



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 1

SISTEMA DE GESTIÓN PARA LA
LENCERÍA DE LA RESIDENCIA UCI

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

Autor: Ilianys Alfonso Milián

Tutores:

Ing. Javier Anias Santos

Ing. Andy Vega Fajardo

La Habana, junio de 2020

“Año 62 de la Revolución”

Declaración de autoría

Declaro por este medio que yo **Ilianys Alfonso Milian**, con carné de identidad **96080204737** soy el autor principal del trabajo titulado “**Sistema de gestión para la lencería de la residencia de la Universidad de las Ciencias informáticas**” 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.

Para que así conste firmamos la presente a los _____ días del mes de _____ de _____

Firma del autor:

Ilianys Alfonso Milian

Firma del tutor

Ing. Javier Anias Santos

Firma del tutor

Ing. Andy vega Fajardo

Resumen

La Universidad de las Ciencias Informáticas (UCI) tiene entre sus objetivos la informatización de todas sus áreas, con el propósito de mejorar el funcionamiento de los procesos que tienen lugar en el centro de altos estudios. El presente trabajo de diploma está dirigido al proceso de gestión de la lencería en la residencia UCI. Para ello se realiza el análisis de una aplicación web que automatice las actividades vinculadas a este proceso. Se ha realizado un estudio profundo de las técnicas de captura y validación de requisitos, así como de las tendencias en el modelado de software y un análisis exhaustivo del negocio y funcionamiento del proceso en la residencia. Para dar cumplimiento con calidad a este proceso, se ha utilizado la metodología de desarrollo de software AUP en su variación para la UCI, usando UML como lenguaje de modelado y Visual Paradigm como herramienta CASE que permitirá la modelación gráfica del sistema. Utilizando toda esta tecnología y siguiendo los pasos requeridos se ha especificado y documentado el negocio con todas sus actividades. De igual manera se ha llevado a cabo un levantamiento de requisitos tanto funcionales como no funcionales, los cuales definen las capacidades que el sistema debe cumplir, más las cualidades que el producto debe tener, cumpliendo con las necesidades reales del cliente. Y a partir del refinamiento de dichos requisitos se ha obtenido la modelación y especificación del sistema propuesto.

Palabras clave: lencería, sistema, gestión, residencia, lavandería.

Índice

Introducción5

Capítulo 1. Fundamentación teórica del Sistema de gestión para la lencería de la residencia UCI.13

Introducción13

1.1 Definición del marco teórico conceptual de la investigación13

1.2 Estudio de los sistemas homólogos14

1.2.1 Usados en el mundo14

1.2.2 Usados en Cuba15

Tabla 1. Sistemas de gestión de inventarios. Fuente: Elaboración propia.19

1.3. Selección del entorno de desarrollo para la construcción de la solución19

1.3.1. Metodología de desarrollo de software19

1.4 Lenguajes para la programación21

1.4.1 Del lado del servidor21

1.4.2 Del lado del cliente22

1.4.3 *Framework*. Marco de trabajo23

1.5 Herramientas25

Conclusiones parciales del capítulo27

Capítulo 2. Análisis y diseño del Sistema de gestión para la lencería de la residencia UCI28

Introducción28

2.1 Descripción de la propuesta de solución28

2.2 Especificación de requisitos de software29

2.2.1 Requisitos funcionales30

2.2.2 Requisitos no funcionales33

2.3 Historias de usuarios35

2.4 Patrones de diseño45

2.4.1 Patrones GRASP45

2.4.2 Patrones GoF	46
2.5 Patrones arquitectónicos	46
2.5.1 Patrón arquitectónico Modelo – Vista – Controlador	47
2.6 Modelo de diseño	48
2.6.1 Diagrama de clases de diseño	48
2.7 Modelo de datos	50
2.8 Modelo de despliegue	51
Conclusiones parciales del capítulo	52
Capítulo 3. Implementación y validación del Sistema de gestión para la lencería de la residencia UCI	53
Introducción	53
3.1.1 Diagramas de componentes	53
3.2 Estándares de codificación	55
3.3.1 Pruebas funcionales	56
3.3.2 Pruebas de carga y estrés (Pendiente)	63
3.3.3 Pruebas de seguridad	64
3.3.4 Pruebas de usabilidad	64
CONCLUSIONES	68

Introducción

Los avances tecnológicos de la actualidad evidencian un desarrollo acelerado de las Tecnologías de la Información y la Comunicaciones (TIC), las cuales están cada vez más ligadas a las distintas esferas de la vida social. Múltiples sectores se han visto favorecidos por el avance de las denominadas TIC, dentro de los que se destacan, la educación, la salud, el deporte, la producción y los servicios. Su utilización supone importantes logros en múltiples esferas de la sociedad como la economía, la educación, la salud y en el

desarrollo social, proporcionando recursos y ahorro de tiempo, al simplificar y agilizar los procesos de gestión, así como la toma de decisiones. Cada día el uso de las computadoras y medios tecnológicos se hace más necesario. El sistema de educación cubano aprovecha las facilidades que ofrecen las mismas, aspirando informatizar la mayor cantidad de áreas posible y de este modo mejorar la calidad y el control del trabajo mediante el desarrollo de software y herramientas tecnológicas.

Cuba a pesar de ser un país bloqueado, trabaja actualmente en pos de lograr una digitalización de la mayor cantidad de información y procesos, aplicándose en diferentes esferas de la sociedad, con este propósito surge la Universidad de las Ciencias Informáticas (UCI) en el año 2002, con el fin de llevar la informatización a diferentes regiones del país a partir del desarrollo de software y sistemas propios. La creación de la Universidad dio la oportunidad de que en sus aulas pudiese haber egresados de todo el país, con la peculiaridad del régimen de internado donde una parte de los trabajadores se mantenían en igualdad de condiciones. Como institución universitaria de nuevo tipo, en ella se plasmarían los nuevos conceptos, métodos y medios audiovisuales en función de la formación de profesionales integrales.

Actualmente, en el Sistema de Educación Superior Cubano se concibe y potencia la función de la extensión universitaria como una de las principales vertientes de trabajo, apreciándose como el elemento más dinámico e integrador del vínculo universidad-sociedad. Los documentos aprobados y la práctica de la vinculación universidad-sociedad, demandan el incremento de las investigaciones sociales y su creciente articulación a la toma de decisiones, lo cual constituye un reto.

De manera particular, la UCI ha atendido y perfeccionado los procesos que integran y garantizan su funcionalidad y eficiencia, resolviendo gradualmente sus carencias en las diferentes áreas a partir del desarrollo de sistemas, entre estos, los que hacen posible el trabajo con la información, su organización, control y procesamiento. Para el surgimiento de la Residencia de la UCI, constituyó una premisa, que el

total de los estudiantes fueran becarios, por lo que hubo que preparar condiciones óptimas de hospedaje, tanto para ellos como, para los trabajadores albergados.

La Residencia ocupa un lugar importante dentro del área de la Universidad, sus edificios y apartamentos acogen a la totalidad de los estudiantes y una gran parte de sus trabajadores. Siguiendo las líneas de trabajo del centro y sus principales misiones. Esta cuenta con una capacidad máxima de 6000 estudiantes y 2000 trabajadores residentes, con aproximadamente 137 edificios y una suma cercana a los 1548 apartamentos en la actualidad. Debido a la gran capacidad de alojamiento con que se cuenta es que surge la vicerrectoría de residencia que cuenta con un vicerrector y cuatro direcciones, una para la Dirección de Residencia, dos de estas para la beca de estudiantes y la otra para los trabajadores.

La Dirección de Residencia de la UCI es una de las áreas que han utilizado sistemas para informatizar sus procesos y así agilizarlos, entre los que se encuentran, el sistema de la guardia obrera-estudiantil y el sistema de incidencias. La gran mayoría de los procesos que realiza la Dirección de la Residencia no han sido informatizados, lo que hace muy engorroso el trabajo debido que, la mayoría del trabajo con la información es en formato duro, se necesita de personal para el movimiento de la información a las diferentes áreas, además de la perdida y deterioro de la misma puede presentar problemas para realizar su control.

Desde sus inicios la dirección de la residencia de la UCI ha trazado estrategias para establecer las normas respecto a la entrega, control, tratamiento y destino final de la ropa de cama y otros artículos de lencería a estudiantes, profesores y trabajadores internos. La solicitud de la ropa de cama y artículos de lencería la realiza el Director de la Residencia de estudiantes de primer año, sobre la base de las necesidades (teniendo en cuenta matrícula de estudiantes en cada inicio del curso) y esta es aprobada por el Vicerrector de Residencia.

El Director registra nombre y apellidos del personal que será encargado de gestionar estos artículos y comprobar su estado cualitativo y cuantitativo, además de registrarlos debidamente en el libro de cuantía de la ropa de cama diferenciando cada artículo. Se debe cumplir con el tratamiento indicado en la legislación vigente relacionado con el asunto en caso de deterioro, hurto, pérdida o robo, haciendo constancia a través del Acta de responsabilidad Material firmada por las partes implicadas, definiendo grado de responsabilidad y gravedad de los hechos. Se entrega, por parte de los recepcionistas de lavandería y Técnico General, a los estudiantes del año en el acto de incorporación a la Residencia, la ropa de cama y otras lencerías en las cantidades asignadas y la calidad requerida, quedando en su posesión y custodia bajo acta de recepción oficial firmada debidamente, llamada en el área Modelo de Avituallamiento.

En el caso de la lencería que poseen los becados de la residencia de profesores y especialistas, debido a la acumulación histórica de avituallamiento en boletas activas desde el surgimiento de la Universidad de las Ciencias Informáticas en el año 2002; el lavado realizado por la empresa de Guanabacoa con pro quimia de la propia entidad cuyo servicio no ha sido el mejor por lo cual la calidad del lavado ha disminuido; así como la acumulación en el submayor, se determina sea elaborado un expediente por deterioro que permita viabilizar el proceso con aquella lencería en boletas desde el año 2002 hasta el año actual, cuadro resumen de artículos por año. Serán anexadas a dicho expediente, las boletas de avituallamiento de los residentes de la residencia de profesores que poseen lencería desde el año 2002 hasta el año 2012. Cualquier situación que atente contra el cumplimiento de este procedimiento se debe comunicar al Director de Residencia, Vicerrector de Extensión Universitaria y Residencia o miembros de la comisión de control interno, cualquier situación que atente contra el cumplimiento de este procedimiento. El control de estos medios es llevado a cabo por cada uno de los técnicos generales en varios archivos excel y documentación física. Sin embargo, no existe uniformidad en el manejo de esta información.

Luego de realizar un análisis sobre el proceso de gestión de la lencería en la residencia UCI, se detectaron un conjunto de problemas y necesidades. Entre las necesidades identificadas están, el control de los artículos de lencería, este es un proceso ejecutado por varias personas que intervienen en sus diferentes niveles de aprobación y verificación. Estas personas se encuentran en diferentes áreas dispersas físicamente en la Universidad lo que dificulta y demora la agilidad del proceso. Almacenamiento de la información de forma física, los diferentes anexos y actas, generadas en los procesos están en riesgo de sufrir deterioro debido al largo período que se han encontrado guardados o el tiempo de trabajo para los que han sido utilizados. El flujo de la información es deficiente debido a la cantidad de artefactos que se generan de las diferentes áreas y el tiempo que se demora en su traslado, lo que entorpece el trabajo de algunas áreas.

Tomando en consideración la problemática anteriormente descrita, se plantea como **problema de investigación**: ¿Cómo contribuir a la gestión eficiente de la lencería de la Residencia de la Universidad de las Ciencias Informáticas?

Se definió como **objeto de estudio** los procesos de gestión de información llevados a cabo en la residencia universitaria haciendo énfasis en el proceso de gestión de lencería en la residencia de la Universidad de las Ciencias Informáticas, como **campo de acción**.

Con el fin de darle una solución al problema planteado se define como **objetivo general**: Desarrollar un sistema de gestión basado en tecnologías de desarrollo libre para contribuir a los procesos de lencería de la residencia UCI.

En virtud de solucionar la problemática existente, el objetivo general ha sido desglosado en los siguientes **objetivos específicos**:

- 1) Construir el marco teórico conceptual.
- 2) Realizar el estado del arte respecto al análisis del proceso de Tratamiento de lencería.

3) Desarrollar un sistema que permita la gestión de la entrega, control, tratamiento y destino final de la ropa de cama y otros artículos de lencería a estudiantes, profesores y trabajadores internos.

4) Validar el correcto funcionamiento del sistema desarrollado.

A continuación, se definen las siguientes **preguntas científicas**:

¿Cómo informatizar los procedimientos establecidos respecto a la entrega, control, tratamiento y destino final de la ropa de cama y otros artículos de lencería a estudiantes, profesores y trabajadores internos?

¿Cuáles son las características que debe cumplir el sistema de gestión para la lencería de la residencia UCI?

¿Qué elementos tener en cuenta durante la etapa de implementación del sistema de gestión para la lencería de la residencia UCI?

¿Cómo comprobar que el sistema desarrollado permite a los usuarios desde sus dispositivos móviles la gestión de la lencería de la residencia UCI?

Para lograr los objetivos descritos anteriormente se plantean las siguientes **tareas de investigación**:

1) Realización de consultas bibliográficas que permitan elaborar el marco teórico conceptual.

2) Selección de las técnicas de captura de requisitos que más se adapten al cliente con el que se trabaja, para la correcta identificación de las necesidades del proceso en la obtención de requisitos funcionales y no funcionales.

3) Identificación de los artefactos que se utilizarán para documentar a partir del estudio de la fase de captura de requisitos y análisis de la metodología de desarrollo seleccionada.

4) Desarrollo de un sistema que permita la gestión de la entrega, control, tratamiento y destino final de la ropa de cama y otros artículos de lencería a estudiantes, profesores y trabajadores internos.

5) Validación del correcto funcionamiento del sistema desarrollado.

Métodos de Investigación utilizados:

Métodos teóricos:

Analítico-Sintético: permitió la recopilación y análisis de la información sobre los elementos esenciales referentes a las teorías, documentos y literatura en general relacionada con los sistemas de gestión.

Histórico-Lógico: permitió conocer los antecedentes e importancia de cómo han evolucionado en el tiempo los sistemas de gestión, así como las herramientas y tecnologías utilizadas en el desarrollo de aplicaciones web.

Métodos empíricos:

Modelación: utilizado en la representación, mediante el uso de diagramas, de las características del sistema, relaciones entre objetos; y las actividades que intervienen en los procesos implementados en el subsistema de gestión de la lencería en la residencia UCI

Observación: Se hace un estudio minucioso de trabajos realizados que se encuentran estrechamente vinculados con la problemática a la que se hace referencia.

Entrevista: Se empleó para obtener información perteneciente al Centro de Ideoinformática (CIDI) mediante un modelo estructurado con preguntas predefinida, dirigidas a individuos con vasta experiencia en los sistemas de gestión.

Este trabajo está formado por tres capítulos:

Capítulo 1. Fundamentación teórica del Sistema de gestión para la lencería de la residencia UCI: se presentan un conjunto de conceptos asociados al objeto de estudio de la investigación. Además, se realiza un análisis del estado del arte acerca del proceso relacionado con el tratamiento de lencería.

Capítulo 2. Análisis y diseño del Sistema de gestión para la lencería de la residencia UCI: se

exponen las características del sistema, patrones de diseño, requisitos funcionales y no funcionales, arquitectura utilizada y los artefactos que requiere la metodología de desarrollo utilizada.

Capítulo 3. Implementación y validación del Sistema de gestión para la lencería de la residencia

UCI: se presentan algunos aspectos asociados a la implementación de la solución informática, así como sus componentes. Además, se exhiben los diseños de casos de prueba a utilizar en la validación del sistema y se analizan los resultados arrojados por las pruebas realizadas.

Capítulo 1. Fundamentación teórica del Sistema de gestión para la lencería de la residencia UCI.

Introducción

En el presente capítulo se abordan los principales elementos relacionados con el control de la entrega y recepción de la lencería de la residencia UCI. Además, se presentarán las tecnologías, herramientas y metodología que serán empleadas en la implementación de la propuesta de solución.

1.1 Definición del marco teórico conceptual de la investigación

Para lograr un mejor entendimiento del entorno de la investigación es necesario la identificación y el estudio de las principales bibliografías y así analizar algunos de los conceptos asociados al área de los sistemas de recuperación de información y su personalización.

Gestión: Acción y efecto de administrar; planificación, organización, dirección y control de los recursos de una organización. Entre los diferentes tipos de gestión se encuentran: la ambiental, la tecnológica, la pública, de proyecto, de conocimiento, de la información, etc. La gestión tiene el fin de obtener el máximo beneficio posible; este beneficio puede ser económico o social, dependiendo esto de los fines que persiga la organización.

Sistema de gestión: Un sistema de gestión permite optimizar los recursos, reducir costes y mejorar la productividad en la empresa. Este instrumento de gestión reportará datos en tiempo real que permitirán tomar decisiones para corregir fallos y prevenir la aparición de gastos innecesarios. Estos sistemas están basados en normas internacionales que permiten controlar distintas facetas en una empresa, como la calidad de su producto o servicio, los impactos ambientales que pueda ocasionar, la seguridad y salud de los trabajadores, la responsabilidad social o la innovación [1].

1.2 Estudio de los sistemas homólogos

1.2.1 Usados en el mundo

EZOfficeInventory: Es una solución para la gestión de inventarios basado en la nube, diseñada para todo tipo de empresas. Ofrece escaneo de código de barras, gestión de proveedores, gestión de compras y seguimiento de inventarios. La función de gestión de inventarios del software permite al usuario dar seguimiento de los niveles de inventario usando códigos QR 2 y etiquetas de código de barras. Este sistema cuenta con una aplicación para móviles con funcionalidad de escaneo de código de barras y código QR, lo cual permite a los usuarios registrar el inventario usando su *tablet* o *smartphone*. Una característica clave es su capacidad de seguimiento vía GPS 3, lo cual permite al usuario la localización de todo ítem. Esta funcionalidad ayuda a dar seguimiento a todos sus activos en un mapa interactivo para identificar con facilidad cualquier artículo extraviado. A pesar de todas las cualidades que posee la herramienta, se debe destacar que no está desarrollada sobre una plataforma libre y necesita estar conectada a Internet para su correcto funcionamiento.

Inventoria Inventory Software: Está diseñada para ser tan intuitiva como sea posible, por lo que después de una instalación rápida, optimizará sus procesos de inventario en cuestión de minutos. Ofrece numerosas características útiles como la capacidad de añadir notas, URL 4 y fotografías para describir cada artículo en el inventario, escanea códigos de barras para agregar nuevos elementos. Además, se integra con otros programas empresariales para mantener datos de inventario en todos los aspectos de su negocio y facilita la transferencia de existencias entre distintas ubicaciones. También, hace posible ver los niveles de inventario por ubicación, categoría o general y almacena en el historial del producto cuando se recibieron o se vendieron artículos. Asigna niveles de acceso adecuados a los diferentes usuarios y permite a varios usuarios acceder al programa de administración de inventario. Posibilita el acceso al programa a través de la interfaz web desde dispositivos móviles. La versatilidad de este software para la

administración de inventario hace que el mismo sea una herramienta muy potente pero no se puede dejar de destacar que no es un sistema desarrollado sobre una plataforma libre.

Inventory Tracker Plus: Es un software de gestión de inventario que facilita organizar el inventario y aparejado a esto dar seguimiento a la información de los clientes y vendedores. Utiliza su escáner de código de barras para ingresar números de inventario y agregar elementos de inventario a las facturas. Genera una lista rápida de artículos que necesitan ser reordenados y crea órdenes de compra que se pueden enviar por correo o por correo electrónico para esos artículos de inventario. Este software de inventario crea fácilmente facturas, hojas de embalaje, etiquetas de envío e informes de inventario, mantiene un registro de todos sus clientes, proveedores, vendedores y otros contactos, incluyendo direcciones de facturación y envío e historial de pedidos. Organiza y crea listas de tareas y citas en formatos de calendario diarios, semanales o mensuales. Este software de seguimiento de inventario maneja fácilmente devoluciones, pérdidas en el inventario, ajustes de auditoría, órdenes atrasadas, órdenes especiales y cantidades ilimitadas. Permite añadir un número ilimitado de fotos de sus artículos y ver un historial completo de artículos recibidos y vendidos. Además, puede traducir a cualquier idioma las facturas y los informes de inventario, incluye ayuda integrada, una guía del usuario y una visita guiada completa. Este software es idóneo para el seguimiento de activos por lo que constituye una buena opción para personas que están buscando un enfoque sencillo a la administración de inventario. A pesar de lo antes mencionado se debe enfatizar que no está desarrollado en una plataforma libre y tiene un costo de ciento noventa y nueve dólares (en el momento en que fue analizado).

1.2.2 Usados en Cuba

Versat Sarasola: Constituye el primer sistema integral de gestión de contabilidad certificado. Desarrollado para la gestión económica eficaz y fiable [2]. Este ofrece al usuario la posibilidad de contar con un instrumento seguro, rápido, eficaz y de fácil manejo para la planificación, control y el análisis de la gestión

económica. Ha sido diseñado para ser utilizado en cualquier entidad y permite llevar el control y el registro contable individual de todos los hechos económicos que se originan en las estructuras internas de las mismas y obtener los estados financieros y análisis económicos y financieros en estos niveles. Actualmente es utilizado en Cuba en más de 8 000 entidades de 29 organismos. Este sistema integrado cuenta con un conjunto de 12 módulos entre los que se encuentran control de activos fijos y control de inventarios. En el módulo de Control de Inventarios se definen formatos del clasificador de productos para lograr una uniformidad en el registro y la agregación de información en los reportes de salida, se conceptualizan los movimientos para lograr una información amplia sobre los orígenes y destinos de los recursos. Permite el control de las existencias y movimientos en diferentes monedas. Muestra el cuadro diario de cada uno de los almacenes por las diferentes cuentas. Ofrece la posibilidad de duplicar documentos para agilizar los pases de los mismos y lograr que un mismo documento se convierta en otro con solo adicionar un mínimo de información, realiza un control de las existencias y movimientos por custodios y se emiten diferentes reportes e información de utilidad para la correcta administración de los recursos materiales [2]. El módulo de Control de Activos Fijos recoge las operaciones normales que en esta actividad se realizan (altas, bajas, modificaciones de los medios). Posee un asistente para la configuración del subsistema, la tarea más compleja de este módulo [2]. El sistema Versat Sarasola es una aplicación de escritorio implementado en lenguaje de programación Delphi. Trabaja solamente sobre el sistema operativo Windows y tiene soporte para base de datos SQL 5 Server 2000. Además, el software tiene incluidos otros módulos que no son necesarios en la propuesta de solución lo que entorpece el proceso de gestión de los medios y hace más lenta la capacitación y ejecución del software.

SISCONT5: Es un sistema que se aviene a las definiciones y conceptos del Ministerio de la Industria Básica, aunque por las acciones contables financieras que permite puede ser utilizado en otras entidades nacionales [2]. Está formado por varios módulos dentro de los que se encuentra inventarios y activos fijos tangibles. Puede ser explotado en régimen mono usuario y multiusuario. Se define para mono entidad y

multientidad, en esta última existe el control de su acceso para las entidades en un mismo equipo de cómputo como servidor.

El módulo de Inventario maneja toda la información referida al Submayor de Inventarios de la entidad, garantizando el cuadro permanente con las respectivas cuentas de la Contabilidad General. El sistema está preparado para controlar el saldo de cada material en dos monedas, a partir de los procedimientos vigentes en el país en cuanto a política monetaria [2].

Permite el tratamiento de las contabilizaciones de forma transaccional, por resúmenes diarios o de forma personalizada según se defina por parámetros. Incorpora tratamiento de lotes a los productos y dos tipos de valoración de las cuentas FIFO 6 y Promedio Ponderado. Este software no está implementado sobre herramientas libres y es una aplicación de escritorio lo ocasiona que se deba instalar la aplicación en cada estación de trabajo.

Rodas XXI: Sistema multiempresa y multiusuario creado por CITMATEL 7 para la automatización de la gestión empresarial [2]. Contiene diferentes módulos que pueden usarse integrados o independientes dentro de los que se encuentran activos fijos tangibles e inventarios.

El módulo de activos fijos tangibles de RODAS XXI permite tener un control detallado de los activos fijos de su entidad, realizando en el mismo momento que se registra un movimiento, su contabilización. Se pueden realizar todo tipo de operaciones de activos fijos con facilidad en el momento que se desee, generando el documento asociado al movimiento de que se trate de forma automática previa configuración del sistema para ello. Permite el control por separado de los activos fijos que se encuentran en almacén de los que se encuentran en explotación. Es posible además, realizar ajustes a los activos fijos con facilidad mediante una opción que brinda el sistema [2]. Este sistema no está desarrollado sobre tecnologías libres y es una aplicación de escritorio lo que ocasiona que el usuario deba instalar la aplicación en cada estación de trabajo.

Sistema de Gestión Integral (Assets): Es un sistema multiusuario que se monta en una plataforma de servidores SQL, dividido en módulos económicos que trabajan en conjunto para el control de las actividades económica, financiera y contable sobre los medios materiales y financieros. El módulo inventario del Assets divide los medios en dos tipos de cuentas: activos fijos y útiles herramientas. Estas últimas son las que se utilizan para realizar las actividades de mantenimiento, talleres, almacenes, así como los equipos de protección física. Comprende entre otros, herramientas manuales, artículos de protección personal, utensilios de laboratorios, mini calculadoras y utensilios menores de cocina. Es un sistema integral modular concebido para el control de la actividad económica empresarial. Se emplea en la UCI para calcular las nóminas y controlar los recursos laborales de que dispone esta entidad. Este sistema está desarrollado sobre herramientas propietarias y constituye una aplicación de escritorio lo que ocasiona que el usuario deba instalar la aplicación en cada estación de trabajo.

Valoración de los sistemas homólogos

Una vez analizados los sistemas homólogos se concluye que no resultan soluciones factibles para gestionar la lencería de la residencia de la UCI debido a que fueron desarrollados sobre plataformas de software propietario, lo que implica incrementos de gastos en licencias de uso y mantenimiento del software. Además, los sistemas internacionales utilizan tecnologías que no son asequibles para Cuba y algunos de ellos requieren de la conexión a internet para su funcionamiento. Los sistemas nacionales constituyen aplicaciones de escritorio lo que ocasiona que el usuario deba instalar la aplicación en cada estación de trabajo e impide que la aplicación pueda ser ejecutada desde dispositivos móviles. Es importante destacar que el estudio de estos sistemas evidenció la necesidad de desarrollar la propuesta de solución ya que ninguna de las soluciones estudiadas resuelve del todo la problemática existente y ayudó a una mejor identificación de los requisitos funcionales de la solución que se propone. En la Tabla 1 se muestra la comparación realizada entre estos sistemas estudiados:

Tabla 1. Sistemas de gestión de inventarios. Fuente: Elaboración propia.

Sistema	Software libre	Sistema web	Dependencia de Internet
EZOfficeInventory	No	Si	Si
Inventoria Inventory Software	No	Si	No
Inventory Tracker Plus	No	Si	Si
Versat Sarasola	No	No	No
SISCONT5	No	No	No
Rodas XXI	No	No	No
ASSETS	No	No	No

1.3. Selección del entorno de desarrollo para la construcción de la solución

Para el desarrollo de la propuesta de solución se hace necesario el estudio de las herramientas y tecnologías que se emplearán en correspondencia con la metodología usada en el desarrollo del sistema del Sistema de gestión para la lencería.

1.3.1. Metodología de desarrollo de software

Una metodología de desarrollo de software impone un proceso de forma disciplinada sobre el desarrollo de software con el objetivo de hacerlo más predecible y eficiente [3]. Está orientada a la forma adecuada de construcción del software, que puede incluir modelos de sistemas, notaciones, reglas, sugerencias de

diseño y guías de procesos [4]. Otro concepto más ampliado y que agrupa a los dos anteriores es el que plantea que una metodología de desarrollo, en ingeniería de software, es un conjunto de herramientas, técnicas, procedimientos y soporte documental encaminados a estructurar, planificar y controlar el proceso de desarrollo de forma organizada y lógica, que tiene como objetivo apoyar a los desarrolladores en la creación de un nuevo software [5].

Para el desarrollo de la solución se usará AUP en variación de la metodología para la universidad (AUP-UCI), por ser la metodología empleada en el centro CIDI para el desarrollo de proyectos.

La metodología AUP-UCI contribuye de manera significativa al desarrollo del software que se espera obtener, ya que esta metodología además de ser ágil es también adaptable a cualquier circunstancia que pueda surgir en el proceso de desarrollo del proyecto. La UCI le ha realizado modificaciones con el fin de adaptarla al ciclo de vida definido para la actividad productiva de dicha institución; de las 4 fases que encierra la metodología AUP se simplificaron a:

Inicio: esta etapa se mantiene de AUP pero se modifica el objetivo de la misma; realización de la planeación del proyecto en cuanto a un estudio de la organización del cliente, el alcance del proyecto, estimaciones de tiempo esfuerzo y costo que serán necesarios para decidir si se desarrolla el producto o no.

Ejecución: en esta fase se recogen las actividades que desarrolla AUP de elaboración, construcción y transición. Se ejecutan las actividades requeridas para obtener el software y ajustar los planes del proyecto, realizando las tareas de modelado del negocio, obtención y análisis de los requisitos, elaboración de la arquitectura, diseño, implementación y liberación de producto.

Cierre: en esta fase se analizan los resultados del proyecto, su ejecución y se llevan a cabo actividades formales de cierre del proyecto.

1.4 Lenguajes para la programación

Los lenguajes de desarrollo intentan conservar una similitud con el lenguaje humano, con la finalidad de que sean más naturales a quienes los usan. Establecen un conjunto de reglas sintácticas y semánticas, las cuales rigen la estructura del programa de computación que se escribe o edita. De esta forma, permiten a los programadores o desarrolladores, poder especificar de forma precisa los datos sobre los que se van a actuar, su almacenamiento, transmisión y demás acciones a realizar bajo las distintas circunstancias consideradas [6].

1.4.1 Del lado del servidor

PHP: Es un lenguaje interpretado de alto nivel embebido en páginas HTML (hipertextos o en inglés, *Hyper Text Markup Language*), enfocado principalmente a la programación de guiones (*scripts*) del lado del servidor, por lo que se puede recopilar datos de formularios, generar páginas con contenidos dinámicos, o enviar y recibir *cookies* [7]. PHP puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno, es rápido y cuenta con una gran librería de funciones y mucha documentación [8].

Se escogió PHP como lenguaje de programación del lado del servidor por ser un lenguaje de alto nivel e interpretado, utilizado en su mayoría para el procesamiento dinámico de información en la web. Dicha selección se basa fundamentalmente en las características y múltiples ventajas que proporciona; teniendo en cuenta que es un lenguaje completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos. Además, es uno de los lenguajes de programación web que más documentación posee, dispone de un sitio oficial que brinda toda la documentación que los desarrolladores necesiten, y cuenta con una gran comunidad de desarrollo. Lo soportan la mayoría de las plataformas de alojamiento web y está en continuo desarrollo.

1.4.2 Del lado del cliente

Para el trabajo en la web se utilizará HTML, CSS (*Cascading Style Sheets*) y JavaScript, todos estos integrados en el *Framework* Bootstrap.

Bootstrap: Es el *framework* de JavaScript, CSS y HTML que se encuentra entre los más usados para el desarrollo adaptativo de proyectos para móviles en la web. Entre sus principales ventajas se encuentran: facilitar y agilizar el desarrollo de interfaces web, está hecho por personas de todos los niveles de habilidades, dispositivos de todas las formas y proyectos de todos los tamaños, permite tener extensa documentación de los elementos HTML más comunes, docenas de componentes HTML y CSS personalizados, con increíbles *plugins* de JQuery, escala de forma fácil y eficiente los sitios web mediante un solo código, desde teléfonos a *tablets* y a escritorio con consultas de CSS. Es de código abierto, alojado, desarrollado y mantenido en Github [9].

HTML 5: Es un lenguaje de marcado para hipertextos (en inglés, *Hyper Text Markup Language*) usado para estructurar y presentar el contenido para la Web, está especificado como un estándar por el W3C (*World Wide Web Consortium*) que permite la creación de páginas web. HTML5 está relacionado también, con la entrada en decadencia del viejo estándar HTML4, que se combinaba con otros lenguajes para producir los sitios que podemos ver hoy en día. Con HTML5, entra en desuso el formato XHTML, dado que ya no sería necesaria su implementación [10].

La aparición del lenguaje influyó notablemente en el crecimiento de Internet, donde la información era distribuida mediante colecciones fragmentadas de textos, imágenes y sonidos. HTML es independiente de la plataforma utilizada y se basa fundamentalmente en el uso de etiquetas estructurales y semánticas, adecuadas para la creación de documentos relativamente simples que permiten simplificar su estructura [11].

CSS3: Mientras que HTML permite definir la estructura de una página web, las hojas de estilo en cascada (*Cascading Style Sheets* o *CSS*) ofrecen la posibilidad de definir las reglas y estilos de representación en diferentes dispositivos, ya sean pantallas de equipos de escritorio, portátiles, móviles, impresoras u otros dispositivos capaces de mostrar contenidos web [12]. Las hojas de estilo permiten definir de manera eficiente la representación de páginas y es uno de los conocimientos fundamentales que todo diseñador web debe manejar a la perfección para realizar su trabajo.

A partir del año 2005 se comenzó a definir el sucesor de esta versión, al cual se le conoce como *CSS3* o *Cascading Style Sheets Level 3*. Actualmente en definición, esta versión ofrece una gran variedad de opciones muy importantes para las necesidades del diseño web actual. Desde opciones de sombreado y redondeado, hasta funciones avanzadas de movimiento y transformación, *CSS3* es el estándar que dominará la web por los siguientes años [12].

1.4.3 *Framework*. Marco de trabajo

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

A continuación, se realiza un estudio de marcos de trabajo para el desarrollo de software, con el objetivo de escoger el más conveniente para la implementación del Sistema de gestión para la lencería de la residencia UCI.

CodeIgniter: Es un marco de trabajo de código abierto para aplicaciones web en PHP. CodeIgniter tiene muchas características que lo hacen destacar entre sus homólogos, ya que a diferencia de otros *frameworks* es muy exhaustivo con la documentación. En el lado de la programación es compatible con

PHP4 y PHP5, por lo que es posible que corra en la mayoría de los servidores web. También, usa el patrón Modelo-Vista- Controlador. Lo que permite organizar la aplicación en diferentes partes como son el modelo, la vista, y el controlador. CodeIgniter también tiene una implementación del patrón Active Record, lo que hace sencillo escribir consultas y hace la aplicación más entendible [13].

Symfony: Symfony es un framework PHP de alto rendimiento de código abierto para el desarrollo de aplicaciones web, construido sobre una serie de componente web reutilizables. Posee la característica de que es fácil de instalar y puede ser usado tanto en Linux como en Windows. Symfony tiene una corta curva de aprendizaje y una comunidad de desarrollo comprometida para llevar PHP al próximo nivel [14].

Zend framework.: Es un marco de trabajo de código abierto de aplicaciones web, orientado a objetos e implementado en PHP 5, desarrollado con el objetivo de simplificar el desarrollo web. Permite a los desarrolladores reutilizar componentes en sus aplicaciones. Zend Framework presenta como desventaja que es una herramienta pesada que gasta mucha memoria, además de ser muy pobre la comunidad de desarrolladores, por lo que podría ser difícil encontrar un buen manual para realizar un proceso o aclarar una duda. La estructura de archivos de Zend no está definida por sí misma, cualquier desarrollador puede modificarla, esto indica que no sería compatible con otra aplicación desarrollada con Zend [15].

Cada uno de los anteriores *frameworks* de desarrollo PHP se ajustan a las necesidades de implementación del sistema propuesto, pero, se decide utilizar Symfony en su versión 4 por que este permite cambiar con facilidad de cualquier sistema gestor de base de datos (SGBD) en las diferentes fases del proyecto, sencillo de utilizar, sigue la mayoría de mejores prácticas y patrones de diseño para la web, código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo y fácil de extender, lo que permite su integración con las librerías de otros fabricantes.

1.5 Herramientas

PHPStorm: PhpStorm es idóneo para trabajar con Symfony, Drupal, WordPress, Zend Framework, Laravel, Magento, Joomla!, CakePHP, Yii y otros frameworks. El editor realmente obtiene su código y comprende profundamente su estructura, y es compatible con todas las funciones del lenguaje PHP para proyectos modernos y heredados. Proporciona la mejor terminación de código, refactorizaciones, prevención de errores sobre la marcha y más. Aprovecha al máximo las tecnologías de vanguardia, como HTML5, CSS, Sass, Less, Stylus, CoffeeScript, TypeScript, Emmet y JavaScript, con refactorizaciones, depuración y pruebas de unidad disponibles. Ver los cambios al instante en el navegador gracias a Live Edit [16].

Visual Paradigm: Constituye una herramienta robusta. Además, de que utiliza UML (por sus siglas en inglés, *Unified Modeling Language*) como lenguaje unificado de modelado, es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto. Permite control de versiones y realizar ingeniería tanto directa como inversa en diferentes lenguajes. Se pueden representar todos los tipos de diagramas UML para las distintas fases del ciclo de vida de un proyecto como la captura de requisitos, análisis, diseño e implementación. Esta herramienta facilita la comunicación, ya que utiliza un lenguaje estándar común a todo el equipo de desarrollo. Presenta la posibilidad de la interoperabilidad con otras aplicaciones como es el *Rational Rose*. Tiene disponible distintas versiones: *Enterprise*, *Professional*, *Standard*, *Modeler*, *Personal* y *Community* (que es gratuita). También, se facilitan licencias especiales para fines académicos [17].

MySQL.: Sistema gestor de base de datos relacional, disponible para múltiples plataformas. Posee un licenciamiento dual, por un lado se ofrece bajo la Licencia Pública General de GNU (GNU-GPL por sus siglas en inglés) para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les

permita este uso. MySQL posee alta velocidad al realizar las operaciones, lo que le hace que posea un buen rendimiento [18]. Es considerado como de mediana escala y es adecuado para aplicaciones web de tamaño medio. MySQL permite que varios usuarios a través de múltiples hilos puedan acceder a datos relacionales y soporta múltiples herramientas para la gestión [19].

Acunetix Web Vulnerability Scanner: Es la herramienta utilizada en la universidad para la detección de vulnerabilidades en sitios y aplicaciones web. Esta herramienta posee un componente que facilita la realización de pruebas a formularios y a áreas protegidas por contraseña. Permite realizar varias peticiones simultáneamente, siendo capaz de explorar cientos de páginas sin interrupciones. Para la realización de las pruebas de seguridad al sistema de alertas se utiliza esta herramienta en su versión 8.0 [20].

JMeter: La aplicación JMeter de Apache es un software de código abierto, una aplicación Java 100% diseñada para cargar el comportamiento funcional y medir el rendimiento. Originalmente fue diseñado para probar aplicaciones web, pero desde entonces se ha expandido a otras funciones de prueba. Apache JMeter puede emplearse para probar el rendimiento en recursos estáticos y dinámicos, aplicaciones dinámicas en la Web. Se puede utilizar para simular una carga pesada en un servidor, grupo de servidores, red u objeto para probar su resistencia o para analizar el rendimiento general bajo diferentes tipos de carga. Las características de Apache JMeter incluyen: Capacidad para cargar y realizar pruebas de rendimiento en diferentes tipos de aplicaciones / servidor / protocolo. Fácil correlación mediante la capacidad de extraer datos de formatos de respuesta más populares, HTML, JSON, XML o cualquier formato textual, portabilidad completa, almacenamiento en caché y análisis offline/ reproducción de los resultados de las pruebas, núcleo altamente extensible [21].

Conclusiones parciales del capítulo

En este capítulo se han abordado los elementos correspondientes a la fundamentación teórica del sistema de gestión, arribando a las siguientes conclusiones:

- El estudio del estado del arte de los sistemas de gestión y los procesos de control y entrega de la lencería, validaron la inexistencia de un sistema informático que satisfaga las necesidades planteadas por la Dirección de Residencia en aras de optimizar este proceso y contribuir a la preservación de los bienes materiales puestos a disposición de la comunidad universitaria.
- El análisis de la metodología de desarrollo de software, las tecnologías, lenguajes y herramientas utilizadas guían el desarrollo de la solución a implementar.

Capítulo 2. Análisis y diseño del Sistema de gestión para la lencería de la residencia UCI

Introducción

En este capítulo se presentan aspectos relacionados con el diseño del sistema a desarrollar. Se caracteriza la propuesta de solución a través de la definición de los requisitos funcionales (RF) y no funcionales (RNF), que deberá presentar el *software*. Además, se generan los artefactos relacionados con estos requisitos identificados; así como la modelación de diagramas.

2.1 Descripción de la propuesta de solución

Después de realizado el estudio y análisis de la situación problemática existente referente a los procesos relacionados con la Dirección de Residencia y teniendo en cuenta que no existe un sistema centralizado que dé cumplimiento total a lo requerido, se propone desarrollar el sistema de gestión para la lencería de la residencia UCI que debe ser empleado para establecer las normas respecto a la entrega, control, tratamiento y destino final de la ropa de cama y otros artículos de lencería a estudiantes, profesores y trabajadores internos.

Entre las actividades que la presente propuesta de solución informatiza se encuentran:

- La gestión de la boleta de control y entrega de avituallamiento a estudiantes del año.
- La gestión de medios y productos químicos del almacén de la lavandería, el cual consiste en registrarlos para que luego sean entregados.
- La gestión de las transferencias realizadas de una lavandería a otra.
- La gestión del destino final de los medios y productos existentes.

La solución muestra al usuario las opciones a las que está autorizado a acceder, para que así, se realicen en las interfaces solo los cambios que le sean permitidos dependiendo de las responsabilidades que posee. La propuesta aporta soluciones rápidas y satisfactorias a las necesidades planteadas por el cliente.

Con el sistema interactúan 8 roles: Vicerrector de Residencia, Director de Residencia # 1, Recepcionista de lavandería, Técnico General de Medios básicos, Encargado de actividades administrativas, Directores de Residencia (son 3), Director General de Residencia.

Tabla 2. Descripción de roles. Fuente: Elaboración propia.

Rol	Descripción
Vicerrector de Residencia	Responsable de aprobar las solicitudes de artículos de lencería.
Director de Residencia # 1	Responsable de tramitar, controlar y supervisar la entrada de los artículos en la lavandería.
Recepcionista de lavandería	Responsable de recepcionar y custodiar los artículos en la lavandería, así como garantizar seguridad de dichos locales.
Técnico General de Medios básicos, Encargado de actividades administrativas, Recepcionista de lavandería	Responsables de cumplir con todo lo que refiere el procedimiento.
Directores de Residencia (son 3), Director General de Residencia	Responsable de hacer cumplir el procedimiento.

2.2 Especificación de requisitos de software

Los requisitos del *software* permiten establecer lo que el sistema debe hacer, sus características fundamentales y las restricciones en el funcionamiento del sistema y los procesos de desarrollo del

software. De manera general, estos requisitos expresan las necesidades objetivas que presentan los usuarios, ante un sistema que resuelve un problema en particular de un determinado dominio [4].

2.2.1 Requisitos funcionales

Los requisitos funcionales de un sistema describen lo que el sistema debe hacer [4].

Tabla 3. Requisitos funcionales. Fuente: Elaboración propia.

No.	Requisito Funcional	Prioridad
RF 1.	Crear usuario	Alta
RF 2.	Editar usuario	Media
RF 3.	Listar usuario	Baja
RF 4.	Eliminar usuario	Media
RF 5.	Crear boleta	Alta
RF 6.	Mostrar boleta	Baja
RF 7.	Eliminar boleta	Media
RF 8.	Actualizar boleta	Media
RF 9.	Archivar boleta	Media

RF 10.	Crear reporte diario	Alta
RF 11.	Mostrar reporte diario	Baja
RF 12.	Eliminar reporte diario	Media
RF 13.	Actualizar reporte diario	Media
RF 14.	Crear boleta de préstamo	Alta
RF 15.	Mostrar boleta de préstamo	Baja
RF 16.	Eliminar boleta de préstamo	Media
RF 17.	Actualizar boleta de préstamo	Media
RF 18.	Crear medio al almacén de lavandería	Alta
RF 19.	Mostrar medio del almacén de lavandería	Baja
RF 20.	Eliminar medio del almacén de lavandería	Media
RF 21.	Actualizar medio del almacén de lavandería	Media
RF 22.	Crear motivo del préstamo	Alta
RF 23.	Mostrar motivo del préstamo	Baja

RF 24.	Eliminar motivo del préstamo	Media
RF 25.	Actualizar motivo del préstamo	Media
RF 26.	Crear producto químico de lavandería	Alta
RF 27.	Mostrar producto químico de lavandería	Baja
RF 28.	Eliminar producto químico de lavandería	Media
RF 29.	Actualizar producto químico de lavandería	Media
RF 30.	Crear transferencia	Alta
RF 31.	Mostrar descripción de transferencia realizada	Baja
RF 32.	Eliminar transferencia	Media
RF 33.	Actualizar transferencia	Media
RF 34.	Mostrar Medios entregados	Baja
RF 35.	Crear expediente de baja	Alta
RF 36.	Mostrar expediente de baja	Baja
RF 37.	Eliminar expediente de baja	Media

RF 38.	Actualizar expediente de baja	Media
RF 39.	Crear destino final	Alta
RF 40.	Mostrar destino final	Baja
RF 41.	Eliminar destino final	Media
RF 42.	Actualizar destino final	Media

2.2.2 Requisitos no funcionales

Los requisitos no funcionales son aquellos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad y la capacidad de almacenamiento [4].

RNF 1. Requisitos de usabilidad.

RNF 1.1. Facilidad de uso por parte de los usuarios: La interfaz debe ser lo más descriptiva posible, permitiendo que las operaciones a realizar por los usuarios estén bien descritas, de manera que se puedan entender claramente. Los botones tendrán nombres que sugieran la operación a realizar. Se mostrarán mensajes informando la realización de las acciones y de errores.

RNF 2. Requisitos de eficiencia.

RNF 2.1. El sistema deberá estar disponible las 24 horas del día, exceptuándose los días que se encuentre en actualización, corrección de errores, o soporte.

RNF 2.2. La aplicación debe estar concebida para un consumo mínimo de recursos, permitiendo de esta forma no afectar el correcto funcionamiento de la computadora en que se esté utilizando la misma.

RNF 3. Requisitos de seguridad.

RNF 3.1. Confidencialidad: en el sistema solo podrá gestionar información el personal autorizado, con los permisos correspondientes, por lo que será necesaria una autenticación previa.

RNF 3.2. Integridad: La información manejada por el sistema tendrá una cuidadosa protección para que no puedan ser modificadas ya que se alterarían inventarios de almacén y de la lavandería permitiendo desvío de recursos y mal manejo del material con que contamos.

RNF 3.3. Disponibilidad: Los usuarios autorizados a poder hacer cambios, entrar datos, y eliminarlos se registrara en el sistema como lo que son y a partir de eso tendrán la posibilidad de gestionar lo que a ellos le corresponde pero no a todo el sistema para la protección de los datos que en ella se encuentran.

RNF 4. Requisitos de software.

RNF 4.1. Para el servidor de base de datos: MariaDB versión 5.0.12

RNF 4.2. Para el servidor web: Apache versión 2.4.34.

RNF 4.3. Dependencias PHP 7.2.9

RNF 4.4. Sistema operativo Nova Servidor (última versión estable) a 64 Bits.

RNF 4.5. PC cliente: cliente ligero y un navegador que soporte HTML5 y CCS3.

RNF 5. Requisitos de hardware.

RNF 5.1. Para el servidor de base de datos: RAM: 8 GB, CPU: i5 4ta generación, Disco duro: 1 TB.

RNF 5.2. Para el servidor web: RAM: 4GB, CPU: i3 4ta generación Disco duro: 50 GB.

RNF 5.3. PC cliente: RAM: 256 MB, CPU: Celeron 1ra generación.

2.3 Historias de usuarios

La Variación AUP-UCI utiliza en su escenario número 4 para describir los requisitos de *software* las Historias de Usuario, estas son descripciones cortas de las necesidades del cliente para el *software* que está en desarrollo [23]. Su utilización es común cuando se aplican marcos de trabajo ágiles [24]. Cada historia es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas [25].

Tabla 4. Historia de usuario del RF 1. Fuente: Elaboración propia.

Número: 1	Nombre del requisito: Crear usuario
Programador: Ilianys Alfonso Milián	Iteración Asignada:
Prioridad: Alta	Tiempo Estimado:
Riesgo en Desarrollo:	Tiempo Real:
<p>Descripción: Muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> ▪ Nombre(s) (Obligatorio. Campo de texto. Permite caracteres alfabéticos y espacios). ▪ Correo electrónico (Obligatorio. Campo de tipo e-mail). ▪ Empresa o Institución (Obligatorio. Campo de texto. Permite caracteres alfabéticos y espacios). 	

- Contraseña (Obligatorio. Campo de tipo *password*. Longitud mínima de 6 caracteres).
- Confirmar contraseña (Obligatorio. Campo de tipo *password*. Longitud mínima de 6 caracteres).

Observaciones:

- Al completar el registro, si no se incumplen las condiciones del mismo, el sistema notifica el registro satisfactorio, en caso contrario se muestra un mensaje de error y señalan los campos que incumplen las condiciones requeridas.

Prototipo de interfaz:

Registro usuario

Nombre(s)

Apellidos

Correo electrónico

Empresa o Insitución

Contraseña

Confirmar contraseña

Tabla 5. Historia de usuario del RF 5. Fuente: Elaboración propia.

Número: 5	Nombre del requisito: Crear una nueva boleta
Programador: : Ilianys Alfonso Milián	Iteración Asignada:
Prioridad: Alta	Tiempo Estimado:
Riesgo en Desarrollo:	Tiempo Real:
<p>Descripción: En caso de que un nuevo estudiante llegue a nuestra residencia el sistema muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> ▪ Nombre(s) (Obligatorio. Campo de texto. Permite caracteres alfabéticos y espacios). ▪ Cl. (Obligatorio. Campo de texto. Permite caracteres numéricos). ▪ No. Solapín (Obligatorio. Campo de texto. Permite caracteres alfanuméricos). ▪ Sexo (Obligatorio. Campo) ▪ Facultad (Obligatorio. Campo) ▪ Seleccionar medio (Campo checkbox) 	
<p>Observaciones:</p> <ul style="list-style-type: none"> ▪ Al completar la boleta, si no se incumplen las condiciones de la misma, el sistema notifica el registro satisfactorio, en caso contrario se muestra un mensaje de error y señalan los campos que incumplen las condiciones requeridas. 	
Prototipo de interfaz:	

Crear nueva boleta

Nombre:

Apellidos:

Carnet. Id:

Sexo:

No. Solapín:

Facultad:

Seleccionar Medio

Sábana camera Sábana personal

Funda personal Toalla

Frazada Funda camera

Almohada Sobrecama

Tabla 6. Historia de usuario del RF 18. Fuente: Elaboración propia.

Número: 18	Nombre del requisito: Crear medio al almacén de lavandería.
Programador: : Ilianys Alfonso Milián	Iteración Asignada:

Prioridad:	Tiempo Estimado:
Riesgo en Desarrollo:	Tiempo Real:
<p>Descripción: En el caso de que se desee adicionar medios al almacén de la lavandería, el sistema muestra un formulario con los campos siguientes.</p> <ul style="list-style-type: none">▪ Nombre (Obligatorio. Campo de texto. Permite caracteres alfabéticos y espacios).▪ No. Factura (Obligatorio. Campo de texto. Permite caracteres alfanuméricos).▪ Motivo (Obligatorio. Campo de texto. Permite caracteres alfabéticos y espacios).	
<p>Observaciones:</p> <ul style="list-style-type: none">▪ Al completar la boleta, si no se incumplen las condiciones de la misma, el sistema notifica el registro satisfactorio, en caso contrario se muestra un mensaje de error y señalan los campos que incumplen las condiciones requeridas.	
Prototipo de interfaz:	

Crear medio al almacén

Nombre:

No. Factura:

Motivo:

Tabla 7. Historia de usuario del RF 20. Fuente: Elaboración propia.

Número: 20	Nombre del requisito: Eliminar medio del almacén de lavandería
Programador: : Ilianys Alfonso Milián	Iteración Asignada:
Prioridad:	Tiempo Estimado:
Riesgo en Desarrollo:	Tiempo Real:

Descripción: En la lista de los medios de la lavandería, en la parte derecha de cada medio, se encontrara el botón eliminar.

Observaciones:

- El botón eliminar solo será mostrado a los usuarios autenticados en el sistema.
- Al hacer clic en el botón eliminar se mostrara un mensaje de confirmación, en el cual el usuario aceptará si desea eliminar el medio.

Prototipo de interfaz:

Medios	No.Factura	Opciones	
Medio 1	MB456123	Editar	Eliminar
Medio 2	MB789456	Editar	Eliminar
Medio 3	MB147258	Editar	Eliminar
Medio 4	MB369258	Editar	Eliminar
Medio 5	MB159263	Editar	Eliminar
Medio 6	MB357241	Editar	Eliminar

Tabla 8. Historia de usuario del RF 26. Fuente: Elaboración propia.

Número: 26	Nombre del requisito: Crear productos químicos de lavandería
Programador: : Ilianys Alfonso Milián	Iteración Asignada:
Prioridad:	Tiempo Estimado:
Riesgo en Desarrollo:	Tiempo Real:
<p>Descripción: En el caso de que se desee añadir productos a la lista de los registrados en el almacén, el sistema muestra un formulario con los campos siguientes.</p> <ul style="list-style-type: none">▪ Nombre(s) (Obligatorio. Campo de texto. Permite caracteres alfabéticos y espacios).▪ Cantidad (Obligatorio. Campo numérico). <p>En la interfaz del usuario se encontrará el botón añadir, que añadirá el producto.</p>	
<p>Observaciones:</p> <ul style="list-style-type: none">▪ La interfaz adicionar productos solo será mostrada a los usuarios autenticados.▪ Al completar el registro, si no se incumplen las condiciones del mismo, el sistema notifica el registro satisfactorio, en caso contrario se muestra un mensaje de error y señalan los campos que incumplen las condiciones requeridas.	
Prototipo de interfaz:	

The image shows a user interface for creating a product. It features a title 'Crear producto' at the top. Below the title, there are two input fields: 'Nombre' followed by a text box, and 'Cantidad' followed by a spinner box containing the number '0'. At the bottom of the form, there are two buttons: 'Aceptar' and 'Cancelar'.

Tabla 9. Historia de usuario del RF 31. Fuente: Elaboración propia.

Historia de usuario del RF 31	
Número: 31	Nombre del requisito: Mostrar descripción de transferencias realizadas.
Programador: : Ilianys Alfonso Milián	Iteración Asignada:
Prioridad:	Tiempo Estimado:
Riesgo en Desarrollo:	Tiempo Real:
Descripción: En caso de que el usuario desee mostrar la descripción de las transferencias realizadas de una lavandería a otra; en la columna derecha se encontrará el botón mostrar, el cual mostrará todos los	

atributos de la transferencia realizada.

- Nombre(s).
- Origen.
- Destino.
- Fecha de transferencia.

Observaciones:

- La interfaz solo será mostrada a los usuarios autenticados.

Prototipo de interfaz:

Transferencias realizadas

Nombre:	Cloro
Origen:	Lavandería X
Destino:	Lavandería X
Fecha de Transferencia:	X / X / X

Aceptar

Detailed description: The image shows a user interface prototype for a 'Transferencias realizadas' (Completed Transfers) screen. The title 'Transferencias realizadas' is centered at the top in a large, bold, black font. Below the title is a white rectangular box containing a form with four rows of data. Each row consists of a label on the left and a value on the right. The labels are 'Nombre:', 'Origen:', 'Destino:', and 'Fecha de Transferencia:'. The values are 'Cloro', 'Lavandería X', 'Lavandería X', and 'X / X / X' respectively. The labels and values are in a standard black font. Below the white box, in the bottom right corner of the overall grey background, is a grey rectangular button with the text 'Aceptar' in black.

2.4 Patrones de diseño

Los patrones de diseño son soluciones para problemas típicos y recurrentes que se pueden encontrar al momento de desarrollar una aplicación. Aunque la aplicación sea única, tendrá partes comunes con otras aplicaciones: acceso a datos, creación de objetos, operaciones entre sistemas, etc. En lugar de reinventar los métodos de desarrollo, se pueden solucionar problemas utilizando algún patrón, ya que son soluciones probadas y documentadas por multitud de programadores [26].

2.4.1 Patrones GRASP

Los patrones *General Responsibility Assignment Software Patterns* (GRASP), en español Patrones Generales de *Software* para Asignar Responsabilidades. Describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones [27].

Patrón Experto: La solución que propone este patrón es la de asignar una responsabilidad a la clase (experto) que cuenta con la información necesaria para cumplirla. Ofrece una analogía con el mundo real ya que da origen al diseño donde el objeto de software realiza las operaciones que normalmente se aplican al elemento real que representa [28].

Patrón Creador: El patrón creador guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento [28].

Patrón Controlador: Este patrón tiene como objetivo asignar la responsabilidad a una clase de recibir o manejar un mensaje de evento del sistema generado por un actor externo, por lo general a través de una interfaz gráfica de usuario a la que accede un usuario para realizar ciertas operaciones en el sistema [28].

2.4.2 Patrones GoF

Los patrones *Gang of Four* (GoF) o “Pandilla de los Cuatro” en español, describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases, la combinación de clases y la formación de estructuras de mayor complejidad. Permiten crear grupos de objetos para ayudar a realizar tareas complejas [29].

Patrón *Observer* (Observador): Define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él.

Patrón *Decorator* (Decorador): Añade dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender la funcionalidad. Para el entorno de este trabajo, se utiliza de conjunto con el patrón Adaptador, mientras el primero crea los elementos, este le asigna los atributos correspondientes; por ejemplo, cada vez que una lista expandible se expande o se contrae, el decorador indica qué mostrar en cada índice y subíndice.

2.5 Patrones arquitectónicos

Los patrones arquitectónicos de *software* permiten representar de forma concreta la estructura y funcionamiento interno de un sistema, lo que posibilita su entendimiento, la organización del desarrollo del mismo. “La arquitectura de *software* de un sistema es la estructura o estructuras del sistema, lo cual abarca componentes de *software*, las propiedades visibles externamente de esos componentes, y las relaciones entre ellas” [30].

2.5.1 Patrón arquitectónico Modelo – Vista – Controlador

Symfony es un completo *framework* diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón Modelo - Vista - Controlador. Esto quiere decir que para los desarrolladores, separa la lógica de negocio, la lógica de servidor y la capa de presentación.

Aunque *Symfony framework* utiliza el patrón de diseño Modelo Vista Controlador, tiene su propia forma de trabajo.

En línea con Modelo - Vista - Controlador, los archivos de la aplicación se distribuyen en carpetas según su función. La estructura que propone *Symfony 4* es opcional, si no se requiere una carpeta (por ejemplo, *template*) no es necesario crear al directorio.

Model View Controller (MVC): (Modelo Vista Controlador). Es una forma muy popular de organizar el código. La gran idea detrás de MVC es que cada sección de su código tiene un propósito. Esos propósitos son diferentes. Una parte del código contiene los datos de la aplicación, parte del código hace que la aplicación tenga un buen aspecto y parte del código controla cómo funciona la aplicación [31].

- **Modelo:** el código del modelo generalmente refleja cosas del mundo real. Este código puede contener datos en bruto, o definirá los componentes esenciales de su aplicación. Por ejemplo, si estuviera creando una aplicación de Tareas pendientes, el código del modelo definiría qué es una “tarea” y qué es una “lista”.
- **Vista:** el código de vista se compone de todas las funciones que interactúan directamente con el usuario. Este es el código que hace que el aspecto de la aplicación sea bueno y además cómo el usuario la ve e interactúa con ella.

- **Controlador:** el código del controlador actúa como un enlace entre el modelo y la vista, recibiendo las entradas del usuario y decidiendo qué hacer con ellas. Es el cerebro de la aplicación y une el modelo y la vista.

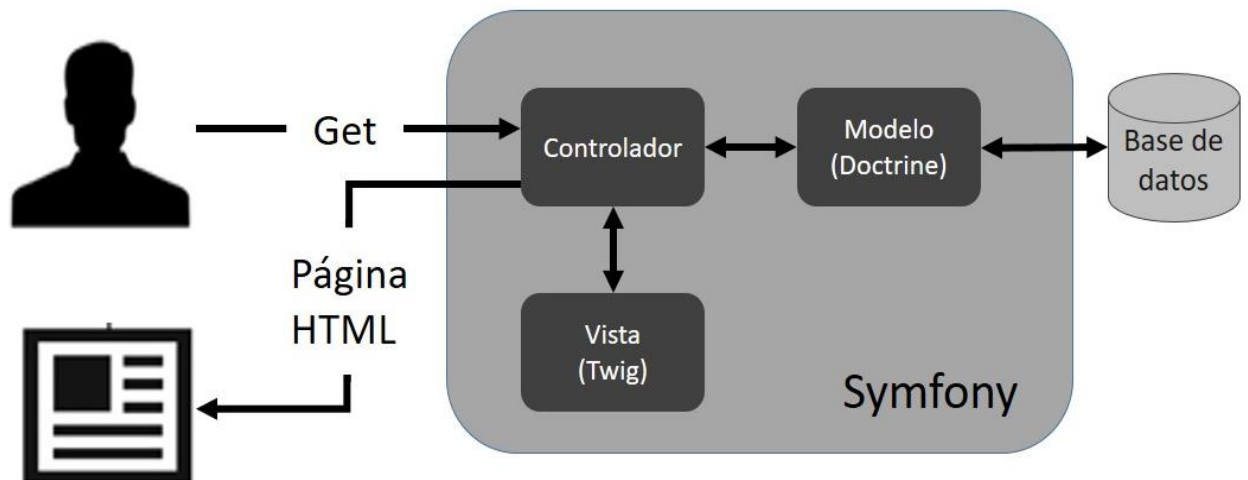


Figura 1. Modelo-Vista-Controlador. Fuente: Elaboración propia.

2.6 Modelo de diseño

El modelo de diseño muestra los objetos o clases de un sistema y los diferentes tipos de relaciones entre estas entidades. Es el puente entre los requerimientos y la implementación del sistema. Está compuesto por clases del diseño, subsistemas, paquetes, colaboraciones, y las relaciones entre ellos.

2.6.1 Diagrama de clases de diseño

Un diagrama de clases de diseño (DCD) representa las especificaciones de las clases e interfaces *software* en una aplicación. A diferencia de las clases conceptuales del Modelo del Dominio, las clases de

diseño de los DCD muestran las definiciones de las clases *software* en lugar de los conceptos del mundo real [32].

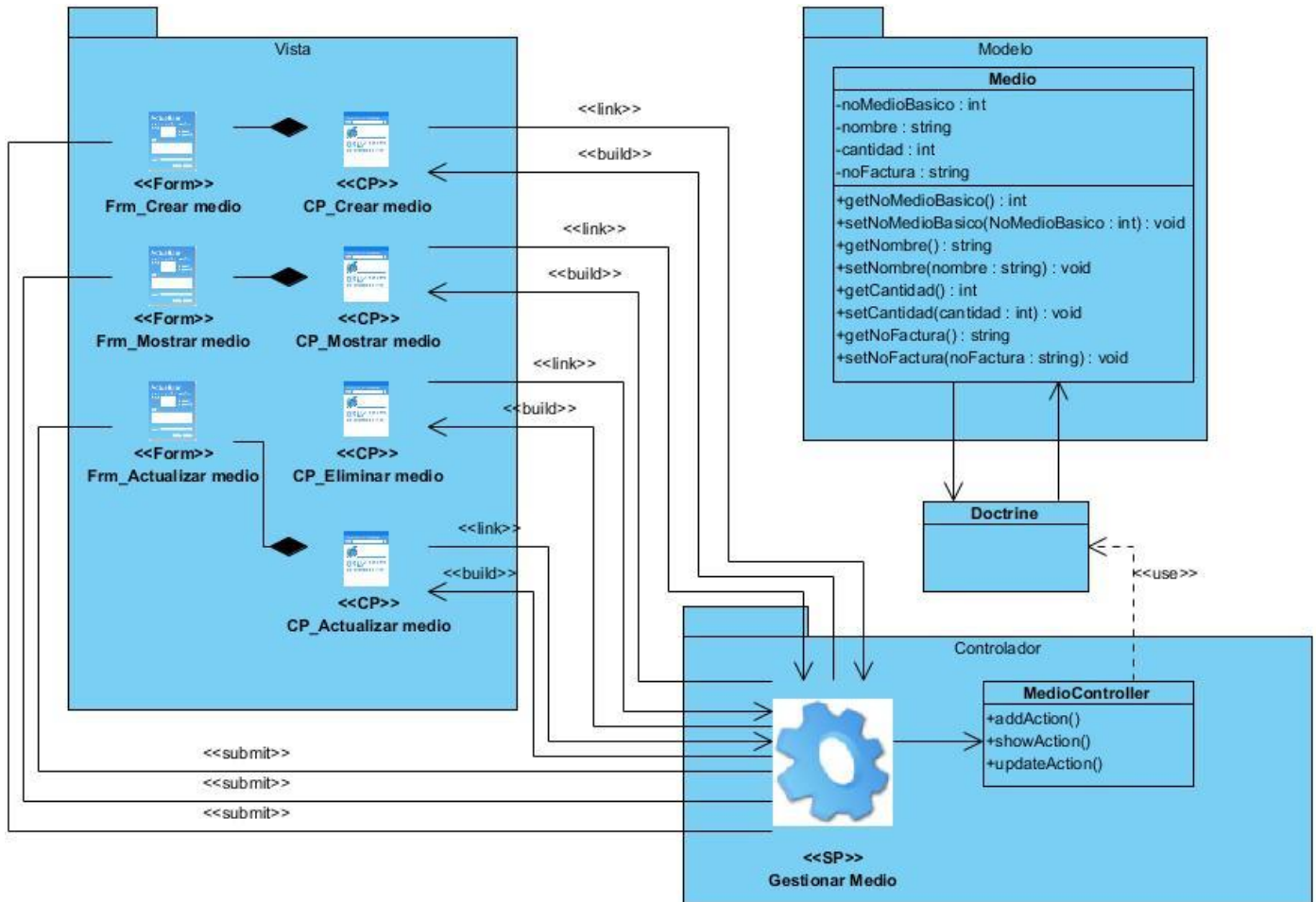


Figura 2. DCD con estereotipos web medio. Fuente: Elaboración propia

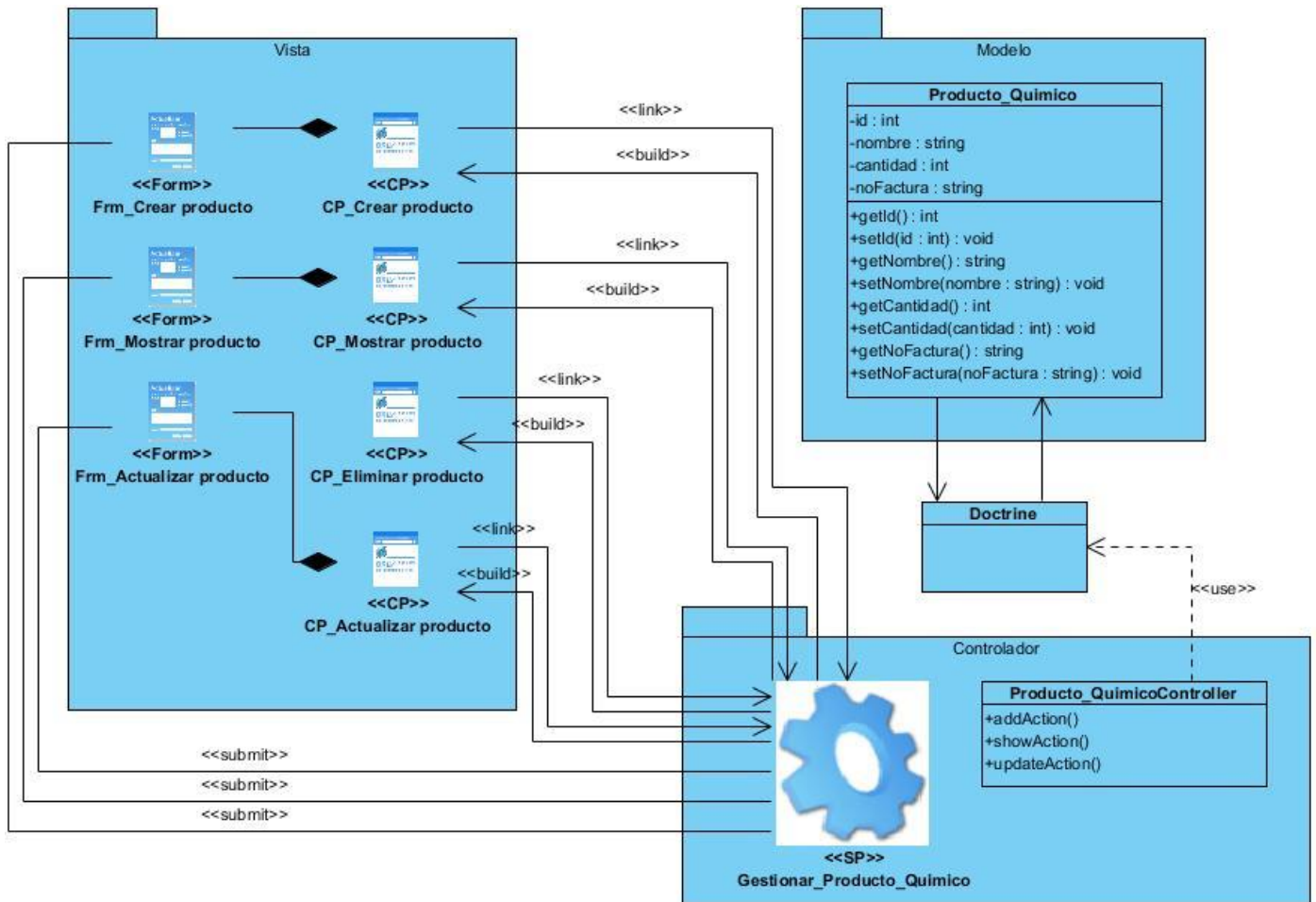


Figura 3. DCD con estereotipos web producto químico. Fuente: Elaboración propia

2.7 Modelo de datos

A continuación, se muestra el modelo de datos físicos de la aplicación, el cual es un esquema de diseño para activos de información que define las estructuras físicas y las relaciones de los datos dentro de un dominio o aplicación específica [33]. El modelo de datos físicos es un modelo específico que representa objetos en las bases de datos y sus relaciones sin tener en cuenta la estructura funcional de la aplicación.

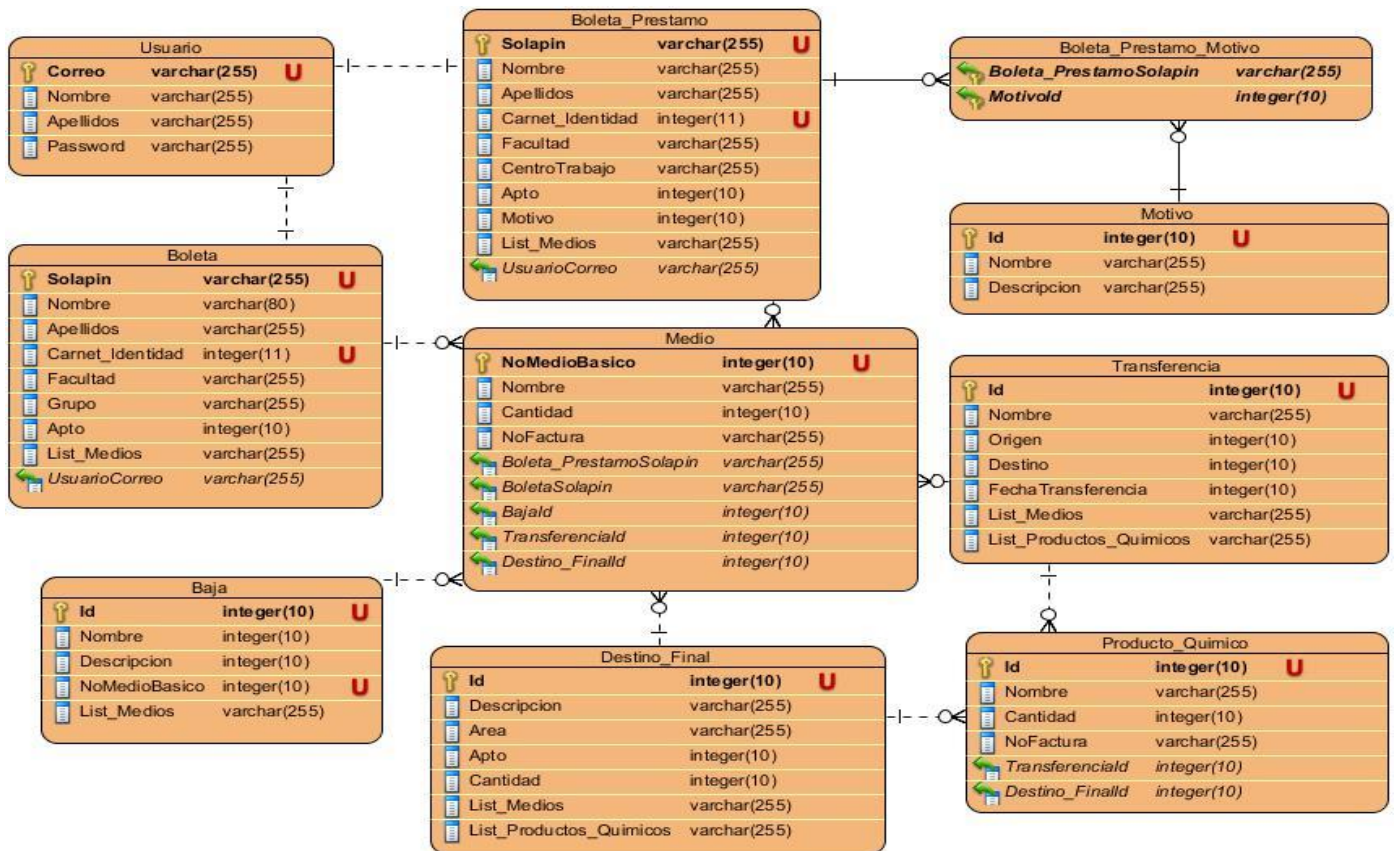


Figura 4. Modelo de datos. Fuente: Elaboración propia

2.8 Modelo de despliegue

Los modelos de despliegue son los complementos de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema. Describen la topología del sistema, la estructura de los elementos de *hardware* y el *software* que ejecuta cada uno de ellos. Los diagramas de despliegue representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red [33].

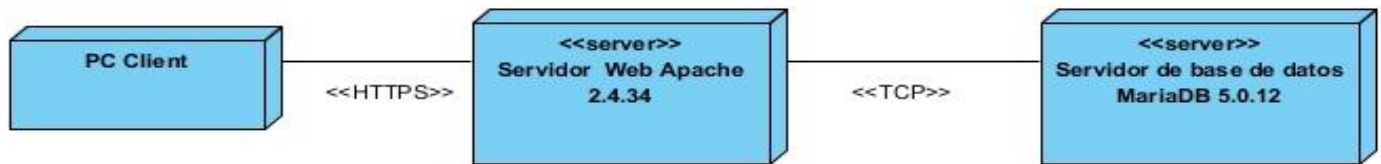


Figura 5. Modelo de despliegue. Fuente: Elaboración propia.

PC Cliente: representa las computadoras que se conectan al servidor de aplicaciones, las mismas se comunican con el servidor a través del protocolo seguro HTTPS.

Servidor web: representa el servidor donde se encuentra instalada la aplicación web. Este accede al servidor de base de datos para el manejo de la información mediante el protocolo TCP.

Servidor de Base de datos: es donde se almacena toda la información de la aplicación.

Conclusiones parciales del capítulo

En el desarrollo de este capítulo se trataron aspectos correspondientes al análisis y diseño del sistema de gestión para la lencería en la residencia UCI. Se definieron detalladamente las características con las que debe cumplir y los requisitos a tener en cuenta durante la etapa de implementación. Basándose en estas actividades se arriba a las siguientes conclusiones:

- Los requisitos funcionales y no funcionales brindan una profunda caracterización de la aplicación y las funciones que esta debe realizar, así como la representación y descripción de los artefactos generados, permitieron una mayor comprensión del negocio y obtener una visión más clara de los resultados que se pretenden obtener.
- La arquitectura MVC seleccionada permitió definir la estructura del software y su interrelación entre los diferentes componentes.
- El diagrama de despliegue propuesto permitió exponer cómo se relacionan los elementos de hardware del sistema.

Capítulo 3. Implementación y validación del Sistema de gestión para la lencería de la residencia UCI

Introducción

La implementación del sistema es una de las fases imprescindibles dentro del proceso de desarrollo de software. Esta fase comprende la materialización, en forma de código, de todos los artefactos, descripciones y arquitectura propuestos en la etapa de análisis y diseño; con el objetivo de conformar el producto final requerido por el cliente [28].

Aparejado al proceso de implementación, el software que se crea debe ser sometido a pruebas que garanticen la correspondencia entre el producto y los requisitos definidos en las etapas anteriores. A esta etapa se le conoce como validación del sistema y en ella, pueden realizarse diferentes tipos de pruebas en función de los objetivos de las mismas.

3.1 Modelo de componentes que integran el sistema

El modelo de componentes representa la forma en que es estructurado un sistema informático atendiendo a las diferentes partes o componentes físicos de un sistema, así como las relaciones entre ellos [34]. El modelo enunciado permite visualizar la estructura de alto nivel del sistema y el comportamiento del servicio que estos componentes proporcionan y usan a través de interfaces [35].

3.1.1 Diagramas de componentes

El diagrama de componentes muestra las interacciones y relaciones de los componentes de un modelo. Entendiéndose como componente una clase de uso específico, que puede ser implementada desde un entorno de desarrollo, ya sea de código binario, fuente o ejecutable; dichos componentes poseen tipo, que

indican si pueden ser útiles en tiempo de compilación, enlace y ejecución. Este tipo de diagrama se representa mediante componentes unidos mediante relaciones de dependencia [36].

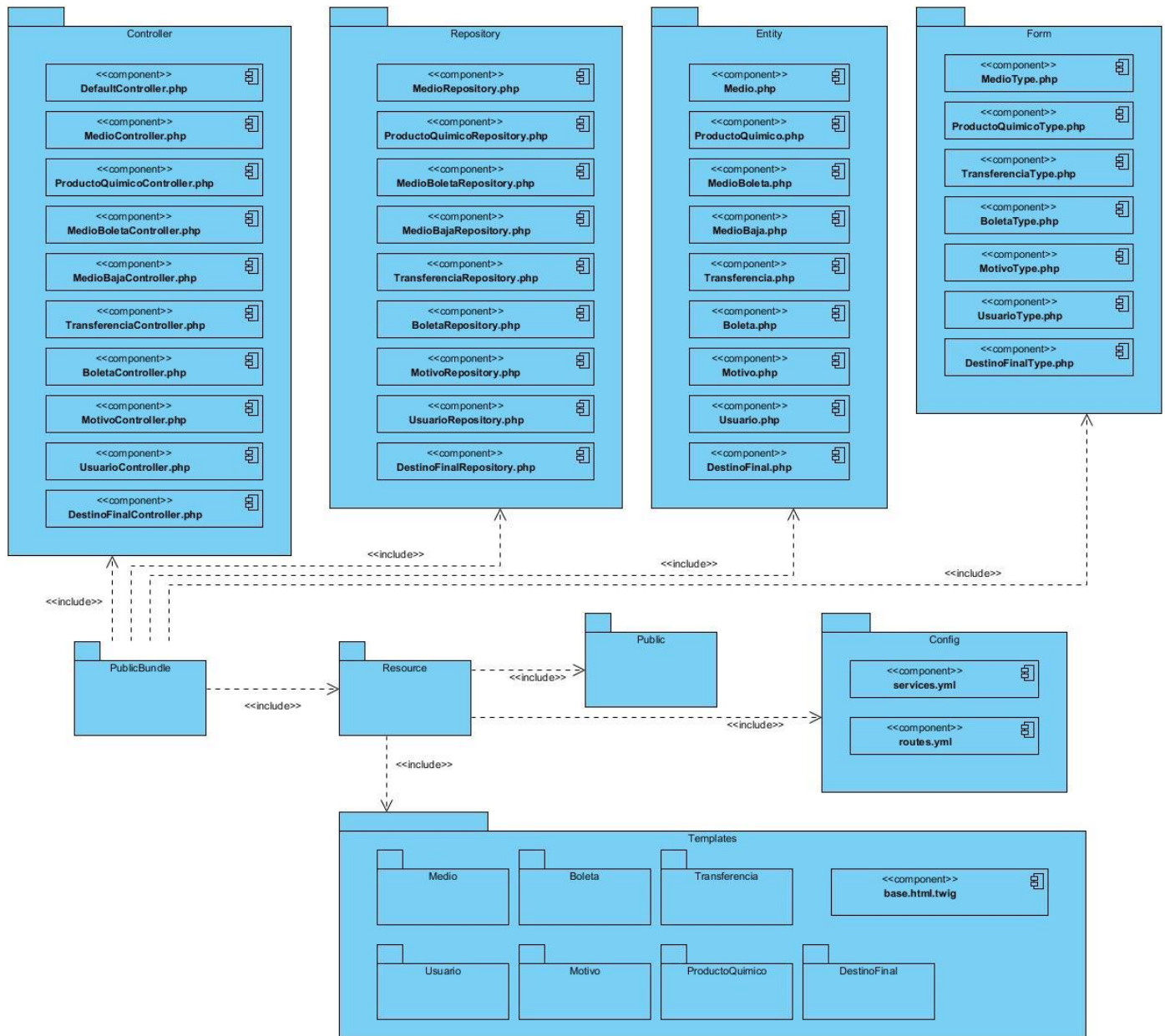


Figura 6. Diagrama de Componentes. Fuente: Elaboración propia.

3.2 Estándares de codificación

Los estándares de codificación son especificaciones o estilos que establecen la forma de generar el código de las aplicaciones informáticas. En ocasiones, los sistemas de cómputo son implementados por varios programadores, la adopción inicial de un único estilo de codificación constituye uno de los factores de mayor peso en la calidad, rendimiento, legibilidad y capacidad de mantenimiento del producto final [37].

En el lenguaje PHP se utilizó el estándar de codificación que establece el *Framework Symfony*, el cual sigue los estándares definidos en los documentos PSR-0, PSR-1, PSR-2, PSR-4.

Clases y métodos

CamelCase: es una forma de notación de texto que sigue el patrón de palabras en minúscula sin espacios y con la primera letra de cada palabra en mayúsculas exceptuando la primera palabra. Este estándar fue utilizado para la definición de clases y métodos en el código de la aplicación.

Existen dos tipos de *CamelCase*:

UpperCamelCase, cuando la primera letra de cada una de las palabras es mayúscula.

Ejemplo: *EjemploDeUpperCamelCase*.

lowerCamelCase, igual que la anterior con la excepción de que la primera letra es minúscula.

Ejemplo: *ejemploDeLowerCamelCase*.

En la aplicación se utilizó *UpperCamelCase* para nombrar las clases existentes y *LowerCamelCase* para la nomenclatura de los métodos.

```
/**
 * @Route("/medio")
 */
class MedioController extends AbstractController
{
    /**
     * @Route("/", name="medio_index", methods={"GET"})
     */
    public function index(MedioRepository $medioRepository):
    Response
    {
        return $this->render('medio/index.html.twig', [
            'medios' => $medioRepository->findAll(),
        ]);
    }
}
```

Figura 7. Ejemplo de uso de CamelCase. Fuente: Elaboración propia.

3.3 Validación del Sistema de gestión para la lencería de la residencia UCI

La validación del sistema incluye un conjunto de actividades para asegurar que el software desarrollado se corresponde con los requisitos del cliente. Dentro de estas actividades se encuentran las pruebas de validación, las cuales tienen como objetivo evaluar la calidad y de manera más pragmática, descubrir errores en el sistema [38]. A continuación, se describen las pruebas realizadas al sistema de gestión para la lencería de la residencia UCI, así como los resultados obtenidos.

3.3.1 Pruebas funcionales

Las pruebas funcionales se aplican a un software determinado, con el objetivo de validar que las funcionalidades implementadas se desempeñen de acuerdo a las especificaciones de los requisitos definidos con anterioridad. Al conocer las funciones específicas que se le asignaron a una aplicación para su realización, pueden llevarse a cabo casos de pruebas que demuestren que cada función es completamente operativa mientras que al mismo tiempo se buscan errores en cada función [39].

Dichas pruebas se realizan a partir de una matriz de datos donde:

V: indica válido.

I: indica inválido.

NA: indica que no es necesario proporcionar un valor del dato, ya que es irrelevante.

Tabla 10. Descripción de las variables correspondientes al caso de prueba para las historias de usuarios "Gestionar perfil de usuario autenticado. Fuente: Elaboración propia.

No	Nombre del campo	Clasificación	Valor nulo	Descripción
V1	Nombre	Campo de texto	no	Se deben escribir palabras comenzando por mayúscula y sin caracteres extraños, que no excedan un total de 60 caracteres.
V2	Apellidos	Campo de texto	no	Se deben escribir palabras comenzando por mayúscula y sin caracteres extraños, que no excedan un total de 60 caracteres.
V3	Correo	Campo de texto	no	Se debe insertar una cadena de texto, que coincida con la estructura válida de un correo.

V4	Empresa o institución	Campo de texto	no	Se deben escribir palabras comenzando por mayúscula y sin caracteres extraños, que no excedan un total de 60 caracteres.
V5	Contraseña	Campo de texto	no	Se debe insertar una cadena de texto que no exceda de 60 caracteres.
V6	Confirmar Contraseña	Campo de texto	no	Se debe insertar una cadena de texto que no exceda de 60 caracteres, idéntica a la del campo "Contraseña".

Tabla 11. Caso de prueba para la historia de usuario: Sección 1: "Crear usuario". Fuente: Elaboración propia.

Sección 1: "Crear usuario"									
Escenario	Descripción	V1	V2	V3	V4	V5	V6	Respuesta del sistema	Flujo central
EC 1.1 Insertar datos personales	Se insertan todos los datos personales del usuario	V	V	V	V	V	V	Inserta un nuevo usuario. Muestra la lista de los usuarios creados y un mensaje indicando que el usuario fue creado	<ol style="list-style-type: none"> Se selecciona la opción "Crear usuario" en la página principal. Se

								satisfactoriamente.	introducen los datos. 3. Se selecciona la opción "Crear".
EC 1.2 Dejar campos vacíos.	Se intenta insertar un usuario dejando campos vacíos.	I	I	I	I	I	I	Señala los campos que aún estén vacíos, para que se llenen. Muestra un mensaje indicando que el campo no puede estar vacío.	1. Se selecciona la opción "Crear Anuncio" en la página principal. 2. Se introducen los datos. 3. Se selecciona la opción "Crear".
EC 1.3 Insertar datos inválidos.	Se intenta insertar un usuario con datos incorrectos.	I	I	I	I	I	I	Señala los campos que tengan datos inválidos. Muestra un mensaje indicando que el dato entrado	1. Se selecciona la opción "Crear usuario" en

									es incorrecto.	la página principal. 2. Se introducen los datos. 3. Se selecciona la opción "Crear".
--	--	--	--	--	--	--	--	--	----------------	--------------------------------------------------------------------------------------

Tabla 12. Caso de prueba para la historia de usuario: Sección 2: "Editar usuario". Fuente: Elaboración propia.

Sección 2: "Editar usuario"										
Escenario	Descripción	V1	V2	V3	V4	V5	V6	Respuesta del sistema	del	Flujo central
EC 2.1 Insertar datos editados de un usuario correctamente.	Se guardan correctamente los cambios realizados.	V	V	V	V	V	V	Se guarda el usuario con los nuevos datos editados. Muestra un mensaje indicando que el usuario fue guardado satisfactoriamente.		1. Se selecciona la opción "Editar". 2. Se editan los datos. 3. Se selecciona la

									opción "Editar".
EC 2.2 Dejar campos vacíos.	Se intenta editar un usuario dejando campos vacíos.	I	I	I	I	I	I	Señala los campos que aún estén vacíos, para que se llenen. Muestra un mensaje indicando que el campo no puede estar vacío.	<ol style="list-style-type: none"> Se selecciona la opción "Editar". Se editan los datos. Se selecciona la opción "Editar". Se muestran mensajes de error.
EC 2.3	Se intenta	I	I	I	I	I	I	Señala los campos que	<ol style="list-style-type: none"> Se selecciona la

Editar con datos inválidos.	editar un usuario con datos incorrectos.							tengan datos inválidos. Muestra un mensaje indicando que el dato entrado es incorrecto.	opción "Editar". 2. Se editan los datos. 3. Se selecciona la opción "Editar". 4. Se muestran mensajes de error.
-----------------------------	------------------------------------------	--	--	--	--	--	--	-----------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------

Tabla 13. Caso de prueba para la historia de usuario: Sección 3: "Eliminar anuncio". Fuente: Elaboración propia.

Sección 3: "Eliminar usuario"			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 3.1 Eliminar correctamente un usuario.	Se elimina un usuario del sistema.	Elimina un usuario de la base de datos del sistema y muestra el listado de los usuarios restantes. Se muestra un mensaje comunicando que el usuario ha sido eliminado satisfactoriamente.	1. Se selecciona la opción "Eliminar" del usuario. 2. Se muestra un mensaje comunicando que el usuario ha sido eliminado satisfactoriamente.

Tabla 14. Caso de prueba para la historia de usuario: Sección 4: “Listar usuario”. Fuente: Elaboración propia.

Sección 4: “Listar usuario”			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 4.1 Listar usuarios correctamente.	Se listan los usuarios del sistema.	Muestra el listado de los usuarios existentes.	1. Se selecciona la opción “Lista de usuario”.

3.3.2 Pruebas de carga y estrés

Las pruebas de carga y estrés se diseñan para enfrentar los programas a situaciones anormales. Evalúan la robustez y la confiabilidad del *software* sometiéndolo a condiciones de uso extremas. Ejecutan un sistema, de manera que demande recursos en cantidad, frecuencia o volúmenes extremos [39].

Para realizar las pruebas de carga y estrés al sistema de gestión para la lencería de la residencia UCI se utiliza la herramienta Apache JMeter en su versión 3.1, pues resulta necesario comprobar el rendimiento del sistema soportando una cantidad máxima de usuarios que soliciten recursos en la web.

Descripción de los parámetros evaluados:

Muestras: cantidad de hilos utilizados para la URL.

Media: tiempo promedio en milisegundos para un conjunto de resultados.

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

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

%Error: porcentaje de error de las respuestas de las peticiones.

Rendimiento: rendimiento medido en los requerimientos por segundo / minuto / hora.

Kb/s Recibidos: rendimiento medido en Kbytes por segundo.

3.3.3 Pruebas de seguridad

Se considera un producto (de software) seguro aquel que protege la confidencialidad, integridad y disponibilidad de los recursos y el servicio, bajo el control del propietario o administrador del sistema [40].

El objetivo principal de la prueba y análisis de la seguridad de software es la comprobación de que el software exhibe las propiedades y conductas deseables siguientes [41]:

- Su comportamiento es predecible y seguro.
- No expone ninguna vulnerabilidad o debilidades.
- Las rutinas que se ocupan de los errores y excepciones le permiten que mantenga un estado seguro frente a modelos de ataque o fallas intencionales.
- Satisface todos los requisitos no funcionales de seguridad especificados e implícitos.
- No viola ninguna restricción de seguridad especificada.
- El código ha sido disimulado para evitar la ingeniería inversa.

3.3.4 Pruebas de usabilidad

La usabilidad es la capacidad de un producto de *software* de ser entendido, aprendido, utilizado y atractivo al usuario, cuando es utilizado bajo condiciones especificadas. Es un atributo intangible del *software*, por lo que es difícil de visualizar, medir y reconocer como un factor determinante de su calidad. Una mayor

atención a este elemento contribuiría a incrementar la calidad del producto percibida por el usuario, sin un aumento excesivo en el costo de desarrollo [42].

Las pruebas de usabilidad intentan encontrar problemas de usabilidad del sistema en su interacción con humanos. Según el enfoque tradicional, estas pruebas se aplican sobre el producto software para garantizar o determinar si el mismo alcanza un nivel aceptable de usabilidad. Entre los modelos que presenta para evaluar se encuentran los heurísticos y los empíricos. Los primeros implican la participación de expertos especialistas en usabilidad, mientras que los segundos constan de técnicas que implican la participación de los usuarios [42].

Para la realización de las pruebas de usabilidad a la herramienta se definieron los siguientes pasos:

Confección de una lista de especificaciones de usabilidad.

Selección de un conjunto de usuarios.

Especificación de las métricas a evaluar durante las pruebas.

Evaluación de la usabilidad del sistema mediante un *test* de usabilidad.

Para la confección de la lista de especificaciones de usabilidad se toma en consideración algunos de los requisitos funcionales que se identificaron durante el proceso de levantamiento de requisitos de la herramienta.

El proceso de selección de los usuarios que participan en las pruebas se realiza de forma aleatoria. Se seleccionaron 5 individuos de un conjunto de posibles candidatos.

Tabla 15. Candidatos seleccionados para las pruebas de usabilidad. Fuente: Elaboración propia.

No.	Experto	Edad
1	María Yaquelín Fajardo Mosegues	42

2	Roberto Rodríguez Rodríguez	43
3	Yaniel Cruz Nuñez	26
4	Reimer Darián Mallesá Romero	23
5	Rachel Vázquez Barrera	18

Tabla 16. Métricas a evaluar en las pruebas de usabilidad. Fuente: Elaboración propia.

Métrica	Descripción	Unidades de medida	Valores
Facilidad de Aprendizaje	Indica qué tan fácil es aprender la funcionalidad básica del sistema.	Cualitativa	Muy Adecuado
			Adecuado
			Inadecuado
Presentación visual apropiada	Indica la correcta selección de colores, tipos de letra, la disposición de los elementos en una ventana, entre otros aspectos de diseño gráfico.	Cualitativa	Muy Adecuado
			Adecuado
			Inadecuado

Resultados de las pruebas de usabilidad

Luego de seleccionados los individuos de pruebas, se sometieron a un *test* de usabilidad definido para evaluar Sistema de gestión para la lencería de la residencia UCI.

Tabla 17. Resultados de las pruebas de usabilidad. Fuente: Elaboración propia.

Métricas evaluadas	Valores	Frecuencia Absoluta	Frecuencia Relativa
Facilidad de Aprendizaje	Muy Adecuado	13	0.86
	Adecuado	2	0.14
	Inadecuado	0	0
Presentación visual apropiada	Muy Adecuado	11	0.73
	Adecuado	4	0.27
	Inadecuado	0	0

Los resultados obtenidos, tras aplicar las pruebas de usabilidad a la herramienta desarrollada, muestran que, de manera general, la aplicación presenta un grado de usabilidad muy adecuado de acuerdo a las especificaciones requeridas por el cliente. Esto lo corroboran los niveles de satisfacción de los individuos involucrados en las pruebas.

CONCLUSIONES

El presente trabajo de diploma da cumplimiento a las tareas de investigación propuestas inicialmente. De esta manera se puede concluir que:

- Se valoró el estado del arte de los sistemas de gestión de medios, garantizando un mejor entendimiento sobre el funcionamiento de los procesos y permitiendo identificar funcionalidades que sirvieron de guía para dar solución al problema planteado.
- Se obtuvo un conjunto de tecnologías y herramientas identificadas en el análisis de propuestas existentes y que posteriormente fueron utilizadas en el desarrollo del subsistema implementado, obteniéndose un buen funcionamiento del mismo.
- La metodología utilizada, permitió guiar todo el proceso de desarrollo del *software*, generando los principales artefactos que permitieron obtener una arquitectura sólida para el sistema.
- Se diseñó e implementó el conjunto de clases que dan cumplimiento a los requisitos funcionales y no funcionales asociados a los casos de uso de la solución propuesta.

Recomendaciones

Los resultados obtenidos, luego del desarrollo de la presente investigación, satisfacen los requisitos definidos, no obstante se recomienda:

1. Realizar las pruebas seguridad y de carga y estrés.
2. Culminar la implementación del sistema de gestión para la lencería de la residencia UCI.
3. Utilizar el presente documento para futuras investigaciones.

BIBLIOGRAFÍA

1. Sistemas de Gestión [En línea], 2016. [Citado el:] Disponible en: <http://www.consultoresdesistemasdegestion.es/sistemas-de-gestion>.
2. FÁBREGA AVILA, C. Implementación del módulo de inventario del subsistema Activo Fijo Tangible del Sistema Integral de Gestión CEDRUX. Trabajo de Diploma. Universidad de las Ciencias Informáticas. La Habana, 2010.
3. SOMMERVILLE, Ian y GALIPIENSO, María Isabel Alfonso. *Ingeniería del software*. s.l. : Pearson Educación, 2005. SBN 9788478290741.
4. ZAMBRANO, R. A. Metodología de la investigación. [En línea] 2013. [Citado el:] <http://es.slideshare.net/Mal3J1Ta/resumen-capitulos1234-del-libro>.
5. Munguía, Enrique. *¿Qué lenguaje de programación debo elegir?* 7 de diciembre de 2015.
6. Nacional, RENA. Red Escolar. RENA. RENA. [En línea] Ministerio del Poder Popular para Educación Universitaria, Ciencia y Tecnología de Venezuela, 2008. [Citado el:] www.rena.edu.ve/cuartaEtapa/Informatica/Tema13.html.
7. Arenas, Jairo. Ventajas y desventajas de los lenguajes de programación web. 11 de agosto de 2015.
8. Otto, Mark y Thornton, Jacob. Bootstrap 3, el manual oficial. [En línea] 2 de diciembre de 2016. [Citado el:] http://librosweb.es/libro/bootstrap_3/.
9. Bootstrap. Página Oficial de Bootstrap. *Bootstrap*. [En línea] [Citado el:] <http://getbootstrap.com>.
10. Bitelia. 2015. [Online] 2015. [Citado el:] <http://bitelia.com/2013/05/entendiendo-html5-guia-paraprincipiantes>.
11. Consortium, W. 2014. World Wide Web Consortium(W3C). [Online] 2014.

12. CSS3. 2016. CSS3 HTML5. [Online] 2016. <http://html5.dwebapps.com/que-es-css3/>.
13. Symfony. Symfony PHP Framework. [En línea] [Citado el:] <https://symfony.com>.
14. Rivera, Navarrete Ricardo. *Zend Framework 2 vs Symfony2*. s.l.: Perú: s.n., 2016.
15. JetBrains. [En línea] [Citado el:] https://www.jetbrains.com/help/phpstorm/2018.2/?top&utm_source=product&utm_medium=link&utm_campaign=PS&utm_content=2018.2.
16. Acunetix. Audit your website security with Acunetix Web Vulnerability Scanner. [En línea]. Acunetix. [En línea] 2016. [Citado el:] <http://www.acunetix.com>.
17. Visual Paradigm. Visual Paradigm for UML. [En línea] 2010. [Citado el:]. <http://www.visualparadigm.com/>.
18. PHP y MySQL: Tecnología para el desarrollo de aplicaciones web. -Ángel Cobo -Google Libros.[En línea] [Citado el :] Disponible en:
http://www.google.com/cu/books?hl=es&lr=&id=zMK3GOMOpQ4C&oi=fnd&pg=PR17&dq=php+es+un+lenguaje+de+programaci%C3%B3n+&ots=FeizY3Hfvi&sig=6Fb7-SA1M8ZKqOyahPKJIHRplhk&redir_esc=y#v=onepage&q=php%20es%20un%20lenguaje%20de%20programaci%C3%B3n&f=false.
19. KEMPPAINEN, T, Data Archive Project [en línea]. 2015. Helsinki: Helsinki Metropolia University of Applied Sciences. [Citado el:] Disponible en:
http://www.theseus.fi/bitstream/handle/10024/93069/Kemppainen_TimoPekka.pdf?sequence=1.
20. Patrones.pdf., 08-Patrones1.ppt -. [En línea] [Citado el:] <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf>.

21. ApacheJMeter. [En línea], 2016. [Citado el:]. Disponible en:
<http://jmeter.apache.org>.
22. ZAMBRANO, R. A. Metodología de la investigación. [En línea] 2013. [Citado el: .]
<http://es.slideshare.net/Mal3J1Ta/resumen-capitulos1234-del-libro>.
23. RODRÍGUEZ SÁNCHEZ, T. Metodología de desarrollo para la Actividad productiva de la UCI v1.2. 2015, La Habana
24. Suaza, K. V. Definición de equivalencias entre historias de usuario y especificaciones en UNLENCEP para el desarrollo ágil de software [Tesis]. Universidad Nacional de Colombia: Medellín, Colombia, 2013.
25. get.php., Tema 1 Patrones Arquitectónicos -. [En línea] [Citado el:.]
<http://www.lsi.us.es/docencia/get.php?id=1130>.
26. Fernandez Ruben. Patrones de diseño: qué son y por qué debes usarlos [En línea] 2014 [Citado el::].
<https://www.genbetadev.com/metodologias-de-programacion/patrones-de-diseno-que-son-y-por-que-debesusarlos>.
27. desarrollo-agil-symfony-2.1.pdf., Desarrollo web ágil con Symfony2 -. [En línea] [Citado el: .]
<http://sunshine.prod.uci.cu/gridfs/sunshine/books/desarrolloagil-symfony-2.1.pdf> .
28. LARMAN, C. 2004. UML y Patrones:una introducción al analisis y diseño orientado a objetos y al proceso unificado. s.l. : Segunda.s.l.: Prentice hall.
29. Pressman, R. S. *Ingeniería de software*. s.l. s.l. : McGraw-Hill Interamericana de España, 2010. ISBN:9786071503145.

30. Systems, Sparx. . Diagrama de Despliegue UML 2. . *Sparx Systems - Tutorial UML 2 - Diagrama de Despliegue*. [En línea] [Citado el: .]
http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.
31. Acosta, Victor Manuel. Symphony framework: ¿Cómo desarrollar una aplicación con Symphony? Revista digital INESEM. 2019. Disponible en: <https://revistadigital.inesem.es/informatica-y-tics/symfony-el-framework-de-moda/>
32. LARMAN, C. UML y Patrones. 2003, vol. 2. [En línea]. [Citado el:].
Disponible en: <http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDiseno.pdf> .
33. Craig, Larman Prentice Hall. UML y Patrones. 2ª Edición. 2003. [Citado el: .].
34. IEEE. 2004. Guide to the Software Engineering Body of Knowledge. 2004. 0-7695-2330-7.
35. Microsoft. 2016. Microsoft. [Online] 2016. <http://www.microsoft.com>.
36. Rumbaugh, James, Jacobson, Ivar y Booch, Grady. El Lenguaje Unificado de Modelado. Manual de Referencia. 2007. [Citado el:].
37. Visual Studios.NET. 2016. Visual Studios.NET. Revisiones de código y estándares de codificación. [Online] Microsoft, 2016. [Cited: abril 15, 2017.]
38. Pressman, R S. 2006. Ingeniería del software. Un enfoque práctico. 2006. Vols. ISBN 970-10-5473-3.
39. (PSR-1)., PHP FRAMEWORK INTEROP GROUP. www.phpfig.org. *PSR-1: Basic Coding Standard*. [En línea] [Citado el: 28 de marzo de 2019.] <http://www.php-fig.org/psr/psr-1/>.
40. Leblanc, M. y Howard, D. 2006. Writing Secure Code. 2006. ISBN-13: 978-0735617223.

41. Vega, R Y. 2012. Guía metodológica para implementar la seguridad durante el desarrollo de aplicaciones informáticas. La Habana. 2012.
42. Marketing digital. Quijije, Elizabeth del Carmen Calderón. julio-septiembre 2016, Revista Contribuciones a la Economía .