuario.....	62
Tabla 29. HU_4 Eliminar usuario.....	63
Tabla 30. HU_5 Asignar roles.....	63
Tabla 31. HU_6 Asignar permisos.....	64
Tabla 32. HU_7 Autenticar usuario.....	64

ÍNDICE DE FIGURAS

Figura 1: Imagen de propuesta de solución del sistema.....	22
Figura 2. Arquitectura MVC.....	30
Figura 3: Diagrama de clase de diseño “Gestionar Medio”.....	34
Figura 4. Modelo de datos.....	36
Figura 5: Modelo de despliegue.....	36
Figura 6. Diagrama de Componentes.....	39
Figura 7. Gráfico de no conformidades.....	45
Figura 8: Diagrama de clase de diseño “Gestionar Área”.....	65
Figura 9: Diagrama de clase de diseño “Gestionar Local”.....	66

INTRODUCCIÓN

La gestión tecnológica representa una serie de acciones, herramientas y técnicas para integrar la ciencia, la ingeniería, los negocios, los procesos organizacionales y el personal asociado para proyectar las fortalezas y corregir las debilidades como medio de incrementar la competitividad de las organizaciones. La internacionalización, la globalización económica y los desarrollos de las nuevas tecnologías de información y comunicaciones, han ocasionado un aumento en el nivel de competitividad de las organizaciones a nivel mundial, influenciado principalmente por las nuevas tecnologías, la calidad en los productos y servicios, y la satisfacción del cliente como fin último de todas las organizaciones. En este contexto, la gestión tecnológica es una de las disciplinas que está generando un interés creciente en el campo de la dirección estratégica. Se puede definir como un conjunto sistemático de procesos orientados a la planificación, organización y ejecución de actividades relacionadas con la evaluación, adquisición y puesta en marcha de tecnologías claves para el cumplimiento de los objetivos estratégicos de una organización; con el objetivo de generar productos o servicios competitivos a partir del aprovechamiento de su capacidad tecnológica (FUENTES, y otros, 2011).

La gestión tecnológica surge y se desarrolla en el seno de las empresas y su objetivo fundamental es el logro de una mejor vinculación investigación-industria-sociedad, que debe entenderse como una relación de mercado. Esto implica comprender que este se rige fundamentalmente por leyes de oferta y demanda. La gestión tecnológica busca integrar el proceso de cambio tecnológico con los aspectos estratégicos y operativos del control y la toma de decisiones de la empresa. Es el instrumento que vincula el sector productivo y de la investigación-desarrollo en el proceso de innovación tecnológica. Requiere de una preparación conceptual y ejecutiva y se realiza para apoyar los procesos de innovación tecnológica que permiten identificar las necesidades y oportunidades tecnológicas e implica una capacidad de manejo del cambio técnico. Por otra parte, garantiza las actividades de investigación y la transferencia de sus resultados a las entidades productivas. Sin embargo no es un campo del saber meramente especulativo sobre la tecnología y su desarrollo; es también una práctica soportada en un conocimiento derivado del análisis y la interpretación de las observaciones del comportamiento del desarrollo tecnológico, como

proceso social, y resultado de las observaciones de este proceso en organizaciones y países y de su relación con el proceso de desarrollo global de las sociedades modernas (Ávila, y otros, 2007).

A raíz del proceso de informatización de la sociedad cubana, la gestión de la información tecnológica es un campo que ha cobrado importancia al permitir la explotación y control de la tecnología disponible para llevar a cabo este proceso. La informatización es en la actualidad uno de los principales objetivos a alcanzar por nuestro país, el cual ha realizado numerosos esfuerzos y concretado ideas como la Universidad de las Ciencias Informáticas (UCI).

La UCI tiene como objetivos fundamentales llevar a cabo la informatización del país y desarrollar la industria del software para contribuir al desarrollo económico y científico del mismo. Está constituida por seis facultades, las cuales comprenden a su vez un conjunto de centros de desarrollo de software. Los objetivos de estos centros giran en torno a la misión de la Universidad y cada uno realiza su aporte en la informatización del país especializándose en diferentes áreas. La Facultad 1 tiene su papel protagónico respecto al tema y cuenta con varios centros de desarrollo. Entre estos centros, se destaca el Centro de Ideoinformática (CIDI), entidad que tiene como misión proporcionar servicios y productos de información de alto valor agregado que contribuyan a la informatización dentro y fuera de la Universidad. Debido a la alta disponibilidad de tecnología que la Universidad pone en manos de estudiantes, profesores y especialistas se alcanza actualmente un alto nivel de desarrollo de sistemas informáticos para diferentes organismos tanto nacionales como internacionales.

Dentro de los procesos que realiza la Facultad se encuentra el control y seguimiento de todos los útiles, herramientas y tecnologías informáticas que están a disposición de la docencia y la producción de software. Teniendo en cuenta que las estaciones de trabajo en explotación y sus accesorios no poseen las mismas características, es muy importante controlar su estado técnico y ubicación en todo momento para evitar pérdidas o actos delictivos que no sean detectados en el momento oportuno. Con base en la entrevista (Anexos 3) realizada al asesor de tecnologías de CIDI, se determinó que, en esta área, la gestión de información del *hardware* de los equipos tecnológicos se realiza de forma manual.

La labor del control de los medios tecnológicos se realiza a través de documentos en formato duro u hojas de cálculo electrónicas. Esto conlleva a que en ocasiones se cometan errores al registrar o controlar un equipo o tecnología de forma manual, hace que la labor de control, además de lenta, sea extensa y

agotadora. Es posible también que la información se duplique, se pierda o sea fácilmente modificable al existir un alto riesgo de acceso, esto posibilita la ejecución de hechos delictivos, pérdidas económicas y de manera general se afecta la actividad productiva del área. Además, no es posible generar de forma rápida reportes estadísticos sobre el estado actual de la tecnología en el centro y en general en la facultad, que apoyen a los directivos en la toma de decisiones.

Por lo anteriormente expuesto surge el siguiente **problema de investigación** ¿Cómo contribuir a la gestión de los medios tecnológicos de la Facultad 1?

El **objeto de estudio** se centra en los sistemas de gestión de medios tecnológicos, comprendido en el **campo de acción** gestión de medios tecnológicos en la Facultad 1 de la UCI.

Como **objetivo general** se plantea desarrollar un sistema para la gestión de los medios tecnológicos de la Facultad 1.

Se define como **idea a defender**: con el desarrollo de un sistema se contribuirá la gestión de los medios tecnológicos en la Facultad 1 de la Universidad de las Ciencias Informáticas.

Para dar cumplimiento al objetivo general se trazaron los siguientes **objetivos específicos**:

1. Realizar el marco teórico referencial de la presente investigación el cual responde a la gestión de los medios tecnológicos.
2. Analizar las herramientas y tecnologías indispensables para la gestión de los medios tecnológicos a través de sistemas web.
3. Realizar el levantamiento de los requisitos funcionales y no funcionales del sistema a desarrollar.
4. Realizar el análisis y diseño del Sistema de gestión de información tecnológica para la Facultad 1 de la Universidad de las Ciencias Informáticas.
5. Implementar el sistema web que permita la gestión de los medios tecnológicos de la Facultad 1 de la UCI.
6. Validar el sistema implementado a partir de los métodos definidos.

Para el desarrollo de la investigación se emplean los siguientes **métodos de investigación**:

Métodos teóricos:

- **Análítico-Sintético:** este método se utiliza para arribar a conclusiones prácticas y teóricas de la investigación que se desarrolla. Mediante este se realiza un análisis de los documentos, las publicaciones, bibliografías y en general toda la información relacionada con sistemas de gestión de información de medios tecnológicos y sistemas web que se utilizan para tal fin.
- **Modelación:** se utiliza para realizar el modelado del sistema web a desarrollar, a través de los artefactos correspondientes a las fases: Modelo del Negocio, Requisitos, Análisis y Diseño e Implementación.

Métodos Empíricos:

- **Entrevista:** se utiliza con el objetivo de conocer la estructura organizativa del proceso de gestión de información del inventario de *hardware* en el área de tecnología de la Facultad 1 de la UCI.
- **Observación:** se utiliza para obtener una información más precisa de cómo se realiza proceso de gestión de información de los expedientes técnicos en el área de tecnología.
- **Encuesta:** Se emplea para la recopilación de información sobre el proceso de gestión de la información de los medios tecnológicos en la Facultad 1 de la UCI.

La estructura del documento está conformada por los siguientes capítulos:

Capítulo 1:

Fundamentación teórica sobre la gestión tecnológica: Se realiza un estudio del estado del arte que permite conocer la situación mundial del tema. Además, se hace referencia a los principales conceptos referentes al mismo. Se describen los sistemas informáticos que existen para evaluar la satisfacción de los clientes. Por último, se caracterizan las metodologías de desarrollo, tecnologías y herramientas posibles a utilizar, seleccionando las que más se ajustan a las necesidades del sistema a implementar, con el fin de desarrollar una aplicación web que permita la gestión de información de los medios tecnológicos de la Facultad 1.

Capítulo 2:

Análisis, planificación y diseño del sistema de gestión de tecnologías: En este capítulo se hace un análisis y descripción general de la propuesta de desarrollo del sistema, así como la realización del levantamiento de los requisitos funcionales y no funcionales, entrada de la implementación del software. También se especifican los patrones del diseño aplicados, así como los artefactos derivados de la metodología de desarrollo de software seleccionada para esta etapa del proceso de desarrollo del sistema propuesto. Se lleva a cabo una explicación de la organización del sistema expuesta en los diagramas de clase para el correcto funcionamiento de este.

Capítulo 3:

Implementación y validación del sistema de gestión de tecnología: En este capítulo se detalla la propuesta de solución al problema planteado. Se describe la organización del sistema en un diagrama de componentes y se especifican los estándares de codificación a utilizar. Se realizan las estrategias de pruebas definidas para el sistema y se muestran interfaces como parte del resultado final.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA SOBRE LA GESTIÓN TECNOLÓGICA

El Sistema de Gestión de Tecnologías (SGT) que se propone desarrollar brinda una solución efectiva para facilitar el trabajo con la información de los medios tecnológicos. Con el objetivo de lograr una mayor comprensión del alcance de la investigación, en el presente capítulo se exponen los fundamentos teóricos asociados al dominio del problema planteado. Se realiza un estudio y análisis de los principales sistemas para la gestión de información identificados, se analizan sus características y procesos, ventajas y desventaja. Se realiza un estudio de las definiciones y términos fundamentales que se usarán a lo largo de la investigación, se explica en detalle las herramientas, tecnologías, metodologías y se fundamenta el uso de los lenguajes de programación seleccionados para el desarrollo de la aplicación web y de esta forma lograr el producto que más se ajusten a las necesidades del cliente.

1.1 Conceptos asociados al dominio del problema

Entre los principales conceptos utilizados en la investigación están los siguientes:

- ❖ **Gestión Tecnológica**

La gestión tecnológica (GT) se define como el proceso de administrar el desarrollo de la tecnología, su implementación y difusión en prácticamente todos los sectores de nuestra sociedad en general. La GT se define como "[...] la utilización de conocimientos, procedimientos y experticias que permiten mejorar la utilización de las capacidades tecnológicas con el propósito de alcanzar mejores niveles de productividad y consecuentemente de competitividad" (Quezada-Torres, y otros, 2018). Todo ello enfocado a un escenario futuro con la finalidad de alcanzar niveles superiores de productividad y competitividad en condiciones de sostenibilidad. La GT responde a la necesidad de administrar el factor tecnológico con un sentido estratégico dentro de la organización para adaptarse a los cambios del entorno y aumentar su competitividad desde un enfoque multifuncional y multidisciplinario que integra los aspectos tecnológicos en los procesos de toma de decisiones. Los procesos de GT en una organización se fortalecen cuando se trabaja con alianzas entre los actores: universidad, industria y gobierno. Además, implica el manejo del proceso de innovación a través de la Investigación y Desarrollo (I+D), lo cual incluye la introducción y uso de tecnología en productos, en procesos industriales, y en otras áreas estructurales y funcionales de la

empresa, así como la utilización de este conocimiento en la solución de los diferentes problemas de la sociedad (Quezada-Torres, y otros, 2018).

La gestión tecnológica es un sistema de conocimientos y prácticas relacionadas con los procesos de creación, desarrollo, transferencia y uso de la tecnología. Algunos conciben este sistema como una colección de métodos sistemáticos para gestionar los procesos de aplicación de los conocimientos, extender el rango de actividades humanas y producir bienes y servicios. Integrado por los conocimientos de: ingeniería, ciencias y disciplinas del área de gestión, para planear, desarrollar e implementar capacidades tecnológicas en el diseño y el logro de los objetivos estratégicos y operacionales de una organización (Soa, y otros, 2007).

La gestión tecnológica, que tiene como objetivo manejar (gerenciar) la variable tecnológica en la estrategia global de la empresa, es el conjunto de actividades de gestión referentes a la identificación y obtención de tecnología, la investigación, el desarrollo y la adaptación de las nuevas tecnologías a la empresa y su explotación en la producción; se ocupa también de la vigilancia y prospectiva tecnológica, con el fin de detectar tecnología de interés en el futuro, la reingeniería, de la ingeniería inversa o análisis de los productos de los competidores, de los derechos de propiedad y licencia, de normas y estándares, de las alianzas estratégicas y de las formas de cooperación. En general, la gestión tecnológica incluye la tecnología de productos y procesos, como la utilizada en funciones de dirección u organización. La gestión tecnológica en la empresa consiste, básicamente, en el conjunto de decisiones vinculadas a la creación y/o adquisición, desarrollo y/o transformación y comercialización de la tecnología, desde un punto de vista estratégico como operacional (Alzate, 2005).

En Cuba la gestión de la información tecnológica es un campo que ha cobrado importancia al permitir la explotación y control de la tecnología disponible para llevar a cabo este proceso. Gran parte de las empresas e instituciones del país se plantean actualmente la tarea de realizar la gestión de su tecnología como parte de la necesidad del control de los medios básicos (Benitez-Odio, y otros, 2017). En la UCI se lleva a cabo la gestión tecnológica utilizando la herramienta Sistema Gestor de Recursos de Hardware y Software (GRHS), la cual carece de determinadas funcionalidades que son necesarias en áreas específicas de la universidad. Esta herramienta no permite editar la información de los medios tecnológicos como las *Central Processor Unit* (CPU), monitores, teclados, mouse, etc. Esta labor se

realiza actualmente de forma manual por los asesores de tecnología de cada área utilizando un documento Excel.

Para el autor de la presente investigación, la gestión tecnológica constituye un sistema de conocimientos y prácticas relacionados con los procesos de creación, desarrollo, transferencia y uso de la tecnología.

➤ Gestión de Información

La gestión de información está estrechamente relacionada con la gestión de datos, sistemas, tecnología, procesos y estrategia. En la actualidad transita por dos variantes; como elemento que contribuye a la gestión del conocimiento, modalidad esencial de aquellas organizaciones cuyos resultados constituyen aportes científicos; y otra no menos importante cuyas prácticas perfeccionan el accionar de los procesos y la toma de decisiones de las organizaciones en sí mismas (Capote, 2011).

Según Ledo y Pérez "...el proceso de organizar, evaluar, presentar, comparar los datos en un determinado contexto, controlando su calidad, de manera que esta sea veraz, oportuna, significativa, exacta y útil y que esta información esté disponible en el momento que se le necesite. Ella se encamina al manejo de la información, documentos, metodologías, informes, publicaciones, soportes y flujos en función de los objetivos estratégicos de una organización" (Ledo, y otros, 2012).

A partir de los conceptos mencionados y en correspondencia con los objetivos de la investigación, se define el concepto de gestión de información como el proceso de organizar, almacenar, eliminar y divulgar la información existente en un contexto determinado, garantizando su integridad, confidencialidad y disponibilidad en el momento que sea necesario.

➤ Sistema de Gestión de Información

Actualmente una de las soluciones que ha permitido gestionar la información de una manera eficiente y organizada son los Sistemas de Gestión de Información (SGI). Un sistema de gestión es un conjunto de sistemas y procedimientos que recopilan información de una variedad de fuentes, la compilan y la presentan en un formato legible. Los mismos recogen y almacenan datos, procesan los datos para generar informes significativos y relevantes, y respaldan las operaciones mediante la aplicación de procesos definidos y la creación de registros de auditoría para obtener reportes de la información organizados para un mejor entendimiento. El propósito principal de un sistema de gestión de la

información es hacer que la toma de decisiones por parte de los directivos sea más eficiente y productiva. Mediante la combinación de la información de una variedad de fuentes en una sola base de datos y la presentación de la información en un formato lógico, un SGI puede proporcionar a los administradores con todo lo que necesitan para tomar decisiones altamente informadas y realizar un análisis en profundidad de las cuestiones operativas (Ingram, 2019).

La característica principal de un sistema de gestión de información es su capacidad para almacenar datos y facilitar información para ser recuperada por los usuarios del sistema. El tipo de base de datos utilizada determina cómo el sistema de gestión de información responde a las peticiones o consultas de información. Los sistemas que utilizan una base de datos relacional almacenan los datos en tablas separadas en vez de una tabla grande. Una vez que se almacenaron los datos, los administradores de bases de datos conectan las tablas de datos pertinentes una a la otra mediante el uso de llaves. Estas llaves identifican la relación entre los datos en una tabla y los datos de la otra tabla. Las bases de datos relacionales proporcionan una respuesta más rápida a las consultas y almacenan más información que las bases de datos jerárquicas. Las bases de datos jerárquicas siempre acceden a los datos, en la parte superior de la base de datos y desplazan hacia abajo. No toman en cuenta la relación entre las diferentes piezas de información (Hamlett, 2019).

A modo de conclusión el autor del presente trabajo considera que un SGI automatizado es la colaboración entre el hombre y las máquinas que posibilita de una manera más fácil y organizada, manejar, recoger, procesar, almacenar y distribuir la información.

1.2 Sistemas homólogos

Situación actual y perspectivas de los Sistemas de Gestión de las Tecnologías en Cuba y el mundo:

En la actualidad se han desarrollado sistemas para la gestión de las tecnologías, los cuales poseen una amplia gama de funcionalidades que facilitan el trabajo en las empresas. A continuación, se analizan algunos sistemas de gestión, con el objetivo de identificar sistemas capaces de solucionar la problemática en su totalidad o que permita reutilizar algunas funcionalidades para brindar una solución al problema planteado anteriormente.

1.2.1 Proyecto TAU

Se empezó creando una interfaz web a una base de datos en Access para el control de inventario de equipos. Posteriormente se ha pasado a una interfaz completa vía web y cuya característica más importante, es que evoluciona en base a las necesidades que se van presentando (Huelva, 2006).

Algunas de las características de este sistema son:

- Su característica más importante, es que evoluciona en base a las necesidades que se van presentando.
- Es modular, basado en PHP + *Asynchronous JavaScript And XML* (AJAX) + *Open Source Database Software* (MySQL).
- Brinda información sobre el inventario de *hardware* y software instalado.
- Gestión de usuarios, aplicaciones, averías, almacén y control de fungibles, etc.
- Conexión Terminal Server (mediante applet java para administrar servidores).
- Utiliza software privativo para su desarrollo.
- Se puede utilizar en sistemas operativos como Linux o Windows.
- En la actualidad, no presenta la característica de gestión de *hardware* o información personalizada.

1.2.2 OCS Inventory NG

OCS Inventory es una aplicación diseñada para ayudar a los administradores de redes y sistema a realizar un seguimiento de los componentes de *hardware* y las configuraciones de software instaladas en las computadoras, además es capaz de detectar todos los dispositivos activos en su red, tales como *switch*, router, impresoras de red y dispositivos desatendidos. Además, es un software propietario por lo que hay que pagar costosas licencias para su utilización. Algunas de las características de OCS InventoryNGson:

- Brinda información sobre del inventario de *hardware* y software instalado.
- Potente sistema de despliegue que permite distribuir la instalación del software y los *scripts* en los equipos sin sobre cargar la red.
- Consola de administración Web.
- Soporte de múltiples sistemas operativos, incluyendo Microsoft Windows, Linux, * BSD, SunSolaris, IBM AIX, HP-UX, Mac OS X.

- Arquitectura de tres capas utilizando los estándares actuales, los protocolos HTTP/HTTPS y el formato de datos XML.
- Servicio Web accesible a través de la interfaz SOAP.
- Análisis de la red.
- Utiliza software privativo para su desarrollo.
- Se puede utilizar en sistemas operativos como Linux o Windows.
- En la actualidad, no presenta la característica de gestión de *hardware* o información personalizada.

1.2.3 Sistema Integral de Gestión CedruX:

Posibilita el trabajo con las existencias de productos, dándole entrada y salida del almacén, regulando el flujo de mercancía, con el fin de hacer más rentable su posesión y garantizar en cierto grado el éxito de la organización. Tiene como objetivo garantizar la disponibilidad continua de los productos; así como ejecutar los procesos de gestión de manera eficiente y dinámica, promoviendo el uso racional de recursos. Facilita el registro de los productos, sus existencias, la gestión y control de lotes; así como los movimientos de entrada y salida de los productos en el almacén. Este sistema, aunque cuenta con una aceptación positiva entre los usuarios, se encuentra en explotación solamente en la base de almacenes perteneciente al Ministerio de las Fuerzas Armadas Revolucionarias de Cuba.

Entre sus características más importantes se encuentran que utiliza software libre para su desarrollo, se puede utilizar en sistemas operativos como Linux o Windows y no presenta la característica de gestión de *hardware* o información personalizada.

1.2.4 Sistema Gestor de Recursos de *Hardware* y *Software*

Gestor de Recursos de *Hardware* y *Software*, es una herramienta informática que facilita el inventario del *hardware* y el software instalado en su red de computadora. Implementada por el centro de Telemática de la Facultad 2 de la Universidad de Ciencias Informáticas (2015). GRHS permite obtener de manera automática los datos del *hardware* y software de las máquinas, previamente instalado en cada una de ellas un cliente (con versión para *Windows* y *GNU/Linux*) los datos extraídos se almacenan en una base de datos que puede ser consultada desde un cliente web. Esto permite tener información actualizada del estado de cada máquina y conocer los detalles del *hardware* y el software de las mismas.

Esta herramienta además permite realizar consultas a la base de datos a partir del uso del API-Rest publicado en el servidor. Entre sus características más importantes se encuentran que utiliza software libre para su desarrollo, se puede utilizar en sistemas operativos como Linux o Windows y no presenta ha desarrollado la característica de gestión de *hardware* o información personalizada.

A continuación, en la Tabla 1 se presenta una tabla resumen donde se recogen los datos más significativos de cada uno de los sistemas analizados:

Tabla 1. Análisis de sistemas homólogos.

Herramienta	Software Libre	Multiplataforma	Gestión personalizada de <i>hardware</i>	Exportar información personalizada
Proyecto TAU	No	Sí	No	No
OCS <i>Inventory</i>	No	Sí	No	No
GRHS	Sí	Sí	No	No
Sistema Integral de Gestión Cedrux	Sí	Sí	No	No

El estudio de los sistemas anteriormente descritos, permitió afirmar que los mismos no pueden ser utilizados para satisfacer las necesidades actuales de los centros asociados a la Facultad 1 de la Universidad de las Ciencias Informáticas. Así mismo, evidenció la necesidad de desarrollar un nuevo sistema que permita la informatización de los procesos de gestión de información de las tecnologías de dicha facultad. No obstante, se toman como apoyo para el nuevo desarrollo, las buenas prácticas que quedaron como saldo del estudio realizado.

1.3 Herramientas, lenguajes y metodología de desarrollo

Después de un exhaustivo análisis de los elementos abordados en los epígrafes anteriores y de plantearse la necesidad de desarrollar una nueva propuesta de solución para la gestión de medios tecnológicos a continuación se procede a la elección de la metodología, lenguajes y herramientas de desarrollo para guiar el proceso.

1.3.1 Lenguaje y herramientas de modelado

Un lenguaje de modelado provee un vocabulario y conjunto de reglas centradas en la representación conceptual y física de un sistema. El Lenguaje Unificado de Modelado (*Unified Modeling Language*), es un lenguaje informático gráfico o textual que sigue un conjunto semántico de reglas y marcos para el diseño y construcción de estructuras y modelos. Se utiliza para visualizar, especificar, construir y documentar software orientados a objetos. Permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma fácil de comprender y comunicar a otras personas (TECHOPEDIA, 2017).

La elección de UML se sustenta en su facilidad para especificar, documentar, visualizar y comprender los artefactos generados en las diferentes fases del desarrollo de software, así como sus ventajas para la comunicación entre el cliente y el equipo de desarrollo. Las herramientas de modelado son el nombre con que se identifica la herramienta utilizada para apoyar las actividades del proceso de software. Facilitan la interacción entre los miembros de un equipo al hacer que la diagramación sea un proceso iterativo y dinámico. Los analistas de sistemas se apoyan en estas herramientas desde el principio hasta el fin del ciclo de vida, para incrementar la productividad, comunicarse de manera eficiente con los usuarios y desarrolladores, e integrar el trabajo que desempeñan en el sistema (Martínez, y otros, 2014).

Visual Paradigm

Visual Paradigm es la herramienta *Computer Aided Software Engineering* (CASE) escogida para trabajar en la investigación porque es multiplataforma y contribuye al desarrollo de sistemas de software fiables, mediante un enfoque orientado a objetos. Su preferencia está determinada por las oportunidades que ofrece para la construcción de aplicaciones de calidad, con mayor rapidez (necesidad fundamental de esta investigación) y menor costo. Soporta el ciclo completo de desarrollo de software y permite su documentación en diferentes formatos, empleando UML como lenguaje de modelado.

1.3.2 Framework de desarrollo

Un marco de trabajo (*framework*) es una gran librería o conjunto de librerías donde además de facilitar funciones para su uso, dispone de una sintaxis o metalenguaje específico del marco de trabajo y una forma de organización de su código. Para usar un marco de trabajo no basta con conocer el nombre de la función a utilizar: hay que saber qué sintaxis emplea, qué obligaciones impone (a la hora de organizar el

código, archivos) y su lógica o filosofía de trabajo. El uso de marcos de trabajo se ha extendido debido a la gran complejidad de las aplicaciones que se desarrollan ya que facilitan su organización y mantenimiento (Gómez, y otros, 2016)

Symfony

Symfony es uno de los *frameworks* PHP más populares entre los usuarios y las empresas. Permite que los programadores sean mucho más productivos a la vez que crean código de más calidad y más fácil de mantener. Es maduro, estable, profesional y está muy bien documentado. Es un marco de trabajo completo diseñado para optimizar el desarrollo de aplicaciones web (como la que se propone desarrollar en esta investigación). Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación (Symfony, 2017).

Django 2.1

Es un marco de desarrollo en Python, de código abierto y muy ligero que permite la creación rápida de páginas y aplicaciones web. Pretende ser sencillo y rápido posibilitando concentrarse principalmente en escribir la aplicación (Django, 2019).

Las principales características que presenta este marco de trabajo son:

- Basado en la filosofía DRY (*Don't Repeat Yourself*, no te repitas). Muchas aplicaciones web y proyectos digitales comparten numerosas líneas de código unos con otros. Django es el marco de desarrollo de refactorización de código casi por excelencia. Permite reutilizar programación de unas aplicaciones a otras sin la obligación de tener que repetir las mismas líneas de código entre distintos proyectos.
- Es de alto nivel basado en el patrón arquitectónico Modelo-Vista-Plantilla. Apuesta por la sencillez, la rapidez y la reutilización de código.
- Viene con SQLite, una base de datos usada por compañías tan importantes como Facebook o Bloomberg.
- Contiene una API (*Application Programming Interface*, Interfaz de Programación de Aplicaciones) propia para el desarrollo de proyectos digitales.

Fundamentación del *framework* a utilizar

Se decide utilizar Symfony porque es compatible con la mayoría de gestores de bases de datos y puede ser ejecutado en múltiples plataformas, razón para ser utilizado en el desarrollo de la presente investigación. Además, es un marco de trabajo basado en el patrón Modelo Vista Controlador (MVC) para el desarrollo de aplicaciones web, implementado en PHP y sigue la mayoría de mejores prácticas y patrones de diseño para la web, de lo que se puede concluir que con él se pueden crear aplicaciones y sitios web rápidos y seguros de una forma profesional.

1.3.3 Lenguaje de programación

PHP 7.2.9: Es un lenguaje de programación utilizado para la creación de sitios web. No necesita ser compilado para ejecutarse. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. PHP está diseñado específicamente para ser un lenguaje más seguro, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución y siguiendo algunas prácticas correctas de programación. Es multiplataforma y puede ser utilizado sobre los sistemas operativos: *GNU/Linux*, Windows, entre otros (PHP, 2017). Dentro de nuestro sistema este lenguaje es utilizado para desarrollar páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor.

HTML5: Como su nombre lo indica es la quinta revisión del estándar HTML y permite soportar multimedia, agrega elementos como video, audio y canvas, como la integración para gráficos vectoriales (SVG) y MathML para fórmulas matemáticas. Estas características permiten incluir y controlar contenido multimedia en la web sin tener que recurrir a *plugins* ni APIs propietarias. El HTML se utiliza en la traducción y descripción de la estructura y la información en forma de texto, así como para complementar el texto con objetos e imágenes (ABURY, 2014). En la presente investigación se utilizará para estructurar y presentar contenido en el sistema web que se propone desarrollar.

JavaScript 5.1: JavaScript se define como un lenguaje de programación interpretado, no requiere la compilación del código fuente para ejecutarlo. Este trabaja sobre un enfoque útil y práctico hace que el proceso de desarrollo de aplicaciones web dinámicas sea eficiente, está disponible en cualquier navegador web, es un lenguaje utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes orientados a objetos mucho más complejos. Este lenguaje

posee varias características, entre ellas se destaca que es un lenguaje basado en acciones que posee menos restricciones. Además, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros (GRADOS, 2016). Se utilizará en la propuesta de solución que se desarrollará para crear diferentes efectos e interactuar con el usuario que trabajará en el sistema web.

CSS 3: CSS es un lenguaje utilizado en la presentación de documentos HTML. Permite separar el contenido de la forma y proporciona a los diseñadores el mantenimiento de un control más preciso sobre la apariencia de las páginas web (Hidalgo, 2015). Su novedad más importante consiste en la incorporación de mecanismos para mantener un mayor control sobre el estilo de los elementos que se muestran en la página (BEATI, 2016). En nuestro sistema utilizamos este lenguaje para organizar, mejorar y desarrollar la presentación por las opciones de colores, tipos y tamaños de letra que nos brinda dicho lenguaje.

1.3.4 Entorno de desarrollo integrado

Según Canut, el *Integrated Development Environment* (IDE) es un programa compuesto por herramientas que facilita el trabajo a la hora de desarrollar un software. Consta de un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. El objetivo de un IDE se centra fundamentalmente en facilitar el acceso a datos y servicios, así como su interacción entre múltiples usuarios (Canut, 2006).

Fundamentación del IDE de desarrollo a utilizar

PhpStorm es un IDE de programación desarrollado por JetBrains. Es uno de los entornos de programación más completos de la actualidad; permite editar código no sólo del lenguaje de programación PHP como lo indica su nombre. Actualmente es compatible con Sistemas Operativos Windows, Linux y Mac OS X. Algo que destaca en PhpStorm es la ejecución de código en la misma interfaz del IDE, así como también la interpretación y visualización inmediata de código PHP hasta en 5 de los navegadores web más populares (JETBRAIN, 2017).

Se opta por este IDE ya que entre otras razones soporta el trabajo con la versión 7.2 de PHP, que es la que se utilizará en el desarrollo de la propuesta. Además, es uno de los más utilizados a nivel mundial y al ser multiplataforma permitirá trabajar en los sistemas operativos Windows y Linux, que son los más utilizados en la Universidad.

1.3.5 Servidor web

Un servidor web es un programa informático que procesa aplicaciones realizando conexiones bidireccionales y/o unidireccionales síncronas o asíncronas con el cliente, generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. Es quien gestiona la mayor parte de las funciones de lógica de negocio y de acceso a los datos de la aplicación. Los principales beneficios de la aplicación de la tecnología de servidores web son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones. Es un servidor web flexible, rápido y eficiente, altamente configurable por su diseño modular, esta característica permite ampliar considerablemente sus capacidades al existir un repositorio extenso completamente gratuito de extensiones y módulos. Es compatible con el lenguaje de programación PHP, comparten muchas de sus características (APACHE, 2017).

Apache es un servidor web que implementa el protocolo HTTP, es software libre y se puede instalar en los sistemas operativos *Windows* y *Linux*. Es de fácil configuración, pues su estructuración en módulos permite al usuario utilizar los servicios y funcionalidades que ofrece. Se define Apache como software para el servidor de aplicaciones en su versión 2.4.34, por su flexibilidad, rapidez, porque es gratuito, multiplataforma e interpreta varios lenguajes como PHP. Estas características promueven la eficiencia y rendimiento del sistema que debe ser capaz de dar respuestas a las peticiones con un nivel aceptable de desempeño, teniendo en cuenta la concurrencia que pueda existir; debe prestar servicios sin que se amplíen los rangos de tiempo de respuesta (APACHE, 2017).

1.3.6 Sistemas gestores de base de datos

Data Base Management System o Sistema de Gestión de Base de Datos (SGBD), es un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. Entre los principales gestores de Bases de Datos a nivel mundial se encuentran Oracle, SQL Server, MySQL y PostgreSQL. Symfony se integra a Doctrine, biblioteca que se utiliza para simplificar las operaciones que se realizan en las bases de datos, posee controladores que permiten la compatibilidad con servidores de código abierto como es el caso de MySQL (Pacheco, 2013).

MySQL:

Es un sistema de gestión de bases de datos relacional, multiusuario, de código abierto, se ofrece bajo la GNU/GPL para cualquier uso compatible con esta licencia. Proporciona un servidor de base de datos SQL (*Structured Query Language*) muy rápido. La principal función de este sistema es la velocidad y la robustez, para las columnas soporta gran cantidad de tipos de datos. Funciona sobre múltiples plataformas y sistemas operativos. Presenta un excelente nivel de seguridad en los datos y es fácil de configurar e instalar. Por su implementación multihilo, tiene un sistema flexible de gestión de usuarios y contraseñas. Su utilización está enfocada en aplicaciones web dinámicas escritas en PHP por la optimización de consultas sencillas y su compatibilidad con el servidor web Apache. Admite hasta 32 índices por tablas, la variedad de tipos de datos para las columnas y la utilización de disparadores (Oracle, 2017).

PostgreSQL:

Es un sistema de gestión de base de datos relacional orientado a objetos y libre, publicado bajo la licencia BSD. Presenta instalación ilimitada, es frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que permite la licencia. PostgreSQL ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que los productos de los proveedores comerciales, conservando todas las características, estabilidad y rendimiento. Es multiplataforma, está disponible en casi cualquier Sistema Unix (34 plataformas en la última versión estable), y una versión nativa de *Windows*. Diseñado para ambientes de alto volumen, PostgreSQL usa una estrategia de almacenamiento de filas llamada *Multi-Version Concurrency Control* (MVCC) para conseguir una mejor respuesta en ambientes de grandes volúmenes (PostgreSQL, 2019).

Selección de Sistema gestor de BD:

Con su última versión MySQL 5.7 anexa nuevas características importantes que lo hacen más competitivo con manejadores como Postgres, SQL Server y Oracle. MySQL incorpora características avanzadas en los temas de seguridad, soporte de aplicaciones, control y seguimiento, así como rendimiento y almacenamiento de datos especializados.

Para el desarrollo del sistema se decide utilizar MySQL como manejador de bases de datos porque es principalmente utilizado en las aplicaciones web en conjunto con PHP (lenguaje a utilizar en el desarrollo

de la propuesta de solución). Es veloz en cuanto a términos de procesamiento de operaciones y garantiza un mayor rendimiento cuando se consultan las bases de datos, ante servidores de bajas prestaciones tiene un bajo consumo y es un gestor fácil de administrar.

1.3.7 Metodología de desarrollo

Una metodología es un conjunto de procedimientos, técnicas, documentación y herramientas que se utilizan en la creación de un producto de software. Indica paso a paso lo que se debe realizar para obtener los distintos productos parciales y finales en el proceso de desarrollo de un software. Permite especificar las personas que van a participar en el proceso, así como el papel o rol que van a jugar en el mismo. El autor de la presente investigación considera que de forma general no existe una metodología que se pueda usar en todos los proyectos, por lo que se decide utilizar la metodología AUP (Proceso Unificado Ágil) en su variación para la UCI, que se adapta al ciclo de vida definido para la actividad productiva de la universidad. La variación de AUP está apoyada en el Modelo CMMI-DEV v1.3 (modelo en el que la universidad trabaja fuertemente por continuar su consolidación) y es una versión simplificada del Proceso Unificado de Racional (RUP). De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, a la que se denomina Ejecución y se agrega la fase de Cierre (Sánchez, 2015).

1.3.8 Herramientas para las pruebas de seguridad

Las pruebas de software, son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto a la parte interesada (Pressman, 2010). La propuesta de solución será validada a través de las pruebas de seguridad utilizando la herramienta **Acunetix Web Vulnerability Scanner**. Acunetix realiza automáticamente auditorías a aplicaciones web comprobando vulnerabilidades de Inyección SQL, Cross site scripting y otras vulnerabilidades que puedan ser explotadas por hackers. Ejecuta una serie de pruebas, totalmente configurables por el usuario, para identificar las vulnerabilidades tanto en la programación de la página como en la configuración del servidor (Acunetix, 2013).

Conclusiones parciales

En este capítulo se trataron elementos teóricos que sustentan la propuesta de solución del problema planteado, arribando a las siguientes conclusiones:

- El análisis de varios conceptos relacionados con el campo de acción permitió un mejor entendimiento de la investigación.
- Con la selección de la metodología, tecnologías y herramientas basadas en software libre se obtuvo una base tecnológica enfocada en el desarrollo de sistemas web.
- De las soluciones homologas estudiadas, se identificó que son sistemas con un alto nivel de aceptación pero cada una responde a necesidades particulares; aun así fue posible reconocer a partir de ellas, elementos que contribuyen y forman parte de la propuesta de solución.

CAPÍTULO 2: ANÁLISIS, PLANIFICACIÓN Y DISEÑO DEL SISTEMA DE GESTIÓN DE TECNOLOGÍAS

Para desarrollar un sistema informático es importante la utilización de métodos y técnicas que permitan la solución de problemas existentes a lo largo del ciclo de vida de un software. La correcta captura de requisitos y el modelado del sistema permiten que se mitiguen las fallas que puedan aparecer durante su desarrollo, además de establecer un entendimiento común entre lo que se desea y lo que se elabora. Dentro de este capítulo se obtienen un conjunto de artefactos de la metodología AUP los que son generados en cada fase y ayudan en el desarrollo del sistema informático. En la fase de Planificación se describen elementos como Historias de Usuarios y Plan de iteraciones. En el diseño se ejecutan las actividades requeridas para el desarrollo del software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura; se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño. Por último, se implementan las tareas definidas para cada historia de usuario.

2.1 Descripción de la propuesta de solución

La propuesta de solución se basa en el desarrollo del Sistema de Gestión de Tecnologías que será utilizado en la Facultad 1 de la Universidad de las Ciencias Informáticas, encargado de gestionar los procesos de obtención de información de *hardware* de medios tecnológicos pertenecientes a las áreas que posee dicha facultad. El sistema a desarrollar, facilita en buena medida el trabajo de los especialistas y directivos que interactúan con este tipo de información. Cuenta con funcionalidades que posibilitan el aumento de la agilidad y organización de este proceso. Facilita servicios de gestión de información y admite generar reportes para todos los usuarios del sistema.

La gestión de usuarios es uno de los elementos fundamentales y permite añadir, modificar o eliminar usuarios del sistema. El acceso a la información se realiza basado en roles que tienen definido los niveles de información y también pueden ser gestionados en el sistema.

Al estar la Facultad compuesta de varias áreas y éstas compuestas por locales, se permite realizar el proceso de gestión de las mismas, añadiendo, modificando o eliminando según las necesidades que se generen. Los medios que se encuentran en uso en cada local también deben ser gestionados para representar los cotidianos procesos de recepción o entrega y devolución de los mismos. Existen diferentes tipos de medios entre los que se encuentran monitores y CPU (Unidad Central de Procesamiento). Las

CPU están compuestas por varios elementos, como las memorias RAM y discos duros de los cuales se gestionan sus principales propiedades. El sistema gestiona como propiedades principales del medio su tipo, rótulo, número de serie, estado, local y área al que pertenece.

Los reportes generados en formato Excel son fundamentales para la toma de decisiones y por este motivo el sistema a desarrollar cuenta con 3 reportes que permiten obtener información sobre las cantidades de medios en existencia en cada local o área, los medios en explotación por cada local o área y los medios rotos por cada local o área. La figura 1, muestra la propuesta de solución del sistema:

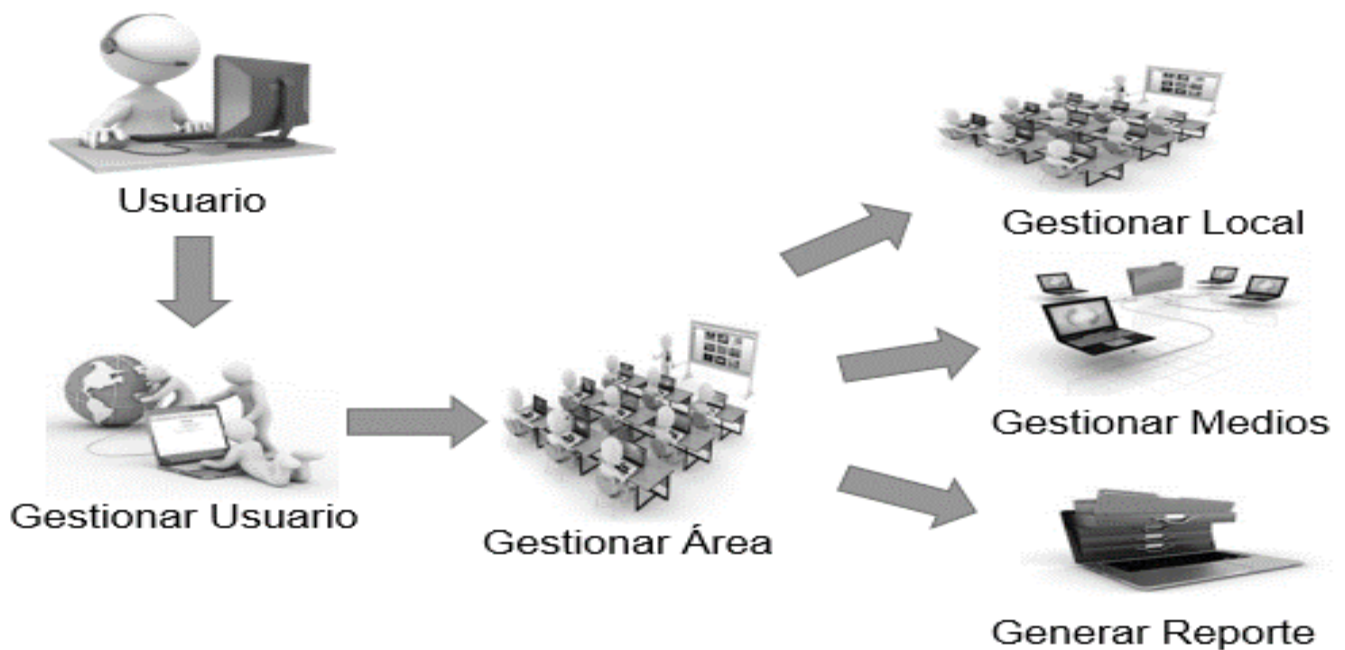


Figura 1: Imagen de propuesta de solución del sistema.

2.2 Requisitos de la propuesta de solución

Los requisitos son la base para un desarrollo exitoso, así como para una plena conformidad con el entregable final. La especificación de los requisitos de software establece una base sólida para su diseño e implementación. Estos se definen las necesidades, describen los escenarios de uso, delinear las funciones y características e identifican las restricciones del proyecto (Pressman, 2010). Para llevar a cabo el proceso de captura de requisitos se aplicó la técnica que a continuación se expone:

➤ Entrevista: se realizaron entrevistas al asesor de tecnología, sustentado en una serie de preguntas, con el objetivo de obtener la mayor cantidad posible de información y comprender los objetivos generales de la solución buscada.

La mayoría de los sistemas de software tienen un número elevado de requisitos, por lo que el desglose del mismo lo convierte en un poco más sencillo en el momento de su desarrollo. Durante la disciplina de captura de requisitos se identificaron 8 requisitos funcionales (RF) con lo que debe constar la aplicación para satisfacer las necesidades requeridas por el cliente.

2.2.1 Requisitos funcionales

Los requisitos funcionales (RF) son aquellos directamente relacionados con las funciones y las reacciones que el sistema debe proporcionar. Son asignados directamente a elementos o características del sistema de software (Sommerville, 2011). A continuación, en la siguiente tabla se muestran los requisitos funcionales:

Tabla 2. Requisitos funcionales.

#RF	Nombre	Descripción	Complejidad
RF1	Adicionar usuario	Debe permitir adicionar un usuario a la base de datos.	Baja
RF2	Listar usuario	Debe permitir listar los usuarios que se encuentran registrados en la base de datos.	Baja
RF3	Modificar usuario.	Debe permitir modificar los usuarios que se encuentran registrados en la base de datos.	Baja
RF4	Eliminar usuario	Debe permitir eliminar los usuarios que se encuentran registrados en la base de datos.	Baja

RF5	Asignar roles	Debe permitir asignar roles a los usuarios que se encuentran registrados en la base de datos.	Alta
RF6	Asignar permisos	Debe permitir asignar permisos a los usuarios que se encuentran registrados en la base de datos.	Alta
RF7	Autenticar usuario	Debe permitir que se verifiquen los datos de los usuarios que posee la base de datos.	Medio
RF8	Adicionar Área.	Debe permitir adicionar un área a la base de datos.	Medio
RF9	Listar Áreas	Debe permitir listar las áreas registradas en la base de datos.	Medio
RF10	Modificar Área	Debe permitir modificar un área registrada en la base de datos.	Medio
RF11	Eliminar Área	Debe permitir eliminar un área registrada en la base de datos.	Medio
RF12	Adicionar local	Debe permitir adicionar un local a la base de datos.	Medio
RF13	Listar locales	Debe permitir listar los locales registrados en la base de datos.	Medio
RF14	Modificar local	Debe permitir modificar un local registrado en la base de datos.	Medio
RF15	Eliminar local	Debe permitir eliminar un local	Medio

		registrado en la base de datos.	
RF16	Adicionar medios	Debe permitir adicionar un medio a la base de datos.	Alta
RF17	Listar medios	Debe permitir listar los medios registrados en la base de datos.	Alta
RF18	Modificar medio	Debe permitir modificar un medio registrado en la base de datos.	Alta
RF19	Eliminar medios	Debe permitir eliminar un medio registrado en la base de datos.	Alta
RF20	Generar reporte estadístico de la propiedad y cantidad de medios por local	Debe permitir generar un reporte mediante un documento Excel de la propiedad y cantidad de medios por local.	Alta
RF21	Generar reporte estadístico de la cantidad y el tipo de medios rotos por local	Debe permitir generar un reporte mediante un documento Excel de la cantidad y el tipo de medios rotos por local.	Alta
RF22	Generar reporte estadístico del total de medios existentes en el área	Debe permitir generar un reporte mediante un documento Excel del total de medios existentes en el área.	Alta

2.2.2 Requisitos no funcionales

Los requisitos no funcionales (RnF) son restricciones de las funciones ofrecidas por el sistema. Generalmente se aplican al sistema en su totalidad y surgen de las necesidades del usuario. Pueden especificar también la utilización de una herramienta en particular o un lenguaje de programación (Ramos,

2016). Para la solución propuesta se identificaron 13 RnF los cuales fueron agrupados por las diferentes categorías:

Seguridad:

RnF 1: El sistema debe verificar que el usuario esté autenticado antes de que pueda realizar alguna acción sobre el sistema.

RnF 2: Presencia de distintos roles que controlen las acciones que pueden realizar los usuarios.

RnF 3: El sistema debe mostrar las funcionalidades de acuerdo a los permisos del usuario que esté activo.

RnF 4: Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador.

Software:

Utilizar para pc_cliente:

RnF 5: Se necesita instalado en la estación de trabajo un sistema operativo: Linux, Windows 7 o superior.

Utilizar para pc_servidor:

RnF 6: Se requiere de la instalación del servidor web Apache en su versión 2.4.34 y PHP en su versión 7.0 o superior para poder visualizar la interfaz web.

RnF 7: Se requiere de un gestor de base de datos MySQL en su versión 5.7 o superior.

Hardware:

Utilizar para pc_cliente:

RnF 8: Requerimientos mínimos: Procesador Pentium IV a 1.40 GHZ con 512 MB de memoria RAM.

Utilizar para pc_servidor:

RnF 9: Requerimientos mínimos: Procesador Intel(R) Core (TM) i3-4160 CPU 3.60GHz de velocidad de procesamiento, 2GB de memoria RAM y una capacidad de 80GB de disco duro.

Usabilidad:

RnF 10: El sistema de gestión tecnológica debe ser intuitivo, facilitando la interacción del usuario con la aplicación.

RnF 11: Los dispositivos cliente que utilizarán la herramienta deben contar con navegadores web que soporten HTML5, CSS3 y JavaScript.

Eficiencia.

RnF 12: Toda funcionalidad del sistema de gestión tecnológica debe responder al usuario en menos de 5 segundos.

RnF 13: La información almacenada debe ser confiable en cuanto a su veracidad e integridad desde su recopilación a través de la validación de los formularios.

2.3 Historia de usuario

La metodología AUP-UCI, en su escenario 4 para la disciplina requisitos, genera las Historias de Usuario (HU), estas se utilizan para especificar los requisitos de software en la aplicación. Las HU se descomponen en tareas de programación y describen las características que el sistema debe cumplir, están escritas en un formato legible por el cliente, sin necesidad de sintaxis técnicas (Rodríguez, 2015).

A continuación, se muestra en las siguientes tablas varias historias de usuarios las cuales fueron generadas a partir de los requisitos funcionales y las mismas se estructuran de la siguiente forma:

Número: A cada HU se le asigna un número para facilitar su identificación por parte del equipo de desarrollo.

Nombre: Nombre descriptivo de la HU.

Prioridad: Grado de prioridad que le asigna el cliente a la HU en dependencia del valor en el negocio. Los valores que puede tomar son (Alta, Media o Baja).

Iteración Asignada: Número de la iteración en la cual será implementada la HU.

Programador: Nombre del programador encargado de desarrollar la HU.

Tiempo estimado: Esfuerzo estimado por el equipo de desarrollo para darle cumplimiento a la HU.

Tiempo real: Plazo real que el equipo de desarrollo tiene para dar cumplimiento a la HU.

Descripción: Descripción simple sobre lo que debe hacer la funcionalidad a la que se hace referencia.

Riesgo en Desarrollo: Grado de complejidad que le asigna el equipo de desarrollo a la HU luego de analizarla. (Alto, Medio o Bajo).

Tabla 3. HU_9 Listar Medios.

Historia de Usuario	
Número: 9	Nombre: Listar Medios
Prioridad: Alta	Iteración Asignada: 1
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 2 puntos	Tiempo real: 2 puntos
Descripción: Debe permitir listar los medios registrados en la base de datos.	
Riesgo en desarrollo: Alto	

Tabla 4. HU_20 Generar reporte estadístico de la propiedad y cantidad de medios por local.

Historia de Usuario	
Número: 20	Nombre: Generar reporte estadístico de la propiedad y cantidad de medios por local.
Prioridad: Alta	Iteración Asignada: 2
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 3 puntos	Tiempo real: 3 puntos
Descripción: Debe permitir generar un reporte mediante un documento Excel de la propiedad y cantidad de medios por local.	

Riesgo en desarrollo: Alto

Tabla 5. HU_21 Generar reporte estadístico de la cantidad y el tipo de medios rotos por local.

Historia de Usuario	
Número: 21	Nombre: Generar reporte estadístico de la cantidad y el tipo de medios rotos por local.
Prioridad: Alta	Iteración Asignada: 2
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 2 puntos	Tiempo real: 2 puntos
Descripción: Debe permitir generar un reporte mediante un documento Excel de la cantidad y el tipo de medios rotos por local.	
Riesgo en desarrollo: Alto	

Tabla 6. HU_22 Generar reporte estadístico del total de medios existentes en el área.

Historia de Usuario	
Número: 22	Nombre: Generar reporte estadístico del total de medios existentes en el área.
Prioridad: Alta	Iteración Asignada: 2
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 4 puntos	Tiempo real: 4 puntos
Descripción: Debe permitir generar un reporte mediante un documento Excel del total de medios existentes en el	

área.

Riesgo en desarrollo: Alto

2.4 Arquitectura de software

Una arquitectura de software define la forma de trabajar en un sistema, también debe dejar intuir el tipo de aplicación que describe. La Arquitectura de Software se refiere a las estructuras de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos. La misma es de especial importancia ya que la manera en que se estructura un sistema tiene un impacto directo sobre la capacidad de este para satisfacer lo que se conoce como los atributos de calidad del sistema. Ejemplos de atributos de calidad son el desempeño, que tiene que ver con el tiempo de respuesta del sistema a las peticiones que se le hacen, la usabilidad, que tiene que ver con qué tan sencillo les resulta a los usuarios realizar operaciones con el sistema, o bien la modificabilidad, que tiene que ver con qué tan simple resulta introducir cambios en el sistema. Los atributos de calidad son parte de los requerimientos (no funcionales) del sistema y son características que deben expresarse de forma cuantitativa (Cervantes, 2018).

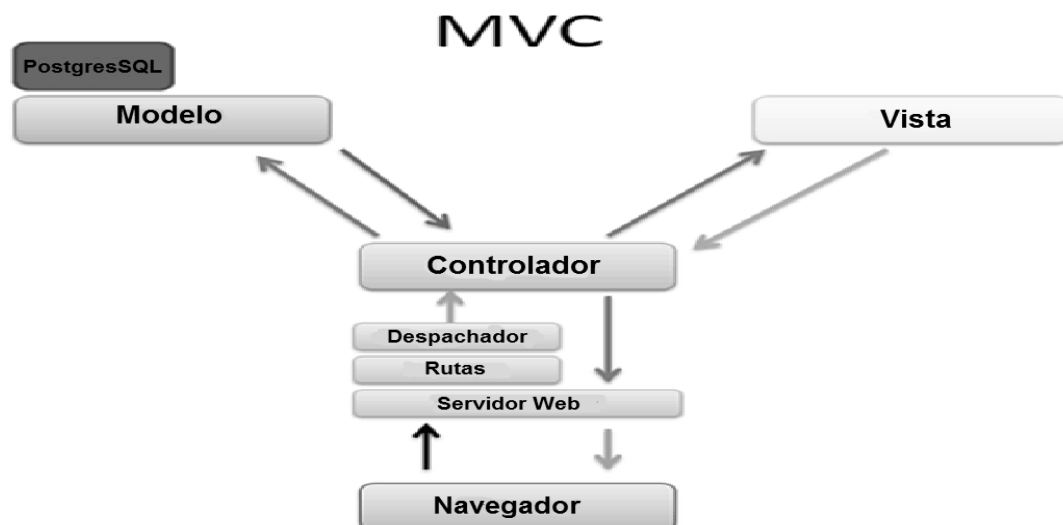


Figura 2. Arquitectura MVC.

Symfony basa su trabajo en el patrón arquitectónico Modelo – Vista – Controlador, como se muestra en la Figura 1, implementada de forma que el desarrollo de aplicaciones sea rápido y sencillo pues el código de la presentación se guarda en la vista, el código de manipulación de datos se guarda en el modelo y la lógica de procesamiento de las peticiones constituye el controlador. Este patrón está formado por tres niveles que a continuación se ejemplifican utilizando algunas de las clases o directorios que conforman la propuesta de solución desarrollada para la presente investigación:

1. **Modelo:** Representa la información con la que trabaja el sistema, su lógica de negocio. El modelo está representado en los archivos del directorio: **Entity**. Dentro de estos archivos se encuentra la información de cada entidad del sistema con sus respectivos atributos.
2. **Vista:** Transforma el modelo en una página web y permite al usuario interactuar con este. Las vistas del sistema están agrupadas en el directorio: **app/Resources/view**. En este directorio se encuentran todas las vistas que va a utilizar el modelo para mostrar al usuario.
3. **Controlador:** Se encarga de procesar las interacciones del usuario, y realiza los cambios apropiados en el modelo o la vista. Las clases controladoras de la propuesta de solución están situadas en el directorio: **Controller**.

En el controlador se encuentran las acciones que permiten gestionar los servicios, son el núcleo del sistema y contienen la lógica de la aplicación, dentro del mismo se encuentran todos los métodos que se van a ejecutar para las acciones del usuario en la interfaz, entre estos métodos se encuentran DeleteAction (objeto), NewAction (), ShowAction () y EditAction (). Cada acción que se realice desde el controlador utiliza el modelo para enviar las variables a la vista (**index.html.twig, edit.html.twig, show.html.twig y new.html.twig**), la misma es la encargada de mostrar las páginas como resultado de las acciones. El controlador de cada entidad del sistema (**areaController.php, localController.php y mediosController.php**) es la capa intermediaria entre la vista y el modelo.

2.5 Patrones de diseño

Un patrón de diseño se caracteriza como una regla de tres partes que expresa una relación entre cierto contexto, un problema y una solución. Para el diseño de software, el contexto permite entender el ambiente del problema y la solución apropiada. Un conjunto de requerimientos, incluidas limitaciones y

restricciones, actúan como sistema de fuerzas que influyen en la interpretación del contexto del problema y cómo podría aplicarse con eficacia la solución (Pressman, 2010). Durante la implementación de la propuesta de solución se utilizaron los siguientes patrones de diseño:

2.5.1 Patrones Generales de Software para la Asignación de Responsabilidades (GRASP)

Los patrones GRASP se utilizan para la asignación de responsabilidades como sus siglas en inglés lo indican (*General Responsibility Assignment Software Patterns*), pero más que patrones se consideran una serie de buenas prácticas de aplicación recomendable en el diseño de software nos guían para ayudarnos a encontrar los patrones de diseño (Larman, 2006).

Controlador: Un Controlador es un objeto de interfaz no destinado al usuario, se encarga de manejar un evento del sistema a una clase que represente el sistema global. En el sitio, los eventos generados por el usuario son redirigidos a una clase controladora que realiza las operaciones solicitadas.

Experto: Asigna la responsabilidad de hacer una labor a aquellos objetos que tienen la información para hacerlo. En el sistema, se pone de manifiesto el uso de este patrón, cuando se realizan cambios en las propiedades de algún medio. La clase Medio.php contiene los atributos necesarios para realizar las modificaciones, esta clase se puede llamar clase experta en información.

Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos para lograr menor dependencia y mayor oportunidad en la reutilización del código. Si se asignan bien el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización. Dentro del sistema este patrón se evidencia en la función *EditAction* la cual es la encargada de tomar los nuevos datos que el usuario pase por parámetro para editar el medio.

Alta cohesión: Cada clase de nuestro diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de las clases. Las mismas poseen un número relativamente pequeño de responsabilidades y realizan solo las funciones para las cuales fueron creadas, esto tiene como resultado un bajo acoplamiento y fomenta la reutilización. En el sistema este patrón se pone de manifiesto en la clase controladora *areaController* las misma confiere responsabilidades a la clase *AreaType* para la creación de formularios relacionados con la entidad *Area.php*.

Bajo acoplamiento: Constituye la relación o dependencia entre las clases del proyecto. Una clase con bajo acoplamiento no depende de muchas clases.

2.5.2 Patrones Gang of Four (GOF)

Describen las formas comunes en que diferentes tipos de objetos, pueden ser organizados para trabajar unos con otros. Gestionan la relación entre clases y la formación de estructuras de mayor complejidad. Además, permiten crear grupos de objetos para ayudar a realizar tareas complejas (Carlos A. Guerrero, 2013). Los patrones GOF utilizados en el sistema se relacionan a continuación:

Estructurales:

Decorator (Decorador): Permite añadir responsabilidades adicionales a un objeto dinámicamente para proporcionar una alternativa flexible a la especialización mediante herencia si se añaden nuevas funcionalidades. En el sistema se observa este patrón en la vista **base.html.twig** de ella heredan el resto de las páginas de la aplicación. Esta vista contiene los elementos que son comunes para el diseño y estructura del sistema.

Otros:

Registry: Este patrón es útil para los desarrolladores en la Programación Orientada a Objetos. Es un medio sencillo y eficiente de compartir datos y objetos en la aplicación, sin la necesidad de conservar numerosos parámetros o utilizar variables globales. Se aplica en la clase de configuración (config.yml) que es la encargada de guardar las variables globales del sistema, como drivers de conexión a la base de datos y rutas bases del proyecto.

2.6 Diagrama de clase del diseño

Un Diagrama de Clase del Diseño muestra la especificación de las clases de una aplicación, sus asociaciones, atributos y métodos, interfaces, navegabilidad y dependencias. El diagrama de clases recoge las clases de objetos y sus asociaciones. En este diagrama se representa la estructura y el comportamiento de cada uno de los objetos del sistema y sus relaciones con los demás objetos. Con el fin de facilitar la comprensión del diagrama, se pueden incluir paquetes como elementos del mismo, donde

cada uno de ellos agrupa un conjunto de clases (Pressman, 2010). A continuación, en la Figura 2, se expone un diagrama de clase del diseño (Figura 2) el cual fue generado para la presente investigación:

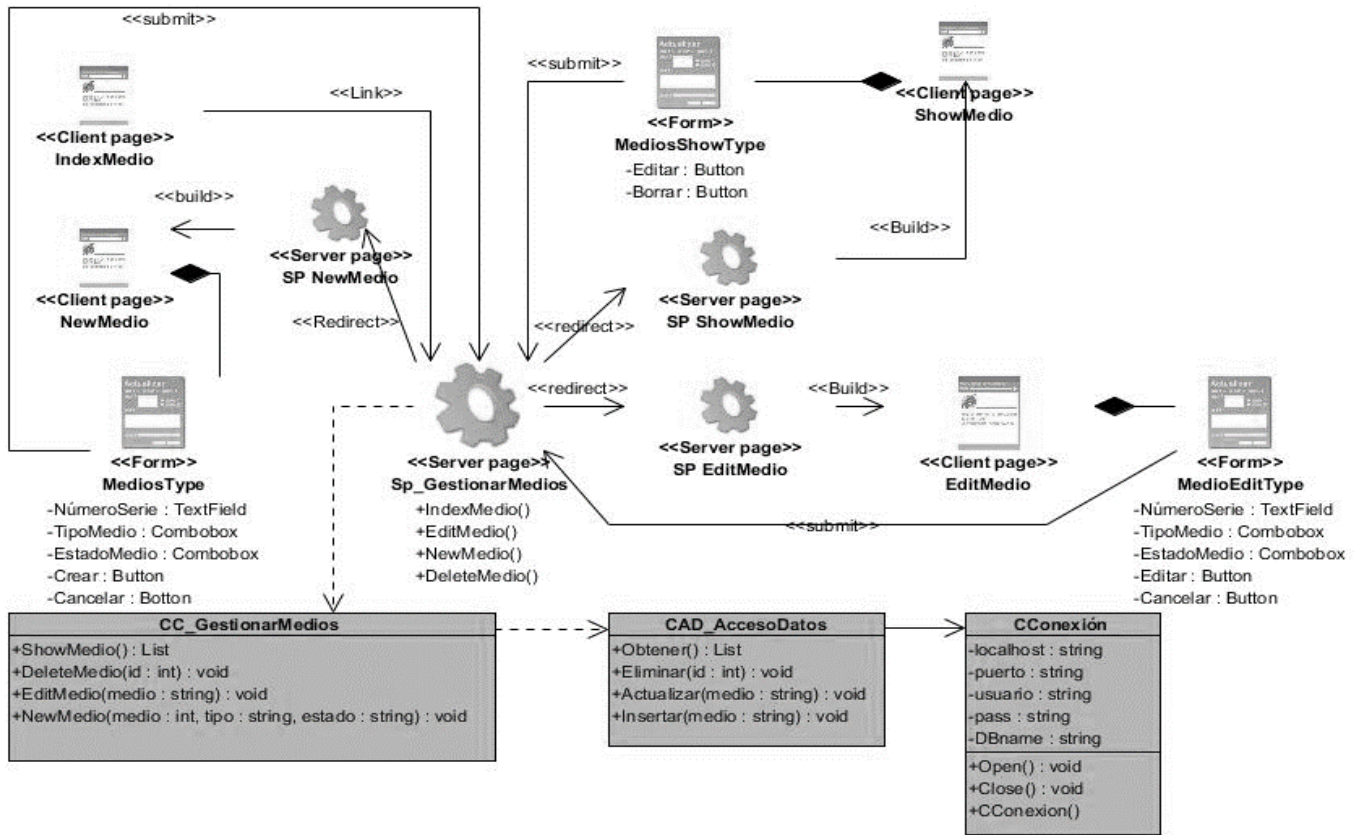


Figura 3: Diagrama de clase de diseño "Gestionar Medio".

2.7 Modelo de datos

Un modelo de datos es un conjunto de herramientas conceptuales encargadas de describir la representación de los datos en una base de datos. Los mismos comprenden aspectos relacionados con estructuras y tipos de datos, operaciones y restricciones (Rouse, 2016). A continuación, se expone el Modelo de datos (Figura 3) el cual fue generado para la presente investigación:

Usuario: Registra la información de los usuarios del sistema. Se almacena en la base de datos los atributos que lo identifican id_Usuario, Usuario, Password, Rol y Cargo.

Facultad: Registra la información de las facultades del sistema. Se almacena en la base de datos los atributos que lo identifica id_Fac y Nombre.

Área: Registra la información de las áreas del sistema. Se almacena en la base de datos los atributos que lo identifica id_Área, Nombre y Tipo.

Local: Contiene toda la información de cada local perteneciente al área y registra la información del área en el sistema. Se almacena en la base de datos los atributos que lo identifica (id_local) y Nombre.

Medio: Registra la información de los medios del sistema pertenecientes al local. Almacena en la base de datos los atributos id_Medios, NúmeroSerie, Tipo y Estado.

Monitor: Contiene la información de los monitores que tiene el local. Almacena en la base de datos los atributos id_Monitor, Marca, Modelo, Tipo y NúmeroSerie.

CPU: Contiene la información de los CPU que tiene el local. Almacena en la base de datos los atributos id_Cpu, MotherBoard, MicroProcesador y Socket.

RAM: Contiene la información de las memorias RAM que tiene el CPU. Almacena en la base de datos los atributos id_Ram, Slot, Capacidad y Tipo.

HDD: Contiene la información de los discos duros que tiene el CPU. Almacena en la base de datos los atributos id_Hdd, Modelo, Capacidad, Fabricante y NúmeroSerie.

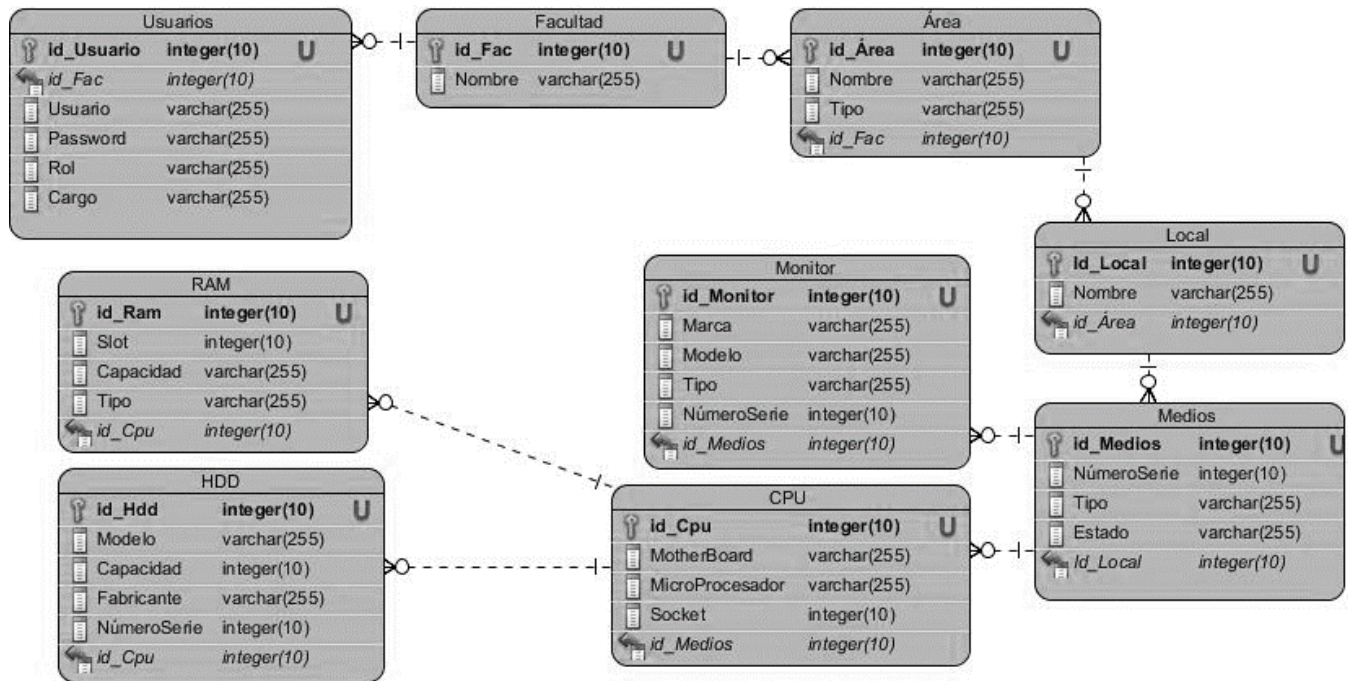


Figura 4. Modelo de datos.

2.8 Modelo de despliegue

Un Modelo de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de *hardware* (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos (Sarmiento, 2016). Un Nodo es un elemento de *hardware* o software. Esto se muestra con la forma de una caja en tres dimensiones. A continuación, se expone el Modelo de despliegue (Figura 4) el cual fue generado para la presente investigación:



Figura 5: Modelo de despliegue.

PC_Cliente: Representa la computadora que utiliza el asesor de tecnologías o jefe de departamento de tecnologías desde la misma pueden consultar y actualizar la información que se encuentra en el Server

Web. La comunicación entre las Pc_Cliente y el Server Web se establece utilizando el protocolo HTTPS para el envío de paquetes de forma segura por la red, evitando que se creen rastros de la información enviada en el navegador o servidor web.

Server Web: Es el nodo que realiza las funciones de intermediario entre las Pc_Cliente que realizan sus peticiones y el servidor donde se encuentra la información o servidor de base de datos. El servidor web toma las peticiones, mediante el protocolo HTTPS, realizadas por la Pc_Cliente, efectúa las consultas necesarias, mediante el Protocolo de Control de Transmisión (TCP), para validar que dicha información se encuentra en la base de datos, y presenta la respuesta.

Server_BD: El servidor de base de datos utiliza el protocolo TCP para la conexión con el servidor web.

Conclusiones parciales

En este capítulo se evidencian las principales características y elementos significativos de la propuesta de solución, así como los diferentes requisitos funcionales y no funcionales, por lo que se puede constatar que:

- La definición de los requisitos funcionales y no funcionales obtenidos a partir del proceso de identificación de los requisitos, garantizó que la solución responda a las necesidades del cliente, sirviendo de guía para el desarrollo de las funcionalidades del sistema de gestión tecnológica.
- Los artefactos generados constituyeron una guía fundamental para el desarrollo del sistema de gestión tecnológica.
- La arquitectura MVC seleccionada permitió definir la estructura del software y su interrelación entre los diferentes componentes.
- El modelo de despliegue propuesto permitió exponer como se relacionan los elementos de hardware del sistema.

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA DE GESTIÓN DE TECNOLOGÍAS

Con el objetivo de asegurar que el sistema propuesto funciona de acuerdo a los requisitos definidos y materializar en forma de componentes la fase de análisis y diseño, en el presente capítulo se describe la etapa de implementación y codificación del sistema. Se documentan los resultados obtenidos al aplicar las estrategias de pruebas, utilizadas para comprobar la calidad del software durante la etapa de validación.

3.1 Diagrama de Componentes

El diagrama de componentes (se describe en la Tabla 6) proporciona una visión física de la construcción del sistema. Muestra la organización de los componentes software, sus interfaces y las dependencias entre ellos (Ramos, 2015). Basándose en la arquitectura de software que propone el marco de trabajo Symfony, seleccionado en la fase de análisis y diseño en la Figura 5 se muestra el diagrama de componentes del sistema de gestión de tecnologías.

Tabla 7. Descripción del Diagrama de Componentes.

Sistema de Gestión de Tecnologías		
Componentes		Descripción
Controlador	MediosController.php	Clase controladora que contiene las acciones relacionadas con la información de los medios. Realiza acciones generales de gestión (insertar, eliminar, mostrar, editar y buscar).
Vista	index.html.twig	Muestra un listado de los medios registrados en el sistema.
	new.html.twig	Permite registrar un nuevo medio en el sistema a través de un formulario.

	edit.html.twig	Permite editar la información de los medios a través de un formulario.
	Delete.html.twig	Permite gestionar y modificar la lista de los medios registrados en el sistema mediante formularios.
	Formulario	Clase responsable de construir los formularios relacionados con los componentes de filtrado del listado de medios.
Modelo	Medios.php	Clase entidad que almacena los datos de los componentes y las configuraciones.
routin.yml	Contiene el mapa de rutas URL que se enlazan a las acciones de los controladores del sistema.	
config.yml	Contiene las configuraciones generales del Sistema de Gestión de Tecnologías.	

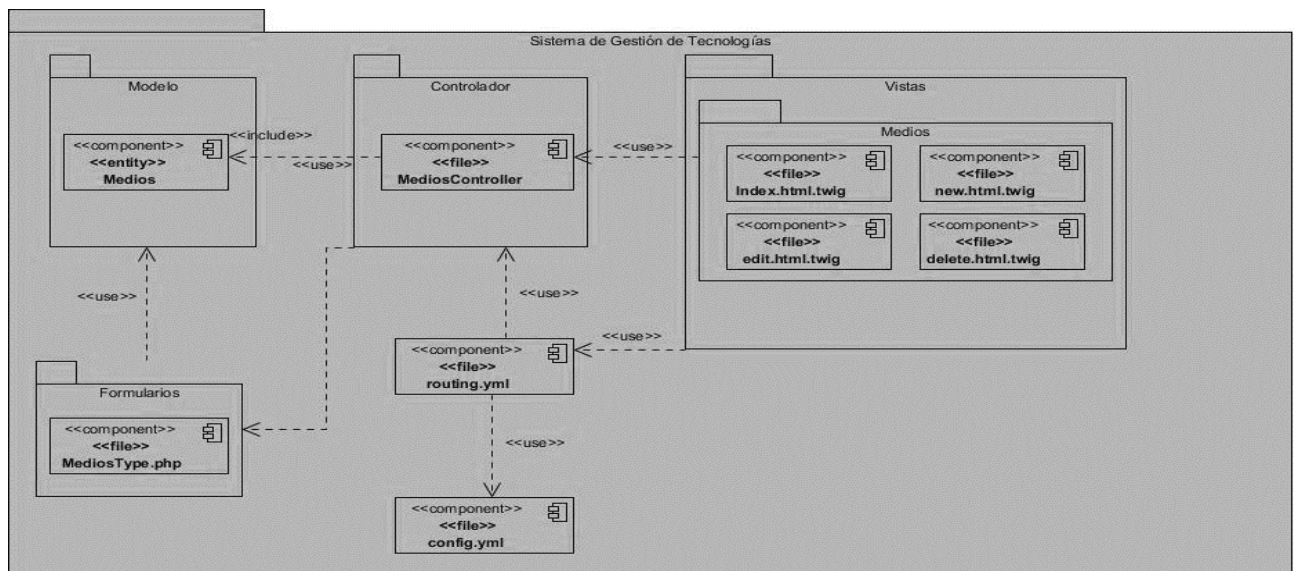


Figura 6. Diagrama de Componentes.

3.2 Estándares de codificación

Los estándares de codificación son un conjunto de convenciones para lograr uniformidad en el código fuente de un software. Permiten una mayor comprensión, modificación, calidad y mantenibilidad del código generado (Standards, 2018). En la siguiente tabla se definen los estándares de codificación utilizados para la implementación del sistema:

Tabla 8: Estándares de codificación.

Tipo de estándar	Descripción
Organización del código	<p>1. La apertura de llaves para clases y funciones deben estar en la próxima línea después de su declaración.</p> <p>2. Las aperturas de llaves en las estructuras de control se colocan en la misma línea.</p> <p>3. El cierre de llaves para clases, funciones y estructuras de control deben estar en la próxima línea después del bloque de código.</p> <pre data-bbox="386 1100 1214 1423"> public function deleteAction(Request \$request, Medios \$medio) 1 { \$form = \$this->createDeleteForm(\$medio); \$form->handleRequest(\$request); if (\$form->isSubmitted() && \$form->isValid()) { 2 \$em = \$this->getDoctrine()->getManager(); \$em->remove(\$medio); \$em->flush(); 3 } return \$this->redirectToRoute('medios_index'); } </pre>
Codificación	<p>1. Los archivos PHP deben utilizar solo las etiquetas de apertura <code><?php</code> omitiendo las etiquetas de cierre <code>?></code></p> <p>2. Utilizar la codificación UTF-8.</p>
Espacios en blanco	<p>1. Se debe utilizar un espacio en blanco después de la palabra clave en las estructuras en</p>

<p>expresiones y sentencias</p>	<p>de control.</p> <p>2. No deben existir espacios en la apertura y cierre de paréntesis.</p> <p>3. Debe usarse un espacio en blanco entre los operadores lógicos, aritméticos, de comparación y asignación.</p> <pre> if(1(\$form->isSubmitted()) &&3 \$form->isValid()) { \$em = \$this->getDoctrine()->getManager(); \$em->remove(\$medio); \$em->flush(); 2 } </pre>
<p>Líneas</p>	<p>1. El tamaño máximo de una línea no excederá de ochenta (80) caracteres.</p> <p>2. Se deberá agregar una línea en blanco delante de la sentencia return.</p> <pre> \$deleteForm = \$this->createDeleteForm(\$medio); 1 return \$this->render('medios/show.html.twig', array('medio' => \$medio, 'delete_form' => \$deleteForm->createView(),)); } </pre>

3.3 Validación de la propuesta de solución

Según la ISO/IEC/IEEE International Standard (2017), la fase de validación es el proceso de evaluar la calidad del software, con el objetivo de detectar posibles errores y corregirlos enfocándose en la lógica interna y los requerimientos especificados. A continuación, se documentan los resultados de las pruebas realizadas al Sistema de Gestión de Tecnologías, con el objetivo de evaluar su calidad y correcto funcionamiento.

3.3.1 Criterio de expertos

El criterio de expertos constituye una opinión informada de personas con trayectoria en el tema, que son reconocidas por otros como expertos cualificados en éste, y que pueden dar información, evidencia, juicios y valoraciones. De esa forma, resulta trascendente la elección de los jueces, ya que es importante dilucidar si se trata de personas conocedoras de la temática por su formación académica o su experiencia laboral. Aunado a ello, cabe considerar sus ocupaciones y concertar los tiempos para la realización de la tarea de validación (Alarcón, y otros, 2017).

Con el objetivo de validar el sistema web desarrollado para la gestión de los medios tecnológicos de la Facultad 1 se utiliza el método Criterio de Expertos siguiendo los pasos que a continuación se detallan:

- Identificar los posibles expertos.
- Seleccionar los expertos.
- Realizar el procesamiento de la información obtenida.

La experiencia de trabajo en el área de las tecnologías y gestión de medios tecnológicos es la condición principal para obtener los expertos que evaluarán la aplicación desarrollada. En la siguiente tabla (**Tabla 8**) se muestran los expertos seleccionados.

Tabla 9. Selección de los expertos.

No.	Experto	Entidad	Años de experiencia
1	Serguey Gonzales Garay	Facultad 1	6
2	Leiny Pons Flores	CIDI	5
3	Yojanhy Chávez Marrero	CIDI	2
4	Adrián Castillo Chávez	Facultad 1	4
5	Sahilyn Delgado Pimentel	CIDI	2

En la siguiente tabla se encuentra el resultado de los juicios emitidos por los expertos encuestados, de acuerdo a los siguientes parámetros:

1. ¿Cómo valora las funcionalidades del sistema web para la gestión de medios tecnológicos?
2. ¿Cómo considera la utilización de los elementos que componen la interfaz gráfica de usuario?

3. ¿Cómo evalúa la aplicabilidad y flexibilidad del sistema web desarrollado?

Con el objetivo de procesar y analizar la información que se obtuvo de la encuesta realizada se analizaron las respuestas de cada uno de los parámetros que aparecen en la encuesta. A continuación, se presentan los resultados a partir de la siguiente clasificación: **MA**: Muy adecuado, **A**: Adecuado, **PA**: Poco adecuado y **NA**: No adecuado. En la **Tabla 9**, N se refiere a la cantidad de encuestados que emitieron una valoración determinada y % al por ciento que representa con respecto al total de encuestados.

Tabla 10. Resultado de las encuestas aplicada al grupo de experto.

Parámetros evaluados	Niveles de valoración									
	MA		A		PA		NA		Total	
	N	%	N	%	N	%	N	%	N	%
1	5	100	0	0	0	0	0	0	5	100
2	4	80	0	0	1	20	0	0	5	100
3	4	80	1	20	0	0	0	0	5	100

Al analizar los resultados de la encuesta realizada, se obtiene que el 100% de los encuestados consideran de muy adecuadas las funcionalidades del sistema web desarrollado. Respecto a la utilización de los elementos que conforman la interfaz gráfica de usuario, el 80% la considera muy adecuada mientras que el 20% la considera poco adecuada. La aplicabilidad y flexibilidad es evaluada de muy adecuada por el 80% de los expertos mientras que un 20% de ellos la califica de adecuada.

Llegado a este punto se puede confirmar que el sistema web desarrollado tiene una buena aceptación entre los expertos. Los elementos evaluados fueron valorados todos entre los niveles de adecuado y muy adecuado demostrando el cumplimiento del objetivo planteado.

3.3.3 Pruebas funcionales

Las pruebas funcionales se realizan para comprobar que las funcionalidades implementadas cumplan con las especificaciones de los requisitos definidos. Para desarrollar esta prueba, el autor selecciona la técnica de Caja Negra o también llamadas pruebas de comportamiento, las cuales se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa (Pressman, 2010).

A continuación, se representa un ejemplo de diseño de casos de prueba de aceptación, utilizado para detectar errores en la aplicación y comprobar que se cumplieron los requisitos funcionales descritos en el capítulo anterior. Se describe el caso de prueba **Listar Medio**:

Tabla 11. Caso de prueba Listar medio.

Caso de prueba Listar Medio.	
Código de caso de prueba: 1	Nombre de Historia de Usuario: Listar Medio
Nombre de la persona que realiza la prueba: Leonardo José Santiesteban Cantallops	
Descripción de la prueba: El usuario se autentica en el sistema. Posteriormente accede desde la interfaz principal a la funcionalidad listar medio. El sistema muestra el listado de todos los medios que se encuentran registrados y guardados en la base de datos.	
Entrada / Pasos de la ejecución: La tabla que muestra los medios cuenta con los siguientes campos: Tipo: Tipo de medio que se encuentra registrado en la base de datos (CPU, monitor). Rótulo: Identificador del medio, conjunto de números y letras que conforman el id único de cada medio. Número de serie: Número que asigna el fabricante a cada medio. Local: Local al que pertenece el medio. Área: Área a la que pertenece el medio.	
Resultado esperado: Se muestra el listado de los medios que se encuentran registrados en el sistema.	

Evaluación de la prueba: Satisfactoria.

En total, se planificaron 2 iteraciones de prueba. La figura 7 aporta la información del resultado, total de no conformidades encontradas y las que se resolvieron por cada iteración. Para un total de 8 requisitos funcionales se detectaron 20 no conformidades en la primera iteración y se resolvieron 20, las cuales se solucionaron satisfactoriamente, y en la segunda iteración se redujo las no conformidades a 0, dando como resultado que fueran resueltas todas las no conformidades que se encontraron en la aplicación de la prueba.



Figura 7. Gráfico de no conformidades.

Entre las no conformidades detectadas al aplicar las pruebas funcionales se encuentran:

- Errores de los contenidos mostrados en las vistas.
- Opciones que no funcionan de las interfaces de usuario a la hora de ejecutar alguna acción en las vistas.
- Los mensajes presentan problemas de idioma y faltas de ortografías.

Para dar solución a los errores de contenido mostrados en las vistas se hizo una búsqueda exhaustiva en el código hasta encontrar dichos errores, los mismos fueron corregidos recodificando las líneas que presentaban problemas. La no conformidad a la hora de cargar las vistas del sistema fue resuelta

cambiando los links de cada opción para realizar las redirecciones hacia las vistas que pide el usuario. La no conformidad de ortografía fue resuelta corrigiendo los errores presentes en los formularios que generan cuando se muestran las vistas.

3.3.3 Pruebas de usabilidad

Para garantizar la seguridad del sistema desarrollado se realizaron pruebas de usabilidad basada en la lista de chequeo, herramienta utilizada en la UCI en los laboratorios de producción de software para agilizar los procesos de prueba, teniendo en cuenta que constituyen la guía básica y única para el probador en la revisión de los artefactos.

Además, son un apoyo en la ejecución de evaluaciones estáticas garantizando una mayor calidad en los artefactos de apoyo a los sistemas desarrollados por la universidad, estableciendo un orden de revisión por subtítulos y acápite que coinciden con la organización del artefacto a evaluar.

Formas de uso:

Evaluación: Es la forma de evaluar el indicador en cuestión. El mismo se evalúa de 1 en caso de mal (cuando la respuesta al indicador sea “No”) y 0 en caso que elemento revisado no presente errores (cuando la respuesta al indicador sea “Sí”).

NP (No Procede): Se usa para especificar que el indicador a evaluar no se puede aplicar en ese caso.

Tabla 12. Indicadores de la categoría. Visibilidad del sistema.

Visibilidad del sistema	Evaluación	NP
¿El sistema refleja la identidad de la empresa, logos, compañía...)?	0	
¿Cuándo se selecciona un icono se diferencia de los no seleccionados?	0	
¿Los enlaces del menú se resaltan cuando se seleccionan?	0	
¿Los iconos que aparecen se identifican claramente con lo que representan?	0	
¿El menú de navegación aparece en un lugar destacado?		X

¿No utiliza más de siete opciones principales en el menú de navegación?	0	
¿Si la respuesta a una acción se retrasa, aparece un mensaje o indicio como que el sistema está procesando la acción?		X
¿El sitio le indica al usuario en que parte de la estructura del sitio web se encuentra, es decir si muestra 'migas de pan'?		X
¿Los títulos de las tablas son descriptivos y distintivos?	0	
¿No se muestran errores ortográficos?	1	

Tabla 13. Indicadores de la categoría. Lenguaje común entre sistemas y usuario.

Lenguaje común entre sistema y usuario	Evaluación	NP
¿El lenguaje es simple, con un tono adecuado?	0	
¿La información que se presenta en la aplicación es fácil de entender y memorizar?	0	
¿Utiliza los conceptos establecidos para las funciones estándar? ("buscar" para las búsquedas, etc.)		X
¿Evita el lenguaje técnico: términos informáticos o propios de Internet?	0	
¿Los acrónimos y abreviaturas son definidos al ser usados por primera vez?		X
¿No hace uso de términos extranjeros?		X
¿Utiliza un texto específico y descriptivo en los vínculos?	0	
¿La información es de rápida lectura, y con una disposición asequible?	0	

De las tablas anteriores concluimos con 18 indicadores de usabilidad usados por el sistema desarrollado, de los mismos 1 presenta errores que se tienen que corregir y 11 que no presentan ningún tipo de error de

usabilidad, el resto no proceden para la aplicación desarrollada. De los 12, el sistema solo cumple con 11 indicadores, representando un 91 % de usabilidad para las funciones presentes.

3.3.4 Pruebas de Seguridad

El objetivo de las Pruebas de Seguridad es medir la confidencialidad, integridad y disponibilidad de los datos, partiendo a identificar amenazas y riesgos desde el uso o interface de usuario final. Durante las pruebas de seguridad, el responsable de la prueba desempeña el papel de un individuo que desea entrar en el sistema. Debe intentar conseguir las claves de acceso por cualquier medio, debe bloquear el sistema, negando así el servicio a otras personas. Una vez ejecutadas las Pruebas de Seguridad es posible medir y cuantificar los riesgos a los cuales se ven expuestos los aplicativos tanto en la infraestructura interna como externa, así como la identificación de vulnerabilidades (Pressman, 2010). Para realizar las pruebas de seguridad al sistema, se utiliza la herramienta *Acunetix Web Vulnerability Scanner*, con el objetivo de validar la seguridad de la solución propuesta, la cual arrojó los siguientes resultados en la segunda iteración:

Tabla 14: Resultado de las pruebas de seguridad

Categoría de vulnerabilidades	Cantidad de errores
Inyección de lenguaje de expresión	1
Formulario HTML sin protección CSRF	1
Credenciales de usuarios enviadas en texto plano	4
Enlaces rotos	2
Total	8

Se detectó que el *Quadratic2.php*, script dentro del bundle *ExcelBundle* utilizado para exportar información a Excel, presentaba vulnerabilidades ante los ataques de *Expression Language Injection*, para solucionar esta vulnerabilidad se validaron estrictamente los datos dentro de este script. Se identificó un formulario *HTML* vulnerable a la falsificación de petición en sitios cruzados (CSFR). El formulario fue comprobado de

forma manual como recomienda la herramienta *Acunetix* y se identificó que no puede ser utilizado por el atacante para alterar el sistema ni obtener ningún tipo de información. Las credenciales de usuario se estaban enviando mediante el protocolo HTTP lo cual hace que las mismas estén comprometidas cuando se realiza el login o registro de un usuario, para solucionar esta vulnerabilidad se cambió el protocolo de envío de dicha información a HTTPS. Dentro del sistema se identificaron 2 enlaces rotos, estos fueron eliminados pues pertenecían a la estructura original del marco de trabajo Symfony y no serían utilizados en nuestro sistema.

CONCLUSIONES PARCIALES

En este capítulo se abordaron una serie de aspectos correspondientes a la implementación y validación del sistema llegando a las siguientes conclusiones:

- La representación y descripción del diagrama de componentes permitió visualizar con más facilidad la estructura general del sistema
- La utilización de estándares de codificación permitió adoptar una estructura homogénea que facilita la comunicación y asegura el fácil mantenimiento del sistema.
- La ejecución de pruebas al sistema desarrollado permitió detectar las deficiencias presentes, subsanarlas en el menor tiempo posible y ofrecer un sistema seguro y usable.

CONCLUSIONES

Una vez completada la presente investigación, se puede concluir que:

- La investigación sobre las relaciones existentes entre los principales conceptos asociados al dominio de la presente investigación permitió una mayor comprensión de la propuesta de solución.
- El estudio del estado del arte de los sistemas de gestión de tecnología y el análisis de las diferentes herramientas y tendencias para la gestión tecnológica permitieron determinar las características que constituyen la base para el diseño de las funcionalidades que se definen en la propuesta de solución.
- La elaboración de los artefactos propuestos por la metodología de desarrollo y el levantamiento de requisitos permitieron una mayor comprensión, identificación de los procesos y características del sistema desarrollado.
- La utilización de la estrategia de pruebas garantizó la identificación temprana de las deficiencias, que al ser corregidas aumentaron la calidad del sistema de gestión de tecnologías.

RECOMENDACIONES

Para el desarrollo de futuras investigaciones relacionadas con la presente, se recomienda:

- Adicionar una funcionalidad que permita agregar nuevos tipos de medio.
- Adicionar una funcionalidad para gestionar facultades.

REFERENCIAS BIBLIOGRÁFICAS

1. **ABURY, Christopher. 2014.** *HTML5 y CSS3: para sitios con diseño web responsive*. Barcelona : Ediciones ENI, 2014.
2. **Acunetix. 2013.** Acunetix Web Vulnerability Scanner. *Acunetix Web Vulnerability Scanner*. [En línea] Acunetix Ltd., 2013. <https://www.acunetix.com>.
3. **Alarcón, Liliana Aidé Galicia, Trápaga, Jorge Arturo Balderrama y Navarro, Rubén Edel. 2017.** Validez de contenido por juicio de expertos: propuesta de una herramienta virtual. *Validez de contenido por juicio de expertos: propuesta de una herramienta virtual*. [En línea] octubre de 2017. [Citado el: 10 de junio de 2019.] http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1665-61802017000300042. ISSN 2007-1094.
4. **Alzate, José Benjamín Gallego. 2005.** Fundamentos de la gestión tecnológica e innovación. *Fundamentos de la gestión tecnológica e innovación*. [En línea] 15 de diciembre de 2005. [Citado el: 10 de junio de 2019.] www.redalyc.org. ISSN: 0123-7799.
5. **APACHE. 2017.** What is the Apache HTTP Server Project? [En línea] 2017. [Citado el: 30 de abril de 2019.] https://httpd.apache.org/ABOUT_APACHE.html.
6. **Ávila, Migdely B. Ochoa, Soa, Mario Valdés y Aballe, Yovanni Quevedo. 2007.** Scielo. *Scielo*. [En línea] 16 de abril de 2007. [Citado el: 6 de junio de 2019.] <http://scielo.sld.cu/pdf/aci/v16n4/aci081007.pdf>. ISBN.
7. **BEATI, Hernan. 2016.** *HTML5 y CSS3-Para diseñadores*. Buenos Aires : Alfaomega Grupo Editor, 2016.
8. **Benitez-Odio, Mileisys, y otros. 2017.** Gestión tecnológica en la relación universidad-empresa estatal ganadera cubana. Parte I. Estudio de caso: fundamentación y propuesta de un modelo. *Gestión tecnológica en la relación universidad-empresa estatal ganadera cubana. Parte I. Estudio de caso: fundamentación y propuesta de un modelo*. [En línea] junio de 2017. [Citado el: 10 de junio de 2019.] http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-03942017000200010. ISSN 2078-8452.

9. **Canut, Carlos Granell. 2006.** *Avances en las infraestructuras de datos espaciales.* s.l. : Publicacions de la Universitat Jaume I, 2006.
10. **Capote, MSc. Jesús Ramón Vasco. 2011.** Una aproximación al concepto de gestión de la información. [En línea] 2011. [Citado el: 12 de 2 de 2019.] <http://www.ilustrados.com/tema/13072/aproximacion-concepto-gestion-informacion.html>.
11. **Carlos A. Guerrero, Johanna M. Suárez y Luz E. Gutiérrez. 2013.** Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. *Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web.* [En línea] 28 de Enero de 2013. [Citado el: 4 de Abril de 2019.] www.researchgate.net.
12. **Cervantes, Humberto. 2018.** *sg.com.mx. sg.com.mx.* [En línea] 7 de mayo de 2018. [Citado el: 4 de Abril de 2019.] <https://sg.com.mx/revista/27/arquitectura-software>.
13. **Chávez, Dayana Leticia López y Ramirez, Pavel Alfonseca. 2018.** Pruebas de seguridad utilizando herramientas para la detección de vulnerabilidades. *Informática Habana.* [En línea] 19 de marzo de 2018. [Citado el: 25 de mayo de 2019.] informaticahabana.cu.
14. **Despliegue, Diagrama de. 2019.** Diagrama de Despliegue UML 2. [En línea] 2019. [Citado el: 1 de mayo de 2019.] http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.
15. **Django. 2019.** Documentación de Django. *Documentación de Django.* [En línea] Django, 2019. [Citado el: 10 de junio de 2019.] <https://docs.djangoproject.com/es/>.
16. **FUENTES, MARGY LILIANA JAIMES, y otros. 2011.** 26, Colombia : Gerencia Tecnológica Informática, 25 de abril de 2011, Vol. 10. ISBN.
17. **Gómez, Ezequiel, Sartorio, Alejandro y Vaquero, Marcelo. 2016.** EasyCard.js Framework. *Tesis Ingeniería en Sistemas Informáticos.* Santa Fe : Universidad Abierta Interamericana, 2016.
18. **GRADOS, Julio. 2016.** JavaScript convive con HTML. [En línea] 2016. [Citado el: 30 de abril de 2019.] <https://devcode.la/blog/que-es-javascript/>.

19. **Hamlett, Kenneth. 2019.** La Voz de Houston. [En línea] 2019. [Citado el: 22 de abril de 2019.] <https://pyme.lavoztx.com/las-caractersticas-de-un-sistema-de-gestin-de-informacin-9451.html>.
20. **Hidalgo, Chistrían. 2015.** *Configuración de Servicios de Red y Portal Web para la Preinscripción a Cursos*. s.l. : El Oro: Unidad Académica de Ingeniería Civil, 2015.
21. **Huelva, Departamento de Informática de la Delegación de Medio Ambiente de. 2006.** PROYECTO TAU. [En línea] 2006. [Citado el: 2 de mayo de 2019.] <http://tauproject.sourceforge.net/>.
22. **Ingram, David. 2019.** La Voz de Houston. [En línea] 2019. [Citado el: 22 de abril de 2019.] <https://pyme.lavoztx.com/qu-es-un-sistema-de-gestin-de-la-informacin-7690.html>.
23. **JETBRAIN. 2017.** Enjoy Productive PHP. *Enjoy Productive PHP*. [En línea] JetBrains s.r.o, 2017. [Citado el: 10 de junio de 2019.] www.jetbrains.com.
24. **JetBrains. 2017.** Descargar PhpStorm Full IDE de programación web. [En línea] 2017. [Citado el: 30 de abril de 2019.] <https://www.jetbrains.com/phpstorm/>.
25. **Larman, Craig. 2006.** jorgesaavedra.wordpress.com. [En línea] 17 de 8 de 2006. [Citado el: 16 de 2 de 2019.] <https://jorgesaavedra.wordpress.com/2007/05/08/patrones-grasp-patrones-de-software-para-la-asignacion-general-de-responsabilidadparte-ii/>.
26. **Ledo, María Josefina Vidal y Pérez, Ana Bárbara Araña. 2012.** Educación Médica Superior. *Educación Médica Superior*. [En línea] 15 de marzo de 2012. [Citado el: 22 de abril de 2019.] http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-21412012000300013.
27. **Martínez, Alejandro y Martínez, Raúl. 2014.** Guía a Rational Unified Process [En línea]. [En línea] 2014. [Citado el: 25 de abril de 2019.] https://www.researchgate.net/publication/268005509_Guia_a_Rational_Unified_Process.
28. **Oracle. 2017.** Getting Started with MySQL. [En línea] 2017. [Citado el: 30 de abril de 2019.] <https://dev.mysql.com/doc/mysql-getting-started/en/>.
29. **Pacheco, Nacho. 2013.** Bases de datos y Doctrine . [En línea] 2013. [Citado el: 30 de abril de 2019.] <http://gitnacho.github.io/symfony-docs-es/book/doctrine.html>.

30. **PHP. 2017.** ¿Qué es PHP? [En línea] 2017. [Citado el: 30 de abril de 2019.]
<http://php.net/manual/es/intro-what-is.php>.
31. **Porto, Julián Pérez, Merino, María. 2012.** Concepto de Gestión. [En línea] 2012.
(<https://definicion.de/gestion/>).
32. **PostgreSQL. 2019.** PostgreSQL: The World's Most Advanced Open Source Relational Database. *PostgreSQL: The World's Most Advanced Open Source Relational Database*. [En línea] 24 de mayo de 2019. [Citado el: 24 de mayo de 2019.] www.postgresql.org.
33. **Pressman, Roger S. 2010.** INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO. *INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO*. s.l. : MC Graw Hill, 2010, pág. 767.
34. —. **2010.** *SOFTWARE ENGINEERING. A PRACTITIONER'S APPROACH*. New York : The McGraw-Hill Companies, Inc., 2010.
35. **Quezada-Torres, Walter David, y otros. 2018.** Scielo. [En línea] 25 de junio de 2018. [Citado el: 25 de abril de 2019.] http://scielo.sld.cu/scielo.php?pid=S1815-59362018000300303&script=sci_arttext&tlng=en.
36. **Ramos, Daniel. 2015.** *Curso de Ingeniería de Software*. s.l. : IT Campus Academy, 2015.
37. —. **2016.** Desarrollo de Software: Estimación Requisitos y Análisis. *Desarrollo de Software: Estimación Requisitos y Análisis*. s.l. : IT Campus Academy, 2016.
38. **Rodríguez, Raidel González, García, Juan Cardentey y García, Xiomara González. 2015.** Scielo. *Scielo*. [En línea] 10 de febrero de 2015. [Citado el: 20 de abril de 2019.] <http://scielo.sld.cu>.
39. **Rodríguez, Tamara. 2015.** Metodología de desarrollo para la actividad productiva de la UCI. Universidad de las Ciencias Informáticas. La Habana. Cuba. 2015.
40. **Rojas López, A., & Avila Aguilera, Y. 2018.** Opuntia Brava. *Opuntia Brava*. [En línea] 21 de enero de 2018. [Citado el: 20 de abril de 2019.] <http://opuntiabrava.ult.edu.cu>.
41. **Rouse, Margaret. 2016.** TechTarget. [En línea] mayo de 2016. [Citado el: 16 de febrero de 2019.] <https://searchdatacenter.techtarget.com/es/definicion/Modelado-de-datos>.

42. **Sánchez, Tamara Rodríguez. 2015.** *Metodología de desarrollo para la Actividad productiva de la UCI.* La Habana : s.n., 2015. PDF Digital.
43. **Sarmiento, Julio. 2016.** UML: Diagrama de despliegue. *UML: Diagrama de despliegue.* [En línea] 2016. [Citado el: 10 de junio de 2019.] umldiagramadespliegue.blogspot.com.
44. **Soa, Mario Valdés, Ávila, Migdely B. Ochoa y Aballe, Yovanni Quevedo. 2007.** Innovación, tecnología y gestión tecnológica. *Innovación, tecnología y gestión tecnológica.* [En línea] 16 de abril de 2007. [Citado el: 10 de junio de 2019.] http://bvs.sld.cu/revistas/aci/vol16_4_07/aci081007.htm.
45. **Sommerville, Ian. 2011.** Ingeniería del Software. *Ingeniería del Software.* s.l. : Addison Wesley, 2011, pág. 667.
46. **Standards, PHP. 2018.** PHP Standards Recommendation. *PHP Standards Recommendation.* [En línea] PHP Framework Interop Group, 2018. [Citado el: 10 de junio de 2019.] www.php-fig.org/psr/.
47. **Symfony. 2017.** What is Symfony? Symfony para programadores . [En línea] 2017. [Citado el: 30 de abril de 2019.] <http://symfony.com/what-is-symfony>.
48. **TECHOPEDIA. 2017.** Modeling Language . *Modeling Language .* [En línea] Techopedia INC, 2017. [Citado el: 10 de junio de 2019.] <http://symfony.com/what-is-symfony> .

ANEXOS

Anexo 1. Historia de Usuario

Tabla 15. HU_8 Adicionar Medio.

Historia de Usuario	
Número: 8	Nombre: Adicionar Medio
Prioridad: Alta	Iteración Asignada: 1
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 2 puntos	Tiempo real: 2 puntos
Descripción: Debe permitir adicionar un medio a la base de datos.	
Riesgo en desarrollo: Alto	

Tabla 16. HU_10 Modificar Medio.

Historia de Usuario	
Número: 10	Nombre: Modificar Medio
Prioridad: Alta	Iteración Asignada: 1
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 2 puntos	Tiempo real: 2 puntos
Descripción: Debe permitir modificar un medio registrado en la base de datos.	
Riesgo en desarrollo: Alto	

Tabla 17. HU_11 Eliminar Medio.

Historia de Usuario	
Número: 11	Nombre: Eliminar Medio
Prioridad: Alta	Iteración Asignada: 1
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 2 puntos	Tiempo real: 2 puntos
Descripción: Debe permitir eliminar un medio registrado en la base de datos.	
Riesgo en desarrollo: Alto	

Tabla 18. HU_12 Adicionar Local.

Historia de Usuario	
Número: 12	Nombre: Adicionar Local
Prioridad: Media	Iteración Asignada: 1
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 1 punto	Tiempo real: 1 punto
Descripción: Debe permitir adicionar un local a la base de datos.	
Riesgo en desarrollo: Medio	

Tabla 19. HU_13 Listar Locales.

Historia de Usuario	
Número: 13	Nombre: Listar Locales
Prioridad: Media	Iteración Asignada: 1
Programador: Leonardo J. Santiesteban Cantallops	

Tiempo estimado: 1 punto	Tiempo real: 1 punto
Descripción: Debe permitir listar los locales registrados en la base de datos.	
Riesgo en desarrollo: Medio	

Tabla 20. HU_14 Modificar Local.

Historia de Usuario	
Número: 14	Nombre: Modificar Local
Prioridad: Media	Iteración Asignada: 1
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 1 punto	Tiempo real: 1 punto
Descripción: Debe permitir modificar un local registrado en la base de datos.	
Riesgo en desarrollo: Medio	

Tabla 21. HU_15 Eliminar Local.

Historia de Usuario	
Número: 15	Nombre: Eliminar Local
Prioridad: Media	Iteración Asignada: 1
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 1 punto	Tiempo real: 1 punto
Descripción: Debe permitir eliminar un local registrado en la base de datos.	
Riesgo en desarrollo: Medio	

Tabla 22. HU_16 Adicionar Área.

Historia de Usuario	
Número: 16	Nombre: Adicionar Área
Prioridad: Media	Iteración Asignada: 1
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 1 punto	Tiempo real: 1 punto
Descripción: Debe permitir adicionar un área a la base de datos.	
Riesgo en desarrollo: Medio	

Tabla 23. HU_17 Listar Áreas.

Historia de Usuario	
Número: 17	Nombre: Listar Áreas
Prioridad: Media	Iteración Asignada: 1
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 1 punto	Tiempo real: 1 punto
Descripción: Debe permitir listar las áreas registradas en la base de datos.	
Riesgo en desarrollo: Medio	

Tabla 24. HU_18 Modificar Área.

Historia de Usuario	
Número: 18	Nombre: Modificar Área
Prioridad: Media	Iteración Asignada: 1
Programador: Leonardo J. Santiesteban Cantallops	

Tiempo estimado: 1 punto	Tiempo real: 1 punto
Descripción: Debe permitir modificar un área registrada en la base de datos.	
Riesgo en desarrollo: Medio	

Tabla 25. HU_19 Eliminar Área.

Historia de Usuario	
Número: 19	Nombre: Eliminar Área
Prioridad: Media	Iteración Asignada: 1
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 1 punto	Tiempo real: 1 punto
Descripción: Debe permitir eliminar un área registrada en la base de datos.	
Riesgo en desarrollo: Medio	

Tabla 26. HU_1 Adicionar usuario.

Historia de Usuario	
Número: 1	Nombre: Adicionar usuario.
Prioridad: Baja	Iteración Asignada: 2
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 4 puntos	Tiempo real: 4 puntos
Descripción: Debe permitir adicionar un usuario a la base de datos.	
Riesgo en desarrollo: Bajo	

Tabla 27. HU_2 Listar usuarios.

Historia de Usuario	
Número: 2	Nombre: Listar usuarios.
Prioridad: Baja	Iteración Asignada: 2
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 4 puntos	Tiempo real: 4 puntos
Descripción: Debe permitir listar los usuarios que se encuentran registrados en la base de datos.	
Riesgo en desarrollo: Bajo	

Tabla 28. HU_3 Modificar usuario.

Historia de Usuario	
Número: 3	Nombre: Modificar usuario.
Prioridad: Baja	Iteración Asignada: 2
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 4 puntos	Tiempo real: 4 puntos
Descripción: Debe permitir modificar los usuarios que se encuentran registrados en la base de datos.	
Riesgo en desarrollo: Bajo	

Tabla 29. HU_4 Eliminar usuario.

Historia de Usuario	
Número: 4	Nombre: Eliminar usuario.
Prioridad: Baja	Iteración Asignada: 2
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 4 puntos	Tiempo real: 4 puntos
Descripción: Debe permitir eliminar los usuarios que se encuentran registrados en la base de datos.	
Riesgo en desarrollo: Bajo	

Tabla 30. HU_5 Asignar roles.

Historia de Usuario	
Número: 5	Nombre: Asignar roles.
Prioridad: Media	Iteración Asignada: 2
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 4 puntos	Tiempo real: 4 puntos
Descripción: Debe permitir asignar roles a los usuarios que se encuentran registrados en la base de datos.	
Riesgo en desarrollo: Alto	

Tabla 31. HU_6 Asignar permisos.

Historia de Usuario	
Número: 6	Nombre: Asignar permisos.
Prioridad: Media	Iteración Asignada: 2
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 4 puntos	Tiempo real: 4 puntos
Descripción: Debe permitir asignar permisos a los usuarios que se encuentran registrados en la base de datos.	
Riesgo en desarrollo: Alto	

Tabla 32. HU_7 Autenticar usuario.

Historia de Usuario	
Número: 7	Nombre: Autenticar usuario.
Prioridad: Media	Iteración Asignada: 2
Programador: Leonardo J. Santiesteban Cantallops	
Tiempo estimado: 4 puntos	Tiempo real: 4 puntos
Descripción: Debe permitir que se verifiquen los datos de los usuarios que posee la base de datos.	
Riesgo en desarrollo: Medio	

Anexo 2. Diagramas de clases de diseño

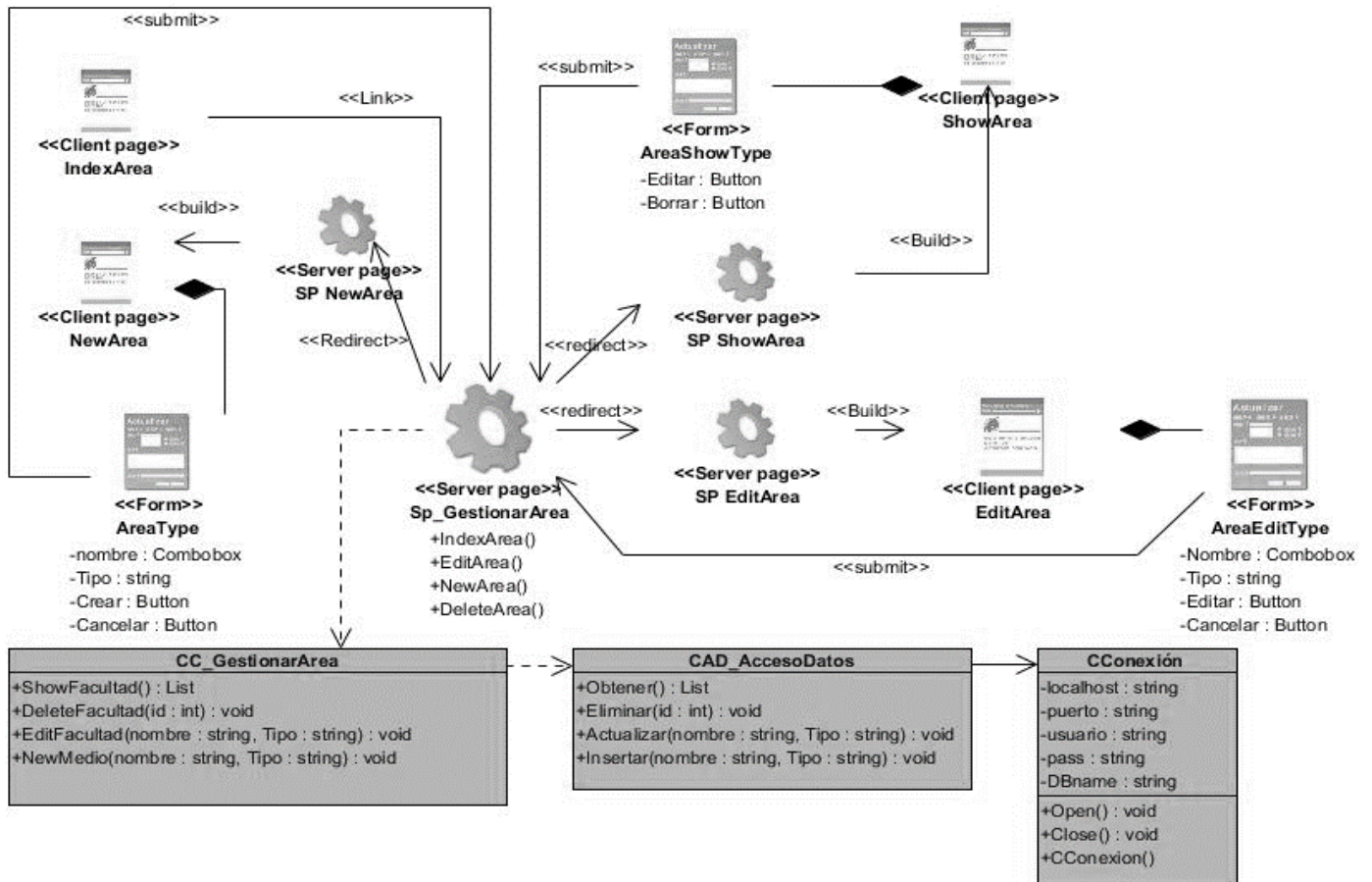


Figura 8: Diagrama de clase de diseño "Gestionar Área".

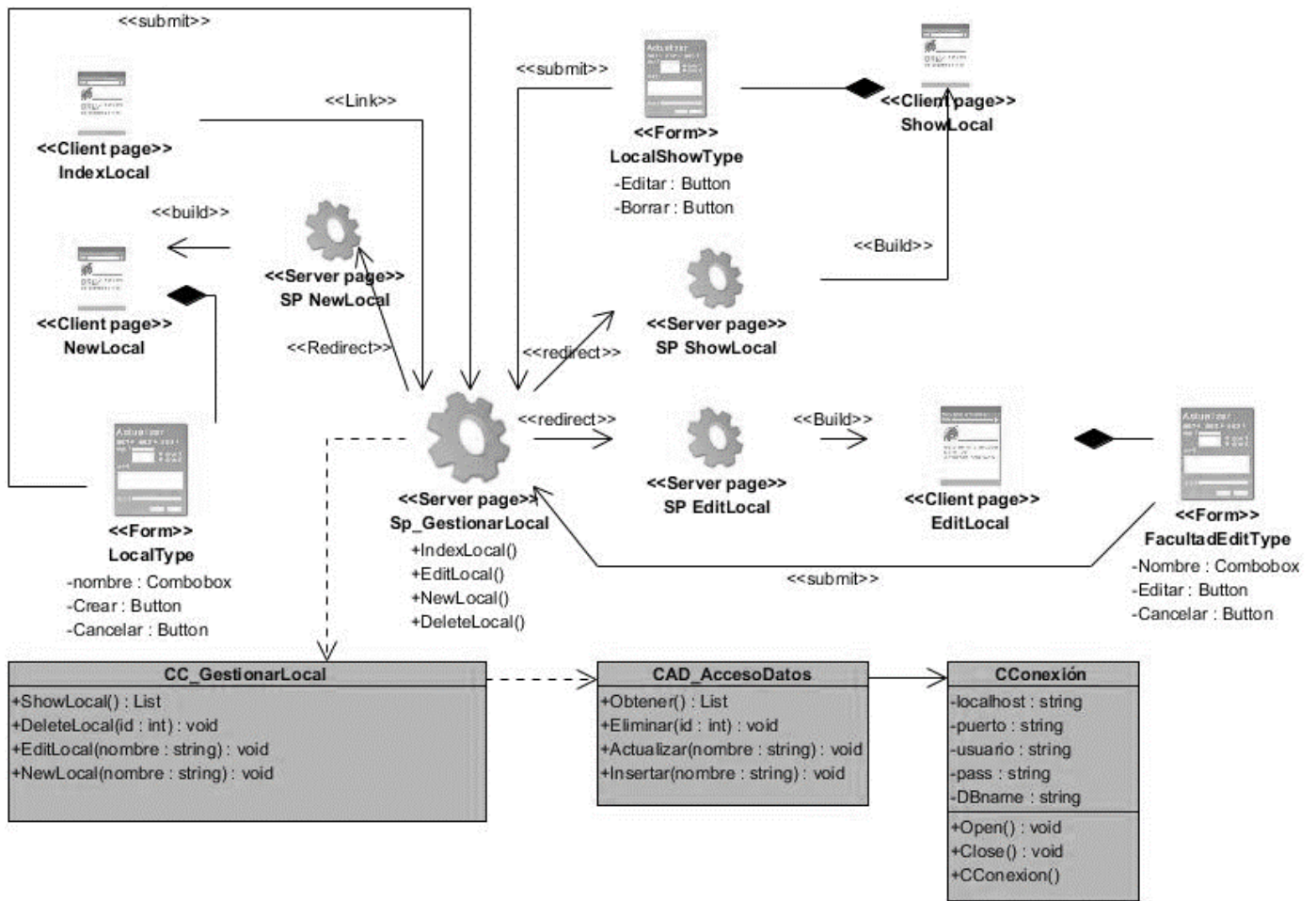


Figura 9: Diagrama de clase de diseño "Gestionar Local".

Anexo 3: Entrevista

- 1- ¿Se resolverán los problemas de gestión de tecnologías del centro CIDI con la implementación de una aplicación informática?

Respuesta: Sí se resolverían pues permitiría tener un control total de donde están ubicados los mismo, saber de cualquier movimiento y tener un informe del estado actual de cada medio.

- 2- ¿Los procesos de gestión de tecnología se realizan de forma manual o están automatizados?

Respuesta: Estos procesos actualmente se realizan de forma manual sujetos a errores.

- 3- ¿Necesita la tecnología ser desarrollada con alguna tecnología específica?

Respuesta: Sí

- 4- ¿Qué problemas se presentan a la hora de realizar estos procesos de forma manual?

Respuesta: El formato duro no permite la edición de los contenidos a no ser que se vuelva a realizar este proceso, al igual que se tienen grandes riesgos de perder, duplicar o cometer errores con la información recolectada.