Universidad de las ciencias Informáticas

Facultad 2

Sistema de Gestión de Información para la Dirección de Seguridad Informática de la UCI

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

Autores:

Fernando Paciencia Luteiro Palaia

Maray Montano Oliva

Tutores:

Ing. Oscar Lázaro Garcés Pérez

Ing. Mariana Thania Leal Rondon

Ing. Néstor Delgado

La Habana, Cuba

Junio, 2020

“Año 61 de la Revolución”

Declaro por este medio que yo **Fernando Paciencia Luteiro Palaia**, con carné de identidad **94111100901** y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo. Para que así conste firmamos la presente a los _____ días del mes de junio de 2020.

_____

Fernando Paciencia LuteiroPalaia

Autor

_____ _____

Ing. Óscar Lázaro Garcés Pérez **Ing. Néstor Delgado**

Tutor Tutor

Declaro por este medio que yo **Maray Montano Oliva**, con carné de identidad **97120309373** y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo. Para que así conste firmamos la presente a los _____ días del mes de junio de 2020.

_____

Maray Montano Oliva

Autor

_____ _____

Ing. Óscar Lázaro Garcés Pérez **Ing. Néstor Delgado**

Tutor Tutor

Quiero agradecer, en un principio, a todas aquellas personas que de una forma u otra me apoyaron incondicionalmente en estos cinco años de carrera.

A todos mis profesores en especial Raúl, Roberto, Madelín, René, Vladimir, que me ayudaron atravesar este camino lleno de obstáculos.

En ese sentido le agradezco a mi tribunal de tesis pues sus sugerencia me sirvieron de guía para lograr un buen trabajo de diploma y a mi oponente Mónica por responder a mis intensas preguntas en estos tiempo de COVID-19.

A mis tutores Óscar y Néstor les agradezco por haberme brindado la oportunidad de recurrir a sus conocimientos, por su dedicación y motivación .A Mariana mi cotutora quiero agradecerle por cada detalle y momento que me atendió aún cuando la distancia no lo permitía, quiero agradecerle por la claridad y exactitud con la que me guio.

A todos aquellas personas, amigos y compañeros del 2501 que de una forma u otra me apoyaron y estuvieron en mis andanzas que han colaborado de alguna manera para que se dé este momento tan especial, ya sea con un consejo en el grupo de Whatsapp, una ayuda, acompañando, incluso regañándome por mi despiste y guiando mis pasos, porque sé que nos imaginábamos un fin de curso muy diferente al que estamos atravesando pero tengo la certeza que cuando salgamos de esta sabremos compensar todo este tiempo sin vernos y la pasaremos en grande, porque es lo que nosotros hacemos! Porque se están ahí a pesar de todo y que puedo contar con ustedes Deborah, Cecilia, Eliany, Elizabeth, Yudith, Abel, Daniel, Erick, Mauricio, Damián, Claudia, Andres, Migue la enciclopedia, Rachel, Reinier, Shakira, José Luis, Kenny, Paulo, Álvaro, los quiero muchos chicos y les deseo todo lo bueno del mundo en el ámbito personal y profesional.

A mi otra familia, uno de los mayores tesoros que una persona puede encontrar durante su vida son los amigos, se vuelven tus hermanos de diferentes madres y padres, esa familia que la vida te da la oportunidad de ir creando con el pasar de los años. Me han animado, apoyado, inspirado y aguantado, son los mejores amigos que nadie haya tenido jamás. A Jennifer "Mivi" por siempre esa hermana que confiar en mí, por darme la fuerza y la capacidad de creer en mí misma, por

ser ese ángel de la guarda que siempre me regaña y a la vez me alaga y hacerme reír cuando estoy triste. A mi flacucha Leidi por ser mi amiga y hermana desde que tengo uso de razón y porque a pesar que en todo este tiempo estuvimos más que separadas me apoyaste en todo y me alentaste cómo pudiste.

A mi Jesús, gracias porque sin ti no se que hubiera sido de mi en la escuela, porque con solo mirarnos ya nos entendemos, porque somos los mejores cómplices del mundo, porque no existe nadie en la escuela que cuando le digan mi nombre no diga que es la que anda con el alto rubio, por ser mi amigo, hermano, cocinero, peluquero, padre y madre, no tengo palabras para poder agradecerte y para decirte lo que significas para mi, sin más eres mi Jesu para toda la vida.

A mis compinches, Deborah, Danet, Cecilia y Kamila el mejor regalo que me ha podido dar la UCI, son lo mejor que ha pasado en estos 5 años, agradecerles por cada momento de felicidad en mi vida, por esas obras de teatro que nos montábamos en un dos por tres, por las locuras de Danet que al final terminábamos haciéndolas también, por las intensidades de Deborah que al principio la dabas por loca pero después ella misma se encargaba de que le hicieras caso y que sabemos que lo hacía todo por nuestro bien, por tener ese gran corazón que no le cabe en el pecho y siempre estar conmigo para todo, eras como mi mama en la UCI, por Cecilia y sus consejos de cocina, por las horas de charla en la salita sentadas hablando de lo que sea y por haber traído a Cristian al grupo que nos saco de muchos apuros, por Kamila con sus actuaciones que no sabías si eran de verdad o mentira, por acompañarme y seguirme en todas mis boberías, a todas ustedes que sepan que las tengo en mi corazón y que eso nunca va a cambiar.

A Marquitos, José y Guillermo por ser mis amigos varones y darme los mejores consejos del mundo, Marcos por escucharme las mismas cosas una y otra vez sin detenerme, José por contar conmigo para todo y tratarme como su hermana, Guille por ser esa amistad que no me esperaba, porque no sé exactamente cuándo es que comenzó pero llegaste sin querer y para quedarte, porque te convertiste en alguien muy especial que me enseñó a querer la Tormenta aun sabiendo que

después no precisamente vendría la calma, por enseñarme a reescribir las estrellas sin alcanzarlas y por darme fuerza y ánimo a cumplir mis sueños.

A Amanda por ser de esas amistades que llegan sin esperarlo y para quedarse, en poco tiempo se ha vuelto mi consejera, mi confidente, mi amiga. Muchas gracias por enseñarme que la amistad va más allá del tiempo que conoces a alguien, de la distancia o las veces que se ven y de las personas que lo rodean, somos esa excepción y estoy muy contenta por haber ido ese día a acompañar a Deborah al dentista.

A Laura por ser esa amiga que conoces de la infancia y crecer y hacernos mujeres juntas, porque a pesar de todo este tiempo y que conocimos nuevas amistades estamos ahí la una para la otra sin importar Universidad, ni distancia.

No ha sido sencillo el camino hasta ahora, pero gracias a sus aportes, a su amor, a su inmensa bondad y apoyo, lo complicado de lograr esta meta se ha notado menos gracias a todos ustedes.

Les agradezco, y hago presente mi gran afecto hacia mi hermosa familia.

En especial a mi tía Mariela, gracias por ser como otra madre para mí, cuidarme, amarme, guiarme y apoyarme en todas mis decisiones.

A mis primos Frank y Fabián por ser como mis hermanos y apoyarme en todas mis boberías.

A Lila y a Porto por ser mis papas de la Habana, por recibirme es su casa con las puertas abiertas y tratarme como a una hija, por siempre poder contar con sus consejos, amor y apoyo en todas mis andanzas.

A mis abuelas Teresa y Luisa por dejarme disfrutar la vida a su lado y llenarme de su experiencia, por creer en mí, y por siempre apoyarme con lo mínimo que fuera para llevarme para la escuela.

A mi hermana Marien por ser mi ejemplo a seguir y la mejor hermana que pudiera pedir, por confiar y creer en mí no solo en este trabajo de diploma sino en todas las cosas que me he propuesto, por guiarme para no caer en sus errores, por hacerme tía y darme a la sobrina más linda que pudiera desear,

gracias por ser esa persona tan noble que eres y tratarme como una amiga a pesar de nuestra diferencia de edad, gracias por todo tata.

A mi padre, por el valor y el coraje que ha tenido para mantener nuestra familia, por las enseñanzas que me ha dado, por la confianza para hacer todo lo que me proponga y por darme ánimos, porque has sido y serás siempre un ejemplo incuestionable de fortaleza e integridad, porque sé el sacrificio que han hecho durante estos 5 años para poder estar yo ahora aquí frente a ustedes, en fin por amarme y cuidarme hasta hoy, gracias, papá.

A mi madre por el gran amor y devoción que tienes a tus dos hijas, por el apoyo ilimitado e incondicional que siempre me has dado, por tener siempre la fortaleza de salir adelante sin importar los obstáculos, por haberme formado como un mujer de bien, por ser una amiga, confidente, todo, por ser la mujer que me dio la vida y me enseñó a vivirla, por ser la fiel escudera de mi papá en sacrificarse para ser quien soy hoy, por ser esa mujer que se enoja conmigo y al rato va riéndose a contarme algo, por obligarme a aprender cosas que no pensaba importantes y al tiempo sin decirte nada, te doy la razón, no hay palabras en este mundo para agradecerte por todo lo que has hecho por mí, gracias mamá.

A todos ustedes, con todo mi corazón.

Maray Montano Oliva

Primeramente, agradezco a Dios todo poderoso por su infinita misericordia, por su amor inagotable, por su fidelidad durante estos años, a él sea honra y la gloría.

A mis profesores por todo el conocimiento que me han transmitido durante la carrera para convertirme hoy un ingeniero en ciencias Informáticas.

A mis tutores por la paciencia, los consejos y regaños. A la universidad de ciencias Informáticas quiero darles mil gracias por recibirnos calorosamente desde el primer día que hemos llego al país en 2014, ustedes han sido una familia, gracias por la atención diferenciada que hemos recibido desde la atención en la beca hasta la docencia. La UCI ha sido y siempre será mi segunda casa, mi segunda familia, aquí he formado mi carácter, mi personalidad, he llorado y me he reído.

A mi papá, mi súper héroe quería tanto verle sentado en las primeras sillas, mirándome y llenándose se orgullo y poder ver con sus ojos todo lo que ha invertido en mí, él me ha dado hasta lo que él no tenía, me podría quizás faltar un par de zapatos, pero nunca un libro u otro material escolar, mi mayor inspiración.

A mis compañeros del aula, aquella brigada que me acompañaba todos los días al docente, en las canchas de futbol, a ellos quiero agradecer por el compañerismo.

A mis queridos compatriotas angolanos, mis compañeros de batalla, muchos de ellos se han convertido mi familia, personas que llevaré por toda la vida.

A los amigos que gané fuera de la UCI, sin ellos tampoco sería posible llegar hasta aquí, a todos que de una u otra manera me ayudaron a concretizar ese sueño que ahora es una realidad. A Luyana Luciano por ser mi bálsamo de verdad, te conocí en poco tiempo y poco tiempo ha sido suficiente para convertirte en una la de las personas más especiales en mi vida, gracias por tu amistad. A esa flor Florinda por ser esa amiga y hermana a la vez, muchas gracias por ser parte de mi otra familia sin sangre, pero familia.

A todos ustedes, muchas gracias.

Fernando Paciencia Luteiro Palaia

A mi hermana Marien por ser mi modelo a seguir y apoyarme en todo.

A mis primos Frank, Fabián y Manuel y mi sobrina Maily para que le sirva de ejemplo.

A mis padres por ser los principales promotores de mis sueños, gracias a ellos por cada día confiar y creer en mí y en mis expectativas.

A ustedes, con todo mi corazón.

Maray Montano Oliva

A mis padres, que se han sacrificado y
han dado todo por mí, a ellos siempre les voy a dedicar
cada logro en mi vida.

A mi hermana mayor que para mí siempre
ha sido una segunda madre, una mujer que tuvo aprender
a luchar desde muy temprano, una eterna guerrera.

A mi familia que ha sido mi refugio, mi puerto seguro.

A todos que de alguna u otra manera me han apoyado,
a ustedes dedico cada párrafo que escribo.

Fernando Paciencia Luteiro Palaia

Tecnologías de la Información y las Comunicaciones (TIC) garantizan muchos beneficios de colaboración y gestión de los recursos digitales en las instituciones, pero también supone riesgos que afectan la seguridad de los sistemas informáticos. Una de las áreas más importantes para enfrentar los incidentes de seguridad constituyen los departamentos especializados en ciberseguridad. La Universidad de las Ciencias Informáticas cuenta con la Dirección de Seguridad Informática (DSI) encargada de proteger los activos digitales y procesar las evidencias generadas a partir de las investigaciones forenses que realizan. La DSI no cuenta con un sistema capaz de almacenar de manera centralizada las evidencias digitales de los controles de seguridad que realizan lo que dificulta el trabajo de los especialistas en los departamentos que componen el área. En este documento se presentan los resultados de una investigación para el desarrollo e implementación de un Sistema de Gestión de Información para la Dirección de Seguridad Informática de la UCI, que contribuya a mejorar los procesos de gestión de las evidencias digitales generadas en el área. Se realizó un estudio de los procesos de gestión de información de seguridad informática utilizando principalmente las normas ISOs 27001 y 27002 para garantizar la integridad, confidencialidad y autenticidad de los datos procesados. Se emplearon tecnologías de código abierto y como base en el proceso de desarrollo la metodología Proceso Unificado Ágil (variación UCI), de acuerdo con las políticas de informatización de la Universidad. Se presentan los diagramas de ingeniería de software necesarios para el diseño de la solución, la implementación del sistema y los resultados de la realización de las pruebas de aceptación y penetración al sistema.

Palabras claves: sistema, gestión de información, seguridad informática

Information and Communication Technologies (ICT) guarantee many benefits of collaboration and management of digital resources in institutions, but they also pose risks that affect the security of computer systems. One of the most important areas to deal with security incidents is the specialized cybersecurity departments. The University of Informatics Sciences has the Information Security Directorate (DSI) in charge of protecting digital assets and processing the evidence generated from the forensic investigations they carry out. The DSI does not have a system capable of centrally storing the digital evidence of the security controls they carry out, which hinders the work of specialists in the departments that make up the area. This document presents the results of an investigation for the development and implementation of an Information Management System for the UCI's Information Security Department, which contributes to improving the management processes of the digital evidence generated in the area. A study of the information security information management processes was carried out, mainly using ISOs 27001 and 27002 standards to guarantee the integrity, confidentiality and authenticity of the processed data. Open source technologies were used and as a basis in the development process the Agile Unified Process methodology (UCI variation), in accordance with the University's computerization policies. The software engineering diagrams necessary for the design of the solution, the implementation of the system and the results of carrying out the acceptance tests and penetration of the system are presented.

Keywords: System, information management, informatic security

En la actualidad se puede apreciar una marcada transformación social provocada por la aparición de modernas tecnologías portadoras de soluciones que afectan radicalmente el quehacer de las personas.
A partir de estos avances científicos producidos en los ámbitos de la informática y las telecomunicaciones surge las Tecnologías de la Información y las Comunicaciones (TIC) definida como el conjunto de tecnologías que permiten el acceso, producción, tratamiento y comunicación de información presentada en diferentes códigos tales como texto, imagen, sonido (Belloch Ortí, 2014).
La información y los procesos que se llevan a cabo en las entidades y la información que gestionan asumen numerosos riesgos. Como consecuencia se deben emplear medidas de control sobre la seguridad de la información y activos informáticos que involucre a todo el personal, desde la alta dirección hasta los operarios de los sistemas. Los usuarios en la actualidad tienen la facilidad de acceder a la información digital utilizando equipos de cómputo, aplicaciones especializadas, conexiones de redes y canales de comunicación que le permite realizar tareas cotidianas como trabajar, estudiar y comunicarse con su familia. Esta tendencia sobre el uso de las TIC define la necesidad de proteger los datos procesados con el fin de garantizan la confidencialidad, integridad y disponibilidad de los sistemas informáticos. Las amenazas y las vulnerabilidades que afectan a los activos de informáticos, involucran desde el desarrollo de las actividades de los empleados hasta los mecanismos de acceso a los mismos. Por esa razón nuestro país es sometido a marcos regulatorios relacionados con la Seguridad Informática además de contar con estándares, modelos y normas internacionales que a través de una serie de mejores prácticas aseguran una adecuada gestión de la seguridad de la información. Una de las normas más reconocidas es la ISO/IEC 27001:2013 que establece las guías, procedimientos y procesos para gestionarla apropiadamente mediante un proceso de mejoramiento continuo.

La información es un recurso de vital importancia para las instituciones contemporáneas. La economía de los países se sustenta en el procesamiento de la información digital generada dentro de sus propias entidades. La visibilidad de los datos sensibles, dentro de una institución debe ser estrictamente controlada a través de mecanismos de seguridad estandarizados.

Los Sistemas de Información (SI) dan la posibilidad de organizar, actualizar y utilizar datos; para que estos sean lo más exactos posibles, evitando redundancias con la información y logrando que esta llegue a la persona correcta, de la manera indicada y en el momento preciso. (Neisy Milagros Arias Santana, 2013).

A nivel internacional se pueden apreciar datos estadísticos de incidentes que afectan a la información, almacenamiento de contraseñas de formato recuperable, información sensible no eliminada, entre otras mostradas en la figura 1.1. (Luis M Pérez 2019)

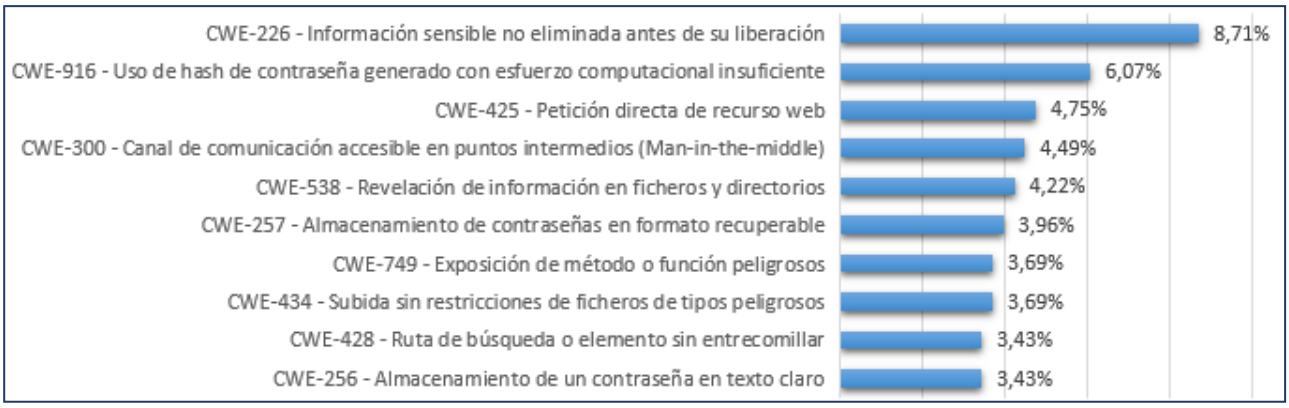


Figura 1.1 Incidentes que afectan a la información (Luis M Pérez 2019)

Teniendo en cuenta que las TIC y los riesgos de seguridad se desarrollan simultáneamente y que las entidades cubanas no están exentas de presentar incidentes de seguridad, el estado cubano procedió a la creación de la Oficina de Seguridad para las Redes Informáticas (OSRI).

La OSRI constituye la entidad rectora a nivel nacional que tiene como objetivo principal prevenir, evaluar, investigar y dar respuesta a las acciones tanto internas como externas que afecten el funcionamiento de las TIC en el país. Fue creada por el Comité Ejecutivo del Consejo de Ministros (Pérez del Cerro y D. Proenza-Pupo 2015)

Según las resoluciones emitidas a nivel nacional relacionadas con la seguridad en las TIC cada institución debe establecer un sistema de seguridad informática que permita identificar y mitigar los riesgos críticos que afecten a la información digital que gestionan. Nuestro país cuenta además con la Universidad de las Ciencias Informáticas (UCI) la cual contiene un departamento de la Dirección de Seguridad Informática (DSI) encargada de gestionar los procesos que sustentan la infraestructura para la protección de la información digital y los servicios telemáticos de la entidad. Con el objetivo de perfeccionar los procesos de la DSI se comenzaron a realizar un conjunto de acciones organizativas que contribuyen a la

implementación de la norma ISO 27001. Actualmente gracias a una entrevista realizada a los trabajadores del departamento se llegó a la conclusión que el DSI no cuenta con un sistema capaz de gestionar las evidencias relacionadas con el trabajo de los especialistas dando lugar a los siguientes problemas:

- Perdidas de datos significativos.
- Información dispersa.
- Gestión incorrecta de la evidencia digital.
- Complejidad en los procesos de análisis de información de seguridad informática.
- Trayectoria no definida para los usuarios implicados en incidentes de seguridad.

Teniendo en cuenta la situación problemática antes descrita, se identificó como **problema de investigación**: ¿Cómo centralizar las evidencias digitales generadas a partir de los procesos establecidos en la Dirección de Seguridad Informática de la UCI?
Se define como **objeto de estudio,** los sistemas de gestión de la información asociada a la seguridad informática y el **campo de acción** estará enmarcado en el control de evidencia digital de la Dirección de Seguridad Informática de la Universidad de Ciencias Informáticas.

Para dar solución al problema de investigación planteado, se define como **objetivo general**: desarrollar un sistema de gestión de información de seguridad informática que contribuye al control de la evidencia digital en la Dirección de Seguridad Informática de la UCI.

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

1. Analizar el marco teórico referencial de la presente investigación el cual responde a la difusión de la información.
2. Analizar las herramientas y tecnologías indispensables para la difusión de la información.
3. Realizar el análisis y diseño de la propuesta de solución para identificar los componentes de software que intervienen en su desarrollo.
4. Validar la solución desarrollada a partir de una estrategia de pruebas.

Se define como **preguntas de investigación**:

1. ¿Qué beneficios traería al implementar un SGSI para la Dirección de Seguridad Informática de la UCI?
2. ¿De qué forma la pérdida de datos significativos afectaría la gestión de las evidencias digitales?
3. ¿Por qué los demás sistemas homólogos no solucionan los problemas de gestión de información en la Dirección de Seguridad Informática de la UCI?

Métodos Teóricos

- **Histórico-Lógico**: Se emplea para evaluar la evolución del problema en los departamentos de la Dirección de Seguridad Informática.
- **Modelación:** Se emplea para modelar los diagramas, definir las relaciones entre los objetos que intervienen en los procesos implementados en la propuesta de solución.
- **Analítico-Sintético:** Se realiza un estudio teórico de la investigación facilitando el análisis de documentos y fuentes bibliográficas como las normas internacionales de la ISO 27001, la revisión de otros sistemas ya implementados para la extracción de los elementos más importantes acerca del proceso de desarrollo del sistema y de la gestión de la información.

Métodos Empíricos

- **Entrevista:** Es utilizado con el objetivo de definir y comprender las necesidades del cliente, capturar los requisitos correspondientes al sistema y obtener información que apoye la realización de la investigación, puesto que cada módulo del sistema se corresponde a un determinado departamento de la Dirección de Seguridad Informática.
- **Observación:** Se utiliza a través del estudio realizado en la dirección de seguridad informática, con el objetivo de comprender los procesos de ahí se desarrollan en la actualidad.

El presente documento consta de tres capítulos:

- **Capítulo1** “Fundamentación teórica sobre el sistema de gestión de información de seguridad informática”: consiste en llevar a cabo la fundamentación teórica del tema a investigar a partir de un estudio del estado del arte a nivel nacional e internacional teniendo en cuenta las tendencias actuales de los SGSI. Además, contendrá la fundamentación del uso de la metodología, tecnologías y herramientas escogidas para el desarrollo de la propuesta de investigación.
- **Capítulo 2** “Descripción del sistema de gestión de información de seguridad informática para la DSI”: contiene una caracterización de cómo será el sistema de gestión. En él se realiza un estudio desde la óptica de la ingeniería de software, englobando aspectos de importancia como la descripción general de la propuesta de solución, especificación de los requisitos de software, entre otros aspectos, para arribar a la conclusión de cómo es el sistema de gestión de información de seguridad informática para la DSI.
- **Capítulo 3** “Implementación y validación del sistema de gestión de información de seguridad informática para la DSI”: en este capítulo se especifican estándares de codificación utilizados durante el desarrollo de la propuesta de solución y se muestran segmentos de códigos de relevancia. Se define la estrategia de pruebas para validar las funcionalidades implementadas y se documentan los resultados obtenidos.

CAPÍTULO 1. Fundamentación Teórica.

En este capítulo se realiza el estudio de algunas soluciones del sistema de gestión de información de seguridad informática, para ello se tienen en cuenta características que permiten comprender su funcionamiento y cuánto aportan a la investigación en curso. Se abordan conceptos asociados a la gestión de información basados en las normas internacionales ISOs 27001 y 27002. Se describen brevemente la metodología, las herramientas, tecnologías y los lenguajes de programación que se utilizan en el desarrollo del sistema.

1.1 Marco Teórico

Para lograr una mejor comprensión de la presente investigación se recogen a continuación un conjunto de conceptos asociados al objeto de estudio.

1.1.1 La Gestión de Información

Es la denominación convencional de un conjunto de procesos por los cueles se controla el ciclo de vida de la información, desde su obtención hasta su disposición final. Tales procesos también comprenden la extracción, combinación, depuración y distribución de la información a los interesados. Responde a la metodología que marca la aparición del conocimiento luego del procesamiento de datos recuperados en una investigación científica. El objetivo de la Gestión de la Información es garantizar la integralidad, disponibilidad y confidencialidad de la información. (Dr. César Antonio González Horruitiner, 2017)

Gestionar información: Es ir en busca de nuevos significados, es el proceso por el cual se garantiza el camino del saber, del aprendizaje, del conocimiento, en estos procesos se incluye como aspecto determinante del Saber el “Saber Dónde”, que es el eslabón perdido para la consolidación de la teoría del Colectivismo como Teoría del Aprendizaje.

1.1.2 Seguridad informática.

Seguridad informática: La seguridad informática está relacionada con las metodologías, procesos y procedimientos para mantener la salvaguarda de la información y los datos confidenciales de una organización, al interior de los sistemas informáticos. Los procesos se estructuran con el uso de

estándares, normas, protocolos y metodologías para mitigar y minimizar los riesgos asociados a la infraestructura tecnológica.

1.1.3 Gestión de la Seguridad de la Informática:

La seguridad de la información es la preservación de la confidencialidad, integridad y disponibilidad de la información.(BELLOCH ORTÍ, 2014) Esto se logra mediante la implantación de un grupo de controles que incluyen políticas, procedimientos, estructuras organizativas y sistemas de hardware y software.(Romero Y.F 2012) Los atributos de la información previamente mencionados poseen las siguientes definiciones consideradas los pilares de la seguridad informática (BELLOCH ORTÍ, 2014)

- La **Confidencialidad:** a veces denominada secreto o privacidad, es la condición que asegura que la información no pueda estar disponible o ser descubierta por personas, entidades o procesos no autorizados.
- Se entiende por **Integridad:** la condición que garantiza que la información solo puede ser modificada, incluyendo su creación y borrado, por el personal no autorizado. Garantiza que la información sea exacta y completa y que el sistema no modifique o corrompa la información o permita que alguien no autorizado lo haga.
- **Disponibilidad:** propiedad que garantiza el acceso a los activos de información y el empleo de los recursos informáticos en cualquier momento por las personas autorizadas. Un sistema seguro debe mantener la información disponible para los usuarios. Disponibilidad significa que el sistema, tanto hardware como software, funciona de forma eficiente y que es capaz de recuperarse rápidamente en caso de fallo. (María Josefina Vidal Ledo, 2016)

A pesar de que pueden existir diferentes enfoques con respecto a las definiciones de “seguridad de la información”, “seguridad informática” y “seguridad de las tecnologías de la información (TI)”, en el presente trabajo se considerarán equivalentes estos términos, tal y como se plantea en la Resolución 128/2019 del Ministerio de la Informática y las Comunicaciones de Cuba (MIC).

1.1.4 Principales estándares y regulaciones sobre seguridad informática

En el contexto de la seguridad informática existen estándares que constituyen normas certificables, marcos de trabajo que representan una recopilación de mejores prácticas y regulaciones que son de obligatorio cumplimiento en determinadas naciones.

Estándares ISO/ 27001

La Organización Internacional para la Estandarización (ISO) y la Comisión Electrotécnica Internacional (IEC) han elaborado un grupo de estándares que constituyen una referencia a nivel mundial en la temática de seguridad de la información. La serie ISO/IEC 27000 está destinada completamente a la seguridad informática, donde los estándares ISO/IEC 27001 y 27002 abordan el tema de manera general, y el resto tratan temas específicos que complementan a los anteriores. El proceso de gestión de la seguridad informática se encuentra descrito en el estándar ISO/IEC 27001, el cual constituye una norma certificable a nivel internacional. Esta norma ofrece un modelo para el establecimiento, implementación, operación, monitorización, revisión, mantenimiento y mejora de un sistema de gestión de la seguridad de la información (SGSI). Se plantea la utilización del modelo PDCA (Plan - planificar, Do - hacer, Check – verificar, Act - actuar) (ISO 27001, 2018) (ISO 27002, 2018) para llevar a cabo estos objetivos (Figura 1.2).

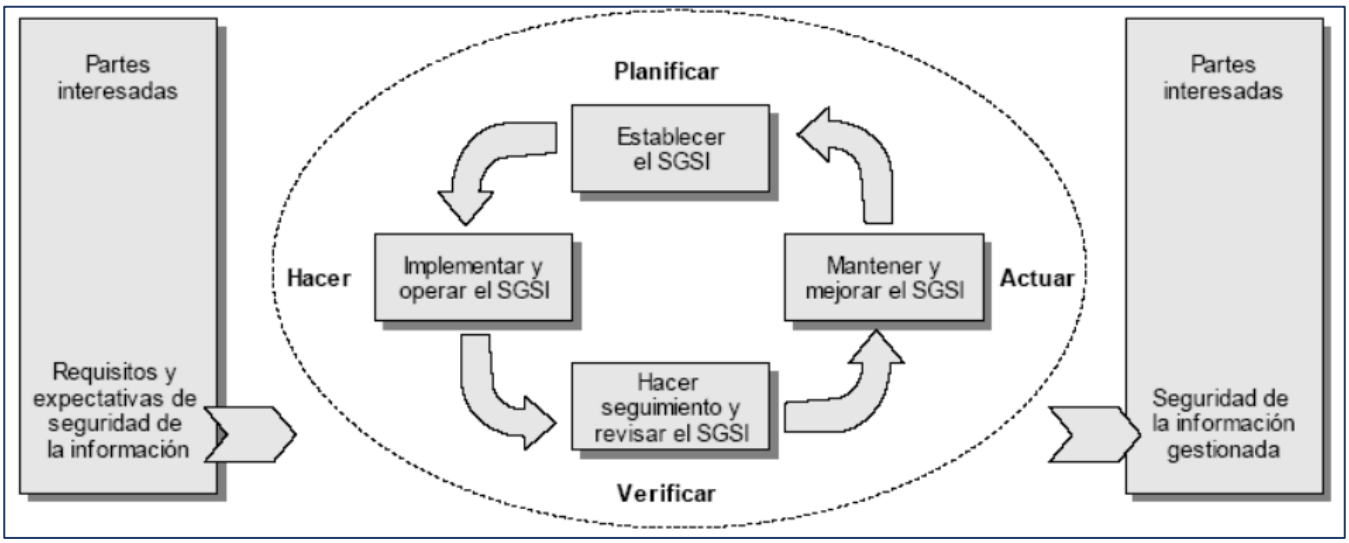


Figura 1.2 Modelo PDCA (ISO 27001, 2018)

1.1.5 Marco regulatorio

Resolución 128 - 2019: Reglamento de Seguridad de las Tecnologías de la Información y la Comunicación. Este reglamento tiene como objetivo complementar las disposiciones del Decreto 360 "Sobre la seguridad de las Tecnologías y la Comunicación y la Defensa del Ciberespacio Nacional" y establecer las funciones de los sujetos que intervienen en esta, así como garantizar un respaldo legal que responda a las condiciones y necesidades del proceso de informatización del país. (Jorge Luis Perdomo Di- Lell, 2019])

Resolución 129 – 2019: Metodología para la Gestión de la Seguridad Informática. Esta metodología tiene por objetivo determinar las acciones a realizar en una entidad durante el diseño, la implementación y posterior operación de un Sistema de Gestión de la Seguridad Informática, en lo adelante SGSI compuesta por dos partes, la primera se dedica al SGSI y la segunda a la estructura y contenido del Plan de Seguridad Informática. Constituye un complemento a lo exigido en el Decreto de Seguridad de las Tecnologías de la Información y la Comunicación y la Defensa del Ciberespacio Nacional y Reglamento de Seguridad para las Tecnologías de la Información y la Comunicación en cuanto a la obligación de diseñar, implementar y mantener actualizado un Sistema de Seguridad Informática, a partir de los bienes a proteger y de los riesgos a que están sometidos. (Jorge Luis Perdomo Di- Lella, 2019)

Decreto Ley 370-2018: Informatización de la sociedad de Cuba. Tiene como objetivo este Decreto-Ley el de fortalecer el proceso de informatización, en función de modernizar coherentemente todas las esferas de la sociedad y contribuir al desarrollo económico y social del país, consolidar el uso y desarrollo de las TIC, como instrumento para la defensa de la Revolución, promover y favorecer el acceso y el uso responsable de los ciudadanos a las TIC, consolidar la defensa política y la ciberseguridad frente a las amenazas los ataques y riesgos de todo tipo, preservar y desarrollar los recursos humanos asociados a la actividad, satisfacer las necesidades generales para incrementar el uso de las TIC y su aplicación por el Estado y el Gobierno, en la Seguridad y Defensa Nacional y el Orden Interior. (Miguel Díaz-Canel Bermúdez, 2018)

Para complementar la aplicación de estas Resoluciones, nuestro país se encuentra realizando actividades de capacitación y formación de especialistas en ciberseguridad. Además de que se encuentra en proceso

otra norma para tipificar los delitos informáticos en Cuba: Resolución para tipificar delitos Informáticos en Cuba.

1.2 Estudio de soluciones homólogas

Con el avance de las tecnologías, la información se ha convertido en el activo más importante de las organizaciones, es por ello que muchas de las empresas poseen sistemas para la gestión eficiente de sus informaciones. A partir de los problemas identificados en la dirección de seguridad informática, se decide hacer un estudio de los principales sistemas de gestión de información a nivel Internacional como nacional con la finalidad encontrar elementos comunes que ayuden el desarrollo del nuevo sistema de gestión de información.

A continuación, se presentan algunas de los sistemas de gestión de información, utilizados para hacer los estudios homólogos de ese trabajo de investigación.

1.2.1 Sistema de gestión de incidentes Internacionales

OTRS: Sistema de Solicitud de Tickets de Código Abierto

Es una herramienta de código abierto, la cual tiene como principales características: reducir los costos de licenciamiento y mejorar la satisfacción al estructurar la comunicación del servicio al cliente. Las funciones del negocio en OTRS ofrece las gestiones de los Ticket, que son verificadas por el sistema en cuanto a lectura de los mensajes, este análisis lleva a un ahorro importante en trabajo, tiempo y dinero; que es donde se centra todo el análisis de las empresas (Vieites 2011). Con el uso de OTRS se permite administrar diversas funcionalidades a los agentes tales como:

- Gestionar los incidentes reportados por los usuarios mediante la generación de un ticket.
- Dar seguimiento al requerimiento si aún no está resuelto para notificarle al cliente del estado del mismo.
- Cerrar los tickets generados.
- Generar estadísticas utilizando criterios de búsquedas especificados por el usuario.

Sistema de Gestión de Seguridad de la Información para la Secretaría de Economía y Empresas de menor tamaño en Chile

Cuentan con un conjunto de sistemas implementados mediante software libre que conforman el SGSI, que permite la gestión de la seguridad de la Información. Paralelo se implementan un conjunto de políticas y procedimientos de seguridad de la información en cumplimiento con los objetivos de control de la norma ISO27001:2013. Contiene un proceso de evaluación de riesgos de seguridad de la información que produce resultados consistentes y comparables entre las evaluaciones de riesgos realizadas año a año. De igual forma posee con un conjunto de indicadores que permitan evaluar la disminución de la ocurrencia y gravedad de los incidentes de seguridad de la información, como son: las pérdidas de información, ingresos no autorizados y la indisponibilidad de la información. (Martínez A. L 2018)

1.2.1. SGI Nacionales

Sistema para la Gestión de la Información de Seguridad Informática en la Universidad de Ciencias Médicas de Holguín

La universidad de Holguín cuenta con una herramienta de apoyo para la gestión de reportes de incidentes, control del estado de protección de los medios informáticos, así como la mejor preparación de los trabajadores en aspectos relacionados con la seguridad informática. Tiene como objetivo fundamental la resolución de un conjunto de dificultades en este aspecto, como son los relacionados con la fluidez de la información, la centralización y confiabilidad en sus datos, para mitigar estos perjuicios se desarrolló una herramienta informática que apoya la gestión de la información sobre seguridad informática.(Sánchez D.A 2018)

1.2.4. Resultados obtenidos del estudio de sistemas homólogos

Los sistemas de gestión de la Información estudiados nos brindaron información referente a las herramientas utilizadas y a sus objetivos específicos. Cada uno cuenta con funcionalidades que cubren necesidades específicas, en algunos casos son muy similares a las que se necesitan establecer pero no con todas las exigidas por nuestro cliente. OTRS no cuenta con la funcionalidad de registrar los resultados una de las funciones más importantes que se requieren, el SGI de Chile no cuenta con auditoría, ni estadísticas, ni nomencladores y por su parte el SGI de Holguín no cuenta con ninguna de las antes

mencionadas en el SGI de Chile además de no registrar resultados, haciéndose necesario el diseño e implementación de una solución propia para el departamento de seguridad informática.

1.3 Metodología, lenguajes, tecnologías y herramientas

Desarrollar un sistema de gestión de la información puede ser un trabajo complicado y muy laborioso si no se dispone de las herramientas adecuadas.

1.3.1. Metodología de desarrollo

Una metodología es un proceso. No existe una metodología de software universal, las características de cada proyecto exigen que el proceso sea configurable. Proporciona una guía para el orden de todas las actividades de un equipo de desarrollo de software. Se encarga de dirigir las tareas de cada desarrollador por separado y del equipo en conjunto. Especifica los artefactos que deben desarrollarse. Ofrece criterios para el control y la medición de los productos y las actividades del proyecto (ROSERO y M. DEL C. BENAVIDES, 2017).

AUP: El Proceso Unificado Ágil de *Scott Ambler o Agile Unified Process* (AUP en inglés) es una versión simplificada del Proceso Unificado Racional (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y concepto que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo:

- Desarrollo Dirigido por Pruebas (*Test Driven Development* - TDD en inglés).
- Modelado ágil.
- Gestión de cambios ágil.
- Refactorización de bases de datos para mejorar la productividad.

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo y recursos) exigiéndose así que el proceso sea configurable, en la UCI, por las características que presenta esta metodología y las posibilidades de adaptación al ciclo de vida de la actividad productiva de la institución, se decidió hacer una variación de la

misma. (ROSERO y M. DEL C. BENAVIDES, 2017)

Variación AUP-UCI

Una metodología de desarrollo de software tiene entre sus objetivos aumentar la calidad del software que se produce, de ahí la importancia de aplicar buenas prácticas. Para ello se busca apoyo en el Modelo CMMI-DEV v1.3, este constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora. Estas prácticas se centran en el desarrollo de productos y servicios de calidad (Nader-Ceballos, 2015)

Según Fernando (2016) “Con la adaptación de AUP que se propone para la actividad productiva de la UCI se logra estandarizar el proceso de desarrollo de software, dando cumplimiento además a las buenas prácticas que define CMMI-DEV v1.3. Se logra hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos”.

En el desarrollo del SGI se utilizó la metodología híbrida de desarrollo de software Variación de AUP para la UCI, una variante realizada por la Universidad de las Ciencias Informáticas a la metodología ágil AUP (Proceso Ágil Unificado) y está definida por la universidad como el documento rector de la actividad productiva y es la definida por el proyecto. Se decidió optar además por el escenario 4, el cual modela el sistema mediante historias de usuario.

Fases de Variación de AUP para la UCI: La metodología Variación de AUP para la UCI define tres (3) fases, (Inicio, Ejecución y Cierre) para el ciclo de vida de los proyectos de la universidad, las cuales resumen las características de las cuatro fases (Inicio, Elaboración, Construcción y Transición) propuestas en AUP. Después de determinar la metodología que guiará el proceso de desarrollo del SGI es necesario realizar un estudio de las herramientas a utilizar para el desarrollo de la propuesta de solución.

1.3.2 Marco de trabajo

En lenguaje informático, un Marco de trabajo es una plataforma de software universal y reutilizable para desarrollar aplicaciones de software, productos y soluciones. En otras palabras, podemos decir que es una especie de biblioteca, una pieza de software que proporciona a los desarrolladores web una base de código y formas consistentes y estandarizadas para crear aplicaciones web.

En los sistemas informáticos, un marco suele ser una estructura estratificada que indica qué tipo de programas pueden o deben construirse y cómo se interrelacionarán. Algunos marcos de sistemas informáticos también incluyen programas reales, especifican interfaces de programación u ofrecen herramientas de programación para usar los marcos. Un marco puede ser para un conjunto de funciones dentro de un sistema y cómo se interrelacionan; las capas de un sistema operativo; las capas de un subsistema de aplicación; cómo se debe estandarizar la comunicación en algún nivel de una red; Etcétera. Un marco generalmente es más completo que un protocolo y más prescriptivo que una estructura. (SYNERGY, 2018)

Laravel 5.8

Laravel es un *framework* de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Fue creado en 2011 y tiene una gran influencia de *frameworks* como *Ruby on Rails,*Sinatra y ASP.NET MVC.

Laravel tiene como objetivo ser un *framework* que permita el uso de una sintaxis profesional y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades. Intenta aprovechar lo mejor de otros *frameworks* y aprovechar las características de las últimas versiones de PHP.

Gran parte de Laravel está formado por dependencias, especialmente de Symfony, esto implica que el desarrollo de Laravel dependa también del desarrollo de sus dependencias.

Características

- Sistema de ruteo, también *RESTful*
- *Blade*, Motor de plantillas
- Peticiones *Fluent*
- *Eloquent* ORM
- Basado en *Composer*
- Soporte para el caché
- Soporte para MVC
- Usa componentes de Symfony
- Adopta las especificaciones PSR

No hay razón para instalar versiones anteriores de Laravel para nuevos proyectos, puesto que las diferentes versiones a partir de la versión Laravel 5.5 de Laravel son pequeñas. Laravel 6.0 es la versión LTS más reciente del *framework*.

Patrón MVC

Laravel propone en el desarrollo usar *'RouteswithClosures*', en lugar de un MVC tradicional con el objetivo de hacer el código más claro. Aun así permite el uso de MVC tradicional.

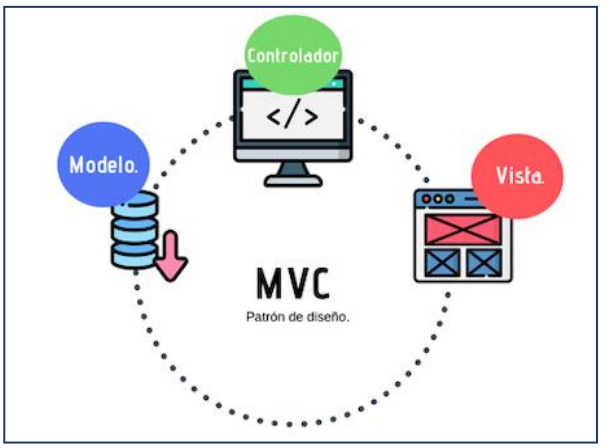


Figura 1.3 Patrón MVC (Elaboración propia)

Modelo

Laravel incluye un sistema de mapeo de datos relacional llamado *EloquentORM* que facilita la creación de modelos. Este ORM se funda en patrón active record y su funcionamiento es muy sencillo. Es opcional el uso de *Eloquent*, pues también dispone de otros recursos que nos facilitan interactuar con los datos, o específicamente la creación de modelos.

Vista

Laravel incluye de paquete un sistema de procesamiento de plantillas llamado *Blade*. Este sistema de plantillas favorece un código mucho más limpio en las Vistas, además de incluir un sistema de Caché que lo hace mucho más rápido. El sistema *Blade* de Laravel, permite una sintaxis mucho más reducida en su escritura.

Controlador

Los controladores contienen la lógica de la aplicación y permiten organizar el código en clases sin tener que escribirlo todo en las rutas. Todos los controladores deben extenderse de la clase *BaseController* (Isaura Luna Suárez, 2019)

3.3 Lenguaje para el modelado

El Lenguaje Unificado de Modelación (UML por sus siglas en inglés) fue creado para forjar un lenguaje de modelado visualmente común, semántica y sintácticamente agradable para la arquitectura, así como para el diseño y la implementación de sistemas de *software* complejos, tanto en estructura como en comportamiento (Solarte, 2018)

Es comparable a los planos usados en otros campos y consiste en diferentes tipos de diagramas. En general, los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contienen.

Es un lenguaje que se utiliza para visualizar, especificar, construir y documentar los artefactos de un sistema. Este permite la modelación de sistemas con tecnología orientada a objetos. Es importante el papel que juega UML, pues no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso (Kendall, Ivar Jacobson 2019).

Como parte del desarrollo de este trabajo se decide utilizar el lenguaje unificado de modelado en su versión 2.0 porque mejora los tiempos totales de desarrollo hasta en un 50 por ciento. Permite modelar sistemas utilizando conceptos orientados a objetos, encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica. Admite crear un lenguaje de modelado utilizado tanto por humanos como por máquinas. Es de gran ayuda pues mejora el soporte a la planeación y al control de proyectos. Ostenta una alta reutilización y minimización de costos (Solarte 2018).

1.3.4 Lenguajes de programación

Tecnologías y lenguajes del lado del cliente.

HTML5

(*Hyper Text Markup Language*, versión 5) es la quinta revisión del lenguaje HTML. Esta nueva versión de conjunto con CSS3 (*Cascade Style Sheets*), define los nuevos estándares de desarrollo web, rediseñando el código para resolver problemas y actualizándolo así a nuevas necesidades. No se limita solo a crear nuevas etiquetas o atributos, sino que incorpora características nuevas y proporciona una plataforma de desarrollo de complejas aplicaciones web (mediante los APIs). HTML5 está destinado a sustituir a HTML4. Esta versión nos permite una mayor interacción entre nuestras páginas web y el contenido media (video, audio, entre otros) así como una mayor facilidad a la hora de codificar nuestro diseño básico. (Jorge Fernández 2020)

Algunas de las nuevas características de HTML5 serían:

- Nuevas etiquetas semánticas para estructurar los documentos HTML, destinados a remplazar la necesidad de tener una etiqueta <*div*> que identifique cada bloque de la página.
- Los nuevos elementos multimedia como <audio> y <video>.
- La integración de gráficos vectoriales escalables (SVG) en sustitución de los genéricos <*object*>, y un nuevo elemento <*canvas*> que nos permite dibujar en él.
- El cambio, redefinición o estandarización de algunos elementos, como <a>, <cite> o <menú>.
- MathML para fórmulas matemáticas.
- Almacenamiento local en el lado del cliente.

Derivado de lo anterior, la compatibilidad con otros lenguajes y su facilidad de uso lo hace ideal para el desarrollo del SGI. Al incorporar etiquetas (*canvas* 2D y 3D, audio, video) para mostrar los contenidos multimedia, otras etiquetas para manejar grandes conjuntos de datos: *Datagrid, Details*, Menú y *Command*, permiten generar tablas dinámicas que pueden filtrar al incluir mejoras en los formularios y nuevos tipos de datos. (Jorge Fernández 2020)

CSS 3

Se decide utilizar como lenguaje de diseño gráfico porque CSS3 está dividido en varios documentos separados, llamados módulos, cada uno añade nuevas funcionalidades a las definidas en CSS2 de manera que preservan las anteriores para mantener la compatibilidad. Facilita la publicación de contenidos en múltiples formatos de presentación basado en parámetros nominales, estos incluyen preferencias explícitas del usuario: diferentes navegadores web, el tipo de dispositivo utilizado para ver el contenido (una PC o un Smartphone), la localización geográfica u otras variables. Entre otras ventajas que trae el uso del CSS están la consistencia del portal, ancho de banda, formateo de página y accesibilidad. Los *framework* CSS son bibliotecas preparadas para permitir la simplificación, y el mayor cumplimento de los estándares en los diseños de páginas web usando el lenguaje CSS. Algunos de los *framework* más comunes son *Foundation, Blueprint y Bootstrap*. (Carlos Fariñas, 2017)

Bootstrap

Es el *framework* de Twitter que permite crear interfaces web con CSS y JavaScript que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, automáticamente se adapta al tamaño de un ordenador o de una tablet sin que el usuario tenga que hacer nada, esto se denomina diseño adaptativo o *Responsive Design* (Lambert, 2016).Para la propuesta de solución se decide utilizar Bootstrap en la versión 4.1.3 pues permite simplificar el proceso de maquetación, sirviendo de guía para aplicar las buenas prácticas y los diferentes estándares. Admite utilizar muchos elementos web: desde iconos a desplegables, combinando HTML5, CSS y Javascript. El diseño va a ser adaptable, sin importar el dispositivo, la escala o resolución. Se integra muy bien con las principales librerías Javascript. (Evelio Morejon, 2020)

JavaScript (JS)

Es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa (Quijano, Cleto, & Stampella, 2019). Se resuelve usar dicho lenguaje en su versión 5.0 ya que agrega dinamismo a los

SGI, así como validación de campos y formularios. No es un lenguaje compilado si no, interpretado, lo hace ideal para ejecutarse en los navegadores actuales. El uso de variables, funciones y operadores es semejante a los demás lenguajes de programación de más alto nivel. JQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. Es la biblioteca de JavaScript más utilizada (Mario López, 2020).

JQuery

Es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. Al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. Por todo lo abordado anteriormente se decide utilizar como biblioteca multiplataforma: JQuery 1.10.2, porque concibe la manipulación de la hoja de estilo CSS, efectos y animaciones personalizadas. (Ernesto Sosa, 2019)

Lenguajes del lado del servidor

PHP

PHP es el acrónimo de *HypertextPreprocessor* es un lenguaje de código abierto interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Es un lenguaje de programación muy popular utilizado. Se distribuye bajo la licencia PHP, lo que hace posible reutilizar o modificar el código fuente sin costes adicionales. La función básica de PHP es leer datos de formularios y convertirlos en variables PHP. Posteriormente, las variables pueden, por ejemplo, ser introducidas en una base de datos o enviadas por correo electrónico. Las ventajas de PHP incluyen su integración con el protocolo de internet y un amplio soporte de diferentes modelos de bases de datos. La principal novedad de PHP 7 es el notable aumento en el rendimiento logrado por el equipo de desarrolladores, a partir de varias optimizaciones del núcleo del lenguaje. Además del aumento de la velocidad (PHP 7 es dos veces más rápido que su predecesor), el

nuevo PHP ocupa menos memoria (Daniel Suárez, 2020).

Otra característica del nuevo PHP son los datos escalares, es decir, aquellos que almacenan solo un valor, así como la definición de los tipos de devolución de códigos de programación. De esta manera, PHP ha sido extendido por tipos como *Integer, Boolean, Float y String.* Para tener un efecto positivo en el comportamiento de navegación de los usuarios, mejorar la velocidad de carga de una web, tener bajos requerimientos de capacidad de almacenamiento, ser capaz de detectar errores y prevenir los fallos del sistema se decide usar el lenguaje de programación PHP 7.0.32.

1.3.5 Servidor web

Apache: El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA Http 1.3, pero más tarde fue reescrito por completo. El enfoque principal de la actualización de Apache 2.4 es la mejora en el rendimiento que se sintetizan en un menor consumo de memoria y mejoras en las concurrencias de las peticiones. Es la versión más rápida de Apache. Los diferentes módulos de multiprocesador disponibles en Apache 2.4 permiten a los administradores de sistemas ajustar Apache para ser más rápido según las necesidades y la naturaleza de las peticiones que tenga que atender. Estos módulos pueden ser seleccionados en tiempo de ejecución con lo que añade una mayor flexibilidad. Incluso presume de tener un rendimiento superior a los servidores orientados a eventos. Los diferentes módulos de multiprocesador disponibles en Apache 2.4 permiten a los administradores de sistemas ajustar Apache para ser más rápido según las necesidades y la naturaleza de las peticiones que tenga que atender. Estos módulos pueden ser seleccionados en tiempo de ejecución con lo que añade una mayor flexibilidad. Incluso presume de tener un rendimiento superior a los servidores orientados a eventos (Gustavo B, 2020).

1.3.6 Gestor de bases de datos

Un Sistema Gestor de Bases de Datos (SGBD) o Data Base Management System (DBMS, por sus siglas en inglés) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el

usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. (Rodolfo Acosta, 2019)

Un SGBD debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

MySQL 10.4.8 es un servidor de bases de datos relacional, multihilo, multiplataforma y multiusuario. Es una idea originaria de la empresa *Open SourceMySQL* AB fundada en 1995, que pasó a manos de *Sun Microsystems* en 2008 cuando adquirió la empresa, luego en 2010 *Sun Microsystems* fue adquirida por la empresa *Oracle Corporation*, lo que justifica el desarrollo de MySQL como software libre en un esquema de licenciamiento dual. Por un lado, se ofrece bajo la GNU GPL 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. (Kendall, Gustavo B, 2019)

PhpMyAdmin 4.5.4.1 es una herramienta de software libre escrito en PHP, cuya intención es facilitar la administración de nuestro servidor MySQL a través de la web. Las operaciones más frecuentes del servidor (gestión de bases de datos, tablas, columnas, relaciones, índices, usuarios, permisos) se realizan a través de la interfaz de usuario, pero también nos ofrece la capacidad de ejecutar directamente cualquier sentencia SQL para las operaciones de mantenimiento y consulta de datos. PhpMyAdmin puede administrar un servidor MySQL completo con todas sus bases de datos (se necesita una cuenta de super-usuario o root), así como una base de datos única. Para este último caso se necesita configurar un usuario de MySQL que pueda leer / escribir sólo sobre la base de datos deseada. (Pedro Damián López, 2015)

1.3.7 Herramientas

Visual Paradigm es una herramienta CASE (Ingeniería de Software Asistida por Computadora o *ComputerAided Software Engineering*) multiplataforma utilizada para el modelado visual a través del UML. Soporta las últimas versiones de UML y permite realizar el modelado, la captura de requisitos, diseño de base de datos y el modelado de procesos de negocio. Ofrece al analista y desarrollador de software un ambiente para analizar, diseñar y mantener aplicaciones de software en una manera disciplinada. Además, aumenta la producción automática de código, bases de datos y generación de informes. (Sánchez Vera y Ocaña La O 2016)

Visual Paradigm para UML es apoyado por un conjunto de idiomas, tanto en la generación del código como en la ingeniería inversa, por mencionar algunos ejemplos los cuales tienen la capacidad de soporte tales como: Java, C + +, CORBA IDL, PHP, Ada y Python. Para maximizar la interoperabilidad de los productos de Visual Paradigm con otras aplicaciones se introdujo la importación y exportación de modelos de proyecto desde o hasta un formato XML. Los usuarios y proveedores de tecnología pueden integrar Visual Paradigm en cada uno de sus modelos para utilizarlos en sus soluciones con un mínimo esfuerzo (Rivera Velázquez, Sánchez Vera, & Ocaña La O, 2016).Proporciona una plataforma escalable para que los desarrolladores puedan agregar funciones al mismo, ellos pueden hacer referencia a los plugins de guiar el desarrollo, a construir sus propios *plugins* para leer, actualizar, recuperar y eliminar los diagramas y los elementos del modelo. (Sánchez Vera y Ocaña La O 2016)

1.4 Conclusiones Parciales

- La definición del marco teórico aportó los elementos que sustentan la solución del problema, así como desglosar los términos principales que se manejan en esta investigación.
- El estudio de las soluciones homologas nos permitió identificar las principales características que debe cumplir el SGSI, para su buen funcionamiento en la gestión correcta de la información.
- Establecer la metodología AUP-UCI como guía para el desarrollo y la base tecnológica lo cual permitió el análisis de herramientas y tecnologías existentes para el desarrollo de la solución propuesta; seleccionándose para el desarrollo, Laravel 5.8 como Framework y PHP 7 como lenguaje de programación.
- Como sistema gestor de bases de datos se escogió MySQL 5.7.24 por ser un sistema seguro, libre y potente, para la gestión de la base del SGSI.
- Como herramienta para el modelado de diagramas se seleccionó Visual Paradigm porque es una herramienta para el diseño de los múltiples artefactos necesarios para representar la información en las metodologías de desarrollo y ofrece diversas facilidades cuando se realizan los diagramas UML.

CAPÍTULO 2. Descripción del SGSI para la UCI

En el presente capítulo se describe la propuesta de solución para lograr mejorar la gestión de información en la Dirección de Seguridad Informática de la UCI. El objetivo del capítulo es presentar los componentes que definen la solución propuesta y explicar su funcionamiento y relación. Se definen las historias de usuarios y los artefactos necesarios que servirán de base para la fase de implementación. Se presentan, además, los requisitos funcionales y no funcionales.

2.1 Propuesta de solución

Utilizando la información recopilada en el capítulo precedente, se propone el desarrollo de un sistema de gestión de información de seguridad informática para UCI con la finalidad de mejorar la eficiencia en la gestión de información en la Dirección de Seguridad Informática. El desarrollo del SGSI permitirá a los especialistas de seguridad informática obtener evidencias digitales de los incidentes informáticos, evitando de esa forma la pérdida y la dispersión de los datos.

En la solución existirán módulos que representan cada uno de los departamentos de la dirección de seguridad informática, el acceso a estos módulos será restringido en función del rol a que pertenece un especialista. El sistema dispondrá de un tercer módulo de estadísticas que permitirá a los especialistas obtener informaciones genéricas de los incidentes informáticos a través de gráficos personalizados. También contará con un buscar para poder filtrar los datos de acorde a ciertos parámetros de entrada como usuarios implicados a un incidente y poder filtrar los datos de un área de supervisión especifico.

Estructura Sistema

El SGSI estará conformado por un conjunto de módulos que serán los diferentes departamentos de la dirección de seguridad informática. El acceso a cada uno de ellos lo realizarán los especialistas de seguridad informática, a los que se les asignarán roles que previamente tendrán permisos asignados.

Tabla 2.1 Arquitectura de información de la propuesta de solución (Elaboración propia)

2.1.1 Especificación de requisitos de software

(Pressman, 2010) establece que la tarea del análisis de requisitos es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refina en detalle el ámbito del software, y se crean modelos de los requisitos de datos, flujo de información y control, y del comportamiento operativo. Se analizan soluciones alternativas y se asignan a diferentes elementos del software. El análisis de requisitos permite al desarrollador o desarrolladores especificar la función y el rendimiento del software, indica la interfaz del software con otros elementos del sistema y establece las restricciones que debe cumplir el software.

Requisitos funcionales

Los requisitos funcionales son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse en situaciones específicas. En algunos casos, los requisitos funcionales también explican lo que el sistema no debe hacer (Sommerville, 2011).

Después del encuentro con el cliente, se obtuvo un total de treinta (30) requisitos funcionales, a los cuales se les asignó una prioridad y complejidad teniendo en cuenta la importancia fijada por el cliente a partir de sus necesidades y la dificultad de su posterior implementación. Los mismos se muestran en la siguiente tabla y se destaca el empleo del patrón CRUD completo para agrupar en un mismo requisito Gestionar.

Requisitos Funcionales (RF)			
No.	**Nombre**	**No.**	**Nombre**
1	Autenticar usuario	**26**	Obtener datos de usuarios por LDAP
2	Filtrar usuario	**27**	Enviar correo de forma manual y planificada
3,4,5,6	Gestionar rol	**28**	Diseñar sistema de alertas para las investigaciones atrasadas
7,8,9,10	Gestionar supervisión	**29**	Subir Informe digital
11,12,13,14	Gestionar Incidente	**30**	Generar Repórtes CSV y PDF
15,16,17,18	Gestionar prueba de penetración	**31**	
19	Filtrar incidente	**32**	
20	Mostrar estadísticas mensual de incidentes	**33**	
21	Filtrar información por criterios individuales	**34**	

Tabla 2.2 Requisitos Funcionales (Elaboración propia.)

Requisitos no funcionales

Los requisitos no funcionales (RnF) son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares. Los requisitos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del mismo (Sommerville, 2011).

Los autores de la presente investigación consideran a los requisitos no funcionales como requerimientos de calidad y para ellos se contemplarán las características que se evidencian en (Sommerville, 2011). Distribuidos en especificaciones de usabilidad, eficiencia, hardware, seguridad, software y legales se obtuvo un total de 18 requisitos no funcionales, los cuales se relacionan a continuación:

- **Usabilidad**

RnF 1: El sistema debe presentar una interfaz agradable e intuitiva para el usuario.
RnF 2: El sistema debe tener visibilidad en los principales navegadores web como Chrome v.45, Firefox v.70*, Safari v.9, Opera v.30.

RnF 3: El sistema podrá ser visualizado en dispositivos desde las resoluciones 320x480, 768x1024, 1024x980 y 1325x980.

- **Rendimiento**

RnF 4: El tiempo de demora del sistema en cada transición debe ser menor de ocho (8) segundos aproximadamente.

- ***Hardware***

RNF 5: El servidor de base de datos debe poseer una capacidad mínima de 20 GB.
RNF 6: El servidor de aplicaciones web debe poseer una capacidad mínima de 80 GB.
RNF 7: Los servidores web y de base de datos deben poseer como mínimo 1 GB de memoria RAM.

- **Seguridad**

RnF 8: En caso de que el sistema presente alguna falla, los errores deben mostrar la menor cantidad de detalles posible, de forma tal, que se evite dar información que comprometa la seguridad e integridad del sistema. Sólo se mostrarán detalles ampliados del error a usuarios con privilegios de administración.
RnF 9: Se asignarán los permisos de acceso, escritura, lectura en dependencia del rol que desempeñe cada usuario del sistema.
RnF 10: Se podrá acceder a las páginas de administración del sistema a través del protocolo HTTPS, utilizando certificados auto firmados generados por el servidor
RnF 11: Se garantizará la integridad de la información mediante mecanismos de control de acceso utilizando usuarios, contraseñas y niveles de accesos para cada usuario, de manera que cada uno pueda tener disponible solamente las opciones que se encuentran en correspondencia con su actividad.
RnF 12: El acceso a la aplicación debe ser restringida por el ip a través de las listas de acceso e iptables para evitar que usuarios dentro de la misma lo accedan sin pertenecer a la dirección de seguridad informática.

- ***Software***

RnF 13: Servidor Apache 2.4
RnF 14: Servidor de base de datos MySQL en su versión 8.0.19 o superior.
RnF 15: Lenguaje de programación PHP 7 como soporte Framework Laravel 6.

- **Legales**

RNF 16: Uso de licencia MITpara el Framework Laravel.

RNF 17: Uso de licencia BSD de MySQL.

RNF 18: Uso de licencia PHP *License*.

Descripción de requisitos de software (Historias de Usuario)

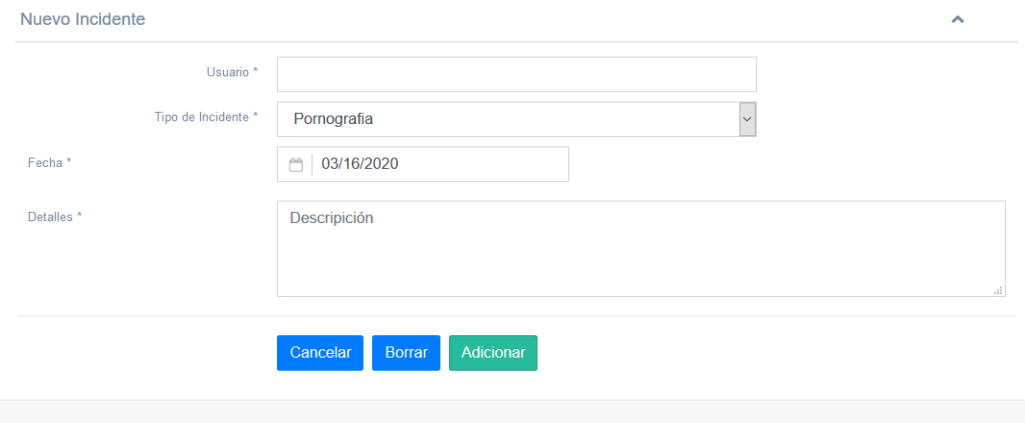
Número: 1. Nombre del requisito: Registrar Incidente.	
Programador: Fernando Paciencia Luteiro Palaia.	Iteración Asignada: 1ra
Prioridad: Alta	Tiempo Estimado: 24h
Riesgo en Desarrollo: Alta	Tiempo Real: 18h
Descripción: El sistema permite almacenar información referente a un incidente.	
Observaciones: En el caso registrar incidente el sistema permite que solo los usuarios con roles especialistas en incidente tengan el permiso de registrar un incidente. Se requieren de forma obligatoria los campos: nombre-usuario, fecha de incidencia, descripción, tipo de incidente.	
Prototipo elemental de interfaz gráfica de usuario: 	

Tabla 2.3 Registrar Incidente (Elaboración propia.)

2.2 Análisis y diseño

2.2.1 Diseño arquitectónico

Patrón arquitectónico Al utilizarse el *framework* Laravel para el desarrollo de la propuesta de solución, la arquitectura de software a utilizar es la definida por el mismo, la cual es una arquitectura modelo vista controlador, es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo (ROMERO, Y.F 2015).

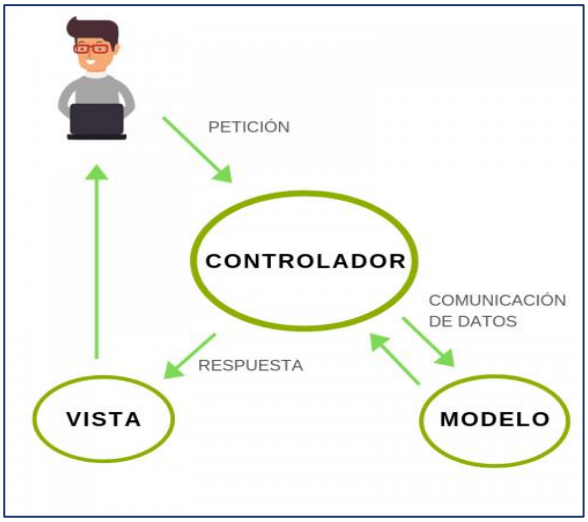


Figura 2.1 Arquitectura MVC (ROMERO, Y.F 2015)

El **Modelo** es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el modelo.

La **Vista** es el objeto que maneja la presentación visual de los datos representados por el modelo. Genera una representación visual del modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo.

El **Controlador** es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el modelo, centra toda la interacción entre la vista y el modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del modelo o por alteraciones de la vista. Interactúa con el Modelo a través de una referencia al propio modelo.

2.2.2 Modelado del diseño

Diagrama de clases del diseño con estereotipos web (DCD)

Según (Pressman, 2010) “Un diagrama de clases del diseño con estereotipos web tiene el mismo objetivo o propósito que un Diagrama de Clases tradicional, con la particularidad de que se emplea para el modelado de aplicaciones web”. El mismo es evidenciado en la siguiente figura:

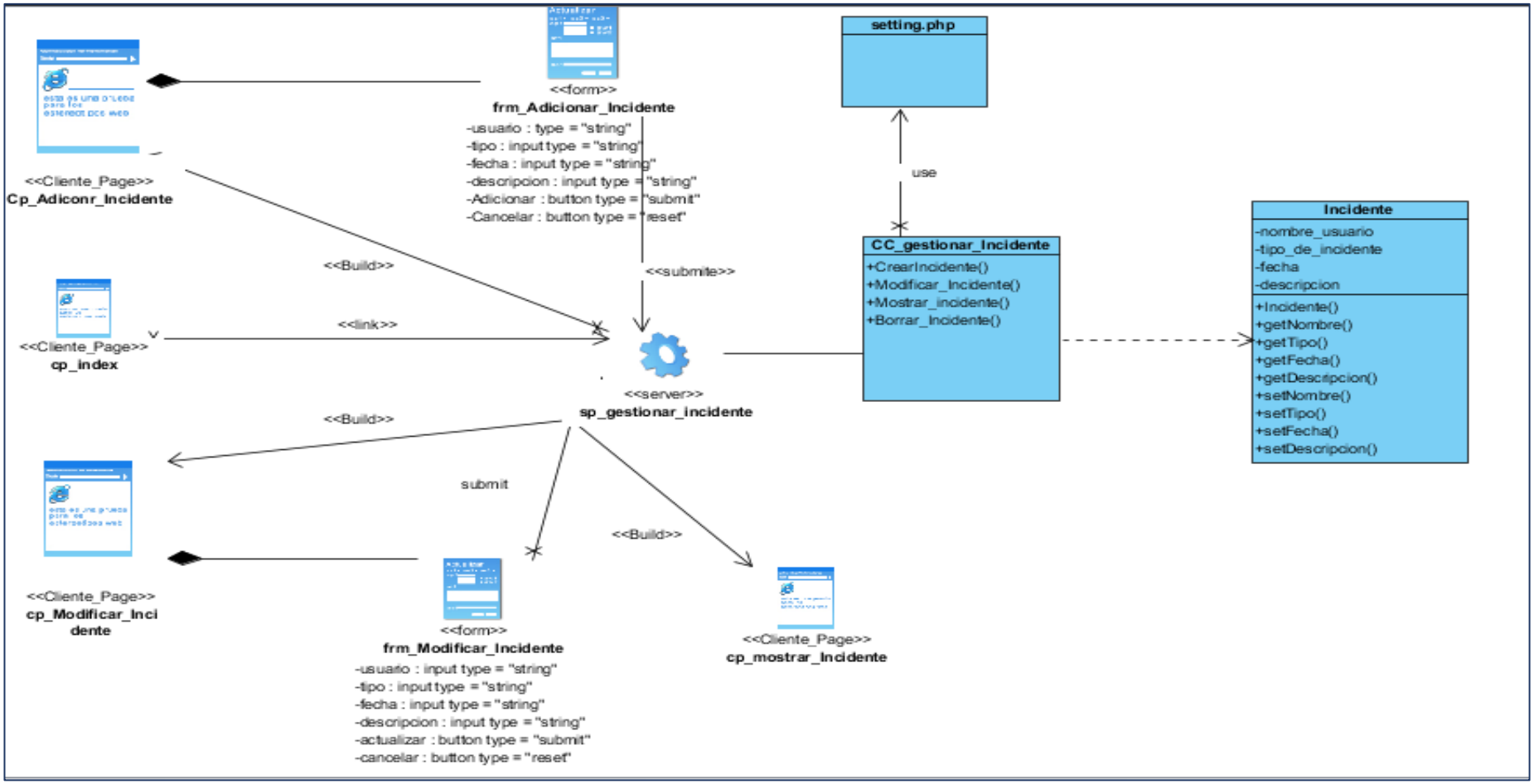


Figura 2.2 DCD con estereotipos web para Gestionar Incidente (Elaboración propia)

Para adicionar, modificar y mostrar un incidente en general la página servidora *sp_gestionar_incidente* construye las páginas adicionar incidentes que posee un formulario *frm_adicionar _incidentes* y este a su vez hace un *submit* de los datos insertados a la página servidora *sp_gestionar_incidente,* la *cp_modificar* que está compuesta por el formulario *frm_mostrar_incidente* que envía los datos a través de un *submit* a la página servidora, la *cp_mostrar_incidente* que muestra el listado de los incidentes una vez que añade un nuevo o cuando se modifica. Para acceder a los datos se hace a través de la clase controladora *CC_gestionar_Incidente.*

Patrones de diseño

Los patrones de diseño son un conjunto de prácticas de óptimo diseño que se utilizan para abordar problemas recurrentes en la programación orientada a objetos. El concepto de patrones de diseño fue el resultado de un trabajo realizado por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, publicado en 1995 en un libro titulado: “Patrones de diseño: Elementos de *software* orientado a objetos reutilizables” en el que se esbozaban 23 patrones de diseño. PHP utiliza patrones de diseño propiamente orientados a objetos como los patrones *Gang of Four* (GoF), permitiendo la construcción de un diseño elegante y robusto. Los patrones GoF son clasificados según el propósito para el cual han sido definidos. Laravel hace uso de patrones de diseño como: *Factory method* (Método de fabricación), *MVC Pattern* (Divide una aplicación en tres partes interconectadas, separando las partes internas de la representación), DAO (*Data Access Object*),Facade (Fachada).Un patrón de diseño resuelve problemas de diseño en desarrollo de aplicaciones informáticas, el cual, debe haber comprobado su efectividad resolviendo dichos problemas y demuestre que puede ser reutilizable (Antonio Leyva, 2016) Los patrones usados para el desarrollo del SGI se evidencian de la forma siguiente:

MVC Pattern este patrón tiene como objetivo abordar el problema de la *separación de conceptos,* es decir, definir de forma clara qué ha de hacer cada uno de los componentes que aparecen en nuestra aplicación. Esto por supuesto se acabará traduciendo en una mejor legibilidad y reutilización del código que se escribe. Laravel por defecto trae consigo una estructura que separa agrupa las carpetas en 3 categorías (MVC). Para el modelo laravel almacena los ficheros en el directorio *database->migrations*, en el caso de las vistas las paginas html o blade.php son almacenadas en la carpeta *resource->view* y para las clases controladores estas se encuentran en la carpeta *app->controllers*. (Jhon S. 2012)

Factory method este patrón permite crear objetos sin tener que especificar la clase exacta a la que han de pertenecer y sin tener que acceder directamente a la lógica de como ella ha sido creada. Este patrón se emplea en laravel de la siguiente manera: cuando se pretende hacer un *request* para acceder a los datos de un formulario al utilizar la clase *Request* uno accede directamente a los elementos del formulario, sino que forma indirecta un objeto de esa clase accede a cada uno de los elementos que componen dicho formulario y para hacer uso de ella el programador no tiene necesariamente que entender que hay por detrás de ella. (Eduardo García, 2014)

DAO (*Data Access Object*): este patrón permite separar la manipulación de datos y acceso a la DB de la lógica propia de la aplicación. Este patrón posee tres componentes principales:

- El acceso a los datos
- El objeto de transferencia de datos (*value object*)
- El cliente que consume esos datos

Este patrón en el sistema se pone en manifiesto cuando el usuario solicita una información y esto ocurre de la siguiente forma: el usuario manda una solicitud al sistema, el sistema a su vez accede a estos datos a través de clases que después se traducen en tablas dentro de la base de datos, una vez que estén disponible los datos el sistema envía a la vista del usuario para que consuma estos datos sin necesidad de conocer con se acceden los datos internamente. (Oscar Blancarte, 2018)

Diagrama de secuencia

Según (Pressman, 2010) un diagrama de secuencia se usa para mostrar las comunicaciones dinámicas entre objetos durante la ejecución de una tarea. Este tipo de diagrama muestra el orden temporal en el que los mensajes se envían entre los objetos para lograr dicha tarea.

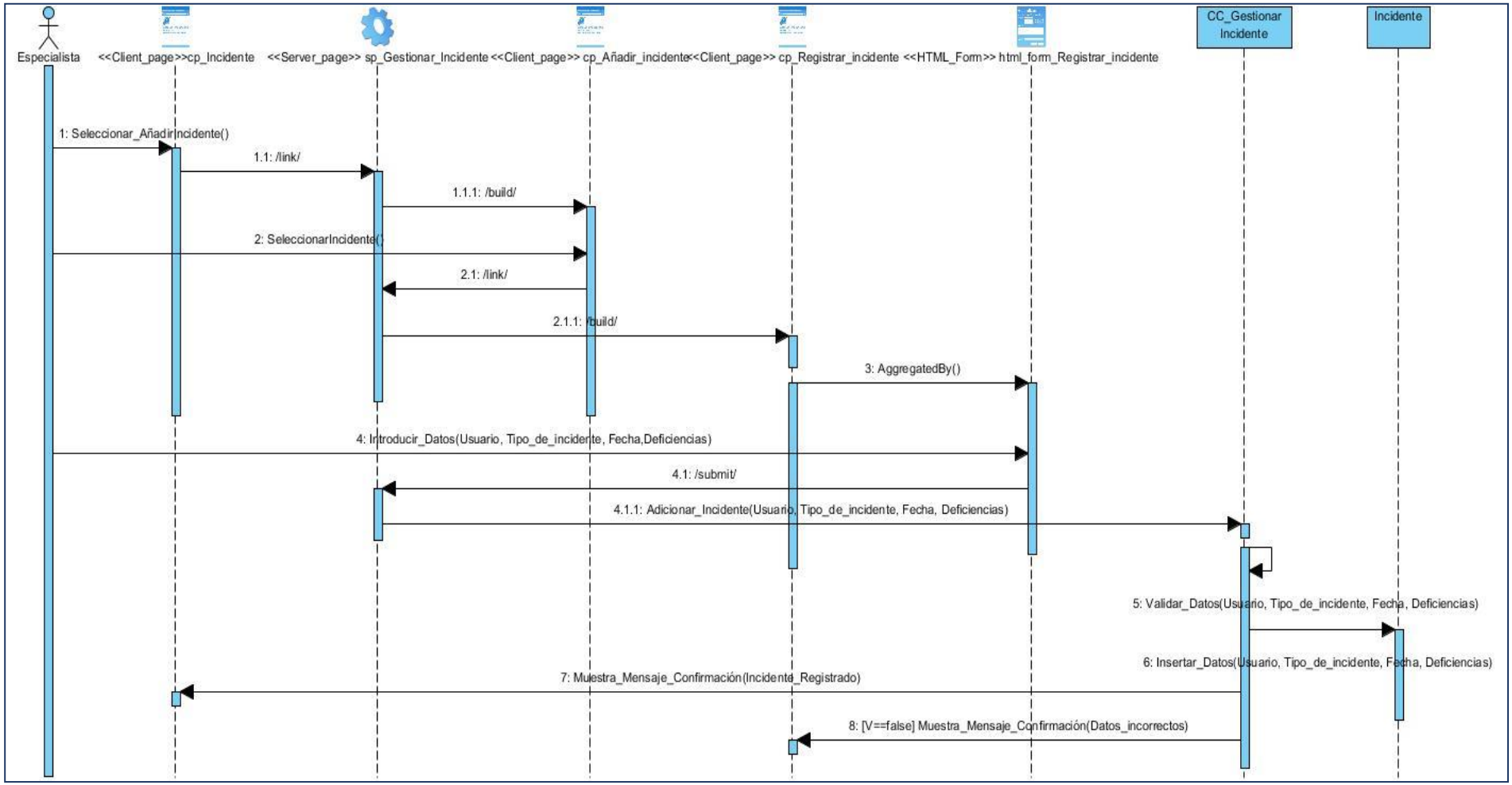


Figura 2.3 Diagrama de Secuencia para Registrar Incidente (Elaboración propia)

2.3 Modelo de despliegue

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema. Es una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar. Muestra la configuración de los componentes *hardware*, los procesos, los elementos de procesamiento en tiempo de ejecución y los objetos que existen en tiempo de ejecución (Pressman, 2010).

El nodo que representa la PC cliente es un conjunto de computadoras utilizadas por los usuarios (especialistas de seguridad informática) para tener acceso a las informaciones que se encuentran en el servidor web (Apache) a través de un navegador. La comunicación entre las PC clientes y el servidor web se establece utilizando el protocolo de comunicación segura HTTPS. El servidor de base de datos, que representa un servidor MySQL y permite el acceso a ella mediante el servidor web. Estos dos servidores se comunican mediante la familia de protocolos TCP. Además, establece una conexión con el LDAP utilizando el protocolo de comunicación segura HTTPS.

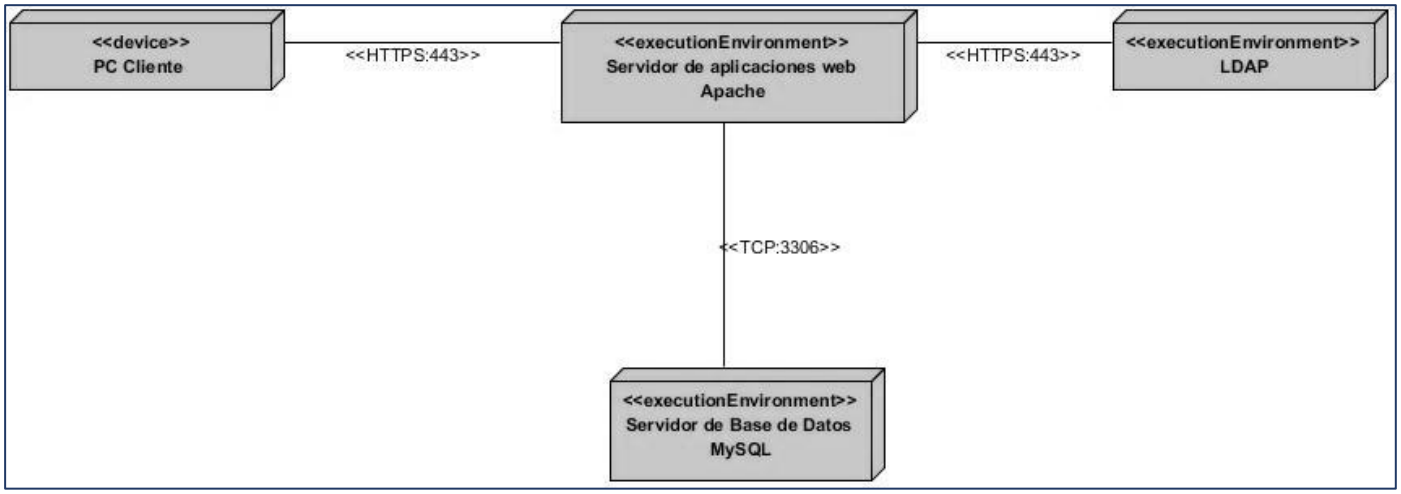


Figura 2.4 Modelo de Despliegue (Elaboración propia)

2.4 Conclusiones parciales

- El análisis de la propuesta de solución que se pretende concretar permitió, con el levantamiento de los requisitos del sistema, determinar las funcionalidades básicas a desarrollar durante el proceso.
- El análisis y diseño posibilitó seleccionar el patrón arquitectónico MVC, el cual permite una arquitectura reutilizable.
- Al emplear el lenguaje UML para modelar los artefactos propuesto por la metodología de desarrollo seleccionada durante el capítulo 1 se alcanzó una mejor comprensión sobre el cómo debe comportarse el sistema.
- El modelo de despliegue permitió identificar la estructura de los elementos de hardware y de software así como la forma en que se representan los nodos y sus relaciones.

CAPÍTULO 3. Implementación y validación del SGSI

En este capítulo se exhiben los diagramas asociados, estándares de codificación y diseños de casos de prueba a utilizar en la validación del sistema y se analizan los resultados de las pruebas realizadas que permiten evaluar la calidad de la propuesta de solución.

3.1. Diagrama de componentes

El diagrama de componentes muestra los componentes de un sistema de software conectados por las relaciones de dependencias lógicas entre cada uno de ellos. Provee una vista arquitectónica de alto nivel del sistema, ayudando a los desarrolladores a visualizar el camino de la implementación. Cada componente representa una unidad del código (fuente, binario o ejecutable), que permite mostrar las dependencias en tiempo de compilación y ejecución. La realización del diagrama posibilita tomar decisiones respecto a las tareas de implementación y los requisitos (Pressman, 2010)

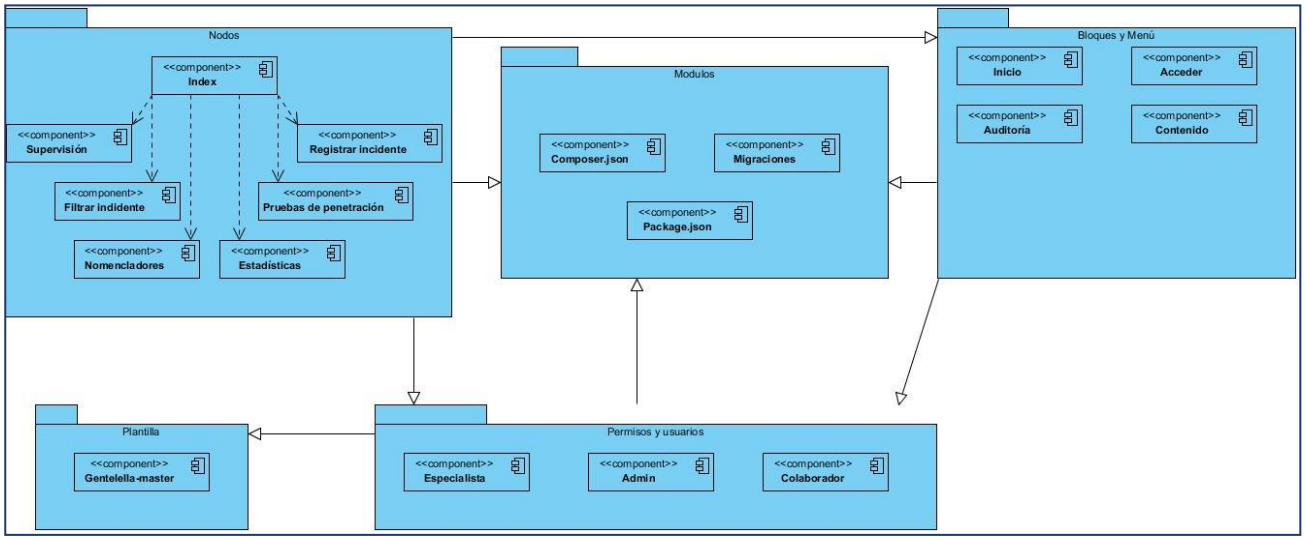


Figura 3.1 Diagrama de Componentes (Elaboración propia)

3.2. Estándares de codificación

Los estándares de codificación constituyen un principio esencial en el desarrollo de software. Garantizan que el código obtenido sea fácil de leer, entendido y modificado independientemente de quién haya sido el desarrollador del producto. Son una guía para el equipo de desarrollo, permiten asegurar que el código presente calidad y no contenga errores. Laravel sigue los estándares PSR-1 y PSR-4. Y además tiene algunas recomendaciones propias. Lo que algunos entornos llaman el "*Laravel <<flavor>> of PSR-2*". A continuación, se detallan los estándares de codificación utilizados en la implementación de la solución propuesta.

- La declaración del *namespace* debe estar en la misma línea que <?php

```php
<?php

namespace App\Http\Controllers;

use App\Http\Requests\CreateIncidenteRequest;
```

Figura 3.2 (Código fuente propio)

- Las llaves de apertura de las clases deben ir en la misma línea que el nombre de la clase.

```php
public function index()
{
return view('incidente.index',[
    'incidente' => Incidente::get(),
    'incid' => subcategoria::where('nomenclador_id','=', 5)->get(),
]);
}
```

Figura 3.3 (Código fuente propio)

- Las funciones y estructuras de control deben seguir el estilo de llaves Allman. El estilo Allman define que la llave de apertura de las estructuras de control debe ir en las líneas de control siguiente. La llave de cierre debe estar al mismo nivel que la de apertura. Y el cuerpo de la estructura debe estar indentado.

```
if($busq=='tipo')
{
    $inc = Incidente::where('tipo', 'like', '%'.$search.'%')->paginate(10);
    //->orWhere('usuario', 'like', '%'.$search.'%')

}
```

Figura 3.4 (Código fuente propio)

3.3. Aplicación de la estrategia de validación.

El único instrumento adecuado para determinar el *status* de la calidad de un producto de *software* es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del *software* o al sistema de *software* en su totalidad, con el objetivo de medir el grado en que cumple con los requerimientos (Sommerville, 2011).

3.3.1. Pruebas de rendimiento

La prueba de rendimiento se diseña para poner a prueba el rendimiento del software en tiempo de ejecución, dentro del contexto de un sistema integrado. Esta prueba ocurre a lo largo de todos los pasos del proceso de prueba. Incluso en el nivel de unidad, puede accederse al rendimiento de un módulo individual conforme se realizan las pruebas. Sin embargo, no es sino hasta que todos los elementos del sistema están plenamente integrados cuando puede determinarse el verdadero rendimiento de un sistema (Pressman, 2010).

***Hardware* de prueba (PC servidor):**

- Sistema Operativo: Linux Mint 19
- Microprocesador: i7 8th gen
- Memoria RAM: 16.00 GB
- Disco Duro: 756(SSD) GB

***Software* instalado en la PC:**

- Tipo de servidor web: Apache 2.4
- Plataforma: SO Linux

- Servidor de BD: MySQL 5.7.24

Luego de definido el hardware, se configuran los parámetros del Apache JMeter logrando un ambiente de simulación con un total de 30 y 40 usuarios conectados concurrentemente en una primera y segunda prueba respectivamente, se realizan peticiones a diferentes módulos del SGSI.

Análisis de los resultados de las pruebas de rendimiento

Para un mejor entendimiento de las pruebas de rendimiento, se explica cada parámetro que la compone a continuación:

- **Usuarios:** total de usuarios.
- **# Muestras:** el número de peticiones.
- **Media:** El tiempo medio transcurrido en milisegundos para un conjunto de resultados.
- **Mín:** El mínimo tiempo transcurrido en milisegundos para las muestras de la URL dada.
- **Máx:** El máximo tiempo transcurrido en un milisegundo para las muestras de la URL dada.
- **% Error:** Porcentaje de las peticiones con errores.
- **Rendimiento:** Rendimiento medido en base a peticiones por segundo/minuto/hora.
- **Kb/s Recibidos:** Rendimiento medido en *Kbytes* por segundos.

Usuarios	# Muestras	Media	Mín	Máx	% Error	Rendimiento (peticiones/segundos)	Kb/s Recibidos
10	60	1204	85	5765	0,00%	5,8/sec	22.4
20	75	2459	106	4894	0,00%	7,5/sec	66.1

Tabla 3.1 Resumen de los resultados de las pruebas de rendimiento (Elaboración propia).

De los resultados obtenidos en las pruebas se determinó que el SGI cumple con los requisitos no funcionales definidos en el levantamiento realizado con el cliente. Para ello se realizó una primera prueba donde para un total de 10 usuarios conectados al sistema de forma concurrente, el mismo respondió 60 peticiones al servidor en un promedio de 1204 segundos, se obtuvo un tiempo mínimo de 2,006 segundos al cargar la página y 5765 segundos como tiempo máximo, para un porciento de error de 0,00 %.

Para reducir el tiempo de respuesta del servidor, se realizaron diferentes acciones. Se especificó el tiempo mínimo de permanencia en caché y la caducidad de las páginas en caché. Se optimizaron los ficheros CSS y JavaScript, quitando espacios y nuevas líneas; almacenando la información en un solo fichero. De esta forma se reduce el tamaño al mínimo posible, mejorando así la velocidad de carga de la web. También se activó la compresión de CSS y JavaScript en la sección *«Optimización de ancho de banda»* pues permite comprimir los recursos de la página en un solo archivo. Se definió un estilo específico a cada tipo de imagen pues el peso de las mismas influye en la velocidad del sistema.

En una segunda prueba se obtuvo un menor tiempo de respuesta del servidor para 20 usuarios conectados concurrentemente en un intervalo de 5,8 de peticiones por segundos, se obtuvo un tiempo mínimo de 3,18 segundos en cargar la página para un tiempo máximo de 5,28 segundos, con un porciento de error de 0,00 % para un total de 75 peticiones al servidor. Atendiendo a la cantidad de peticiones por cada segundo que se enviaron, las prestaciones del *hardware* donde se realizaron las pruebas se considera que constituye un resultado satisfactorio.

3.3.2. Pruebas de seguridad

Según (Pressman, 2010) las pruebas de seguridad intentan verificar que los mecanismos de protección incorporados en el sistema lo protegerán de accesos inapropiados. Durante las pruebas de seguridad, el responsable de la prueba desempeña el papel de un individuo que desea entrar en el sistema. Debe intentar conseguir las claves de acceso por cualquier medio, debe bloquear el sistema, negando así el servicio a otras personas.

Resultados de las pruebas de seguridad Con el objetivo de evaluar la seguridad de la solución propuesta se emplea la herramienta *OWASP ZAP 2.7.0* la cual arrojó los siguientes resultados luego de realizada una primera iteración.

Categorías de vulnerabilidades	Cantidad de errores
Formularios HTML sin protección CSRF	0
Credenciales de usuarios enviadas en texto plano	4
Vínculos rotos	3
Campos de contraseña con auto completamiento activado	2
Campos de usuario y contraseña mostrados	0
Total	9

Tabla 3.2 Resultados de las pruebas de seguridad (Elaboración propia).

Resultados de las pruebas de seguridad

Después de analizar los resultados obtenidos en las pruebas se procedió a corregir las deficiencias encontradas. Para ello se llevaron a cabo un conjunto de acciones que permitió reforzar la seguridad del SGI. Los formularios HTML sin protección CSRF (falsificación de petición en sitios cruzados, en español) es una clase de ataque que afecta a las aplicaciones basadas en web. El ataque funciona mediante la inclusión de un enlace o secuencia de comandos en una página que tiene acceso a un sitio al que se conoce el usuario (o se supone) que se han autenticado. Esta vulnerabilidad también es conocida por otros nombres como sección de manejo y ataque de un *click*. En este caso hubo 0 casos de formularios sin CSRF porque laravel por defecto no permite la creación de un formulario sin utilizar el comando @csrf en las páginas *.blade.php.*

Campos de contraseña con auto completamiento activado**,** cuando se introduce un nuevo nombre y contraseña en un formulario y se envía el formulario, el navegador le pregunta si la contraseña debe ser guardada. Cuando se muestra el formulario, el nombre y la contraseña se rellenan de forma automática o se completan como se introduce el nombre. Un atacante con acceso local podría obtener la contraseña de texto plano de la caché del navegador. Para darle solución, la función de la contraseña de autocompletar debe ser desactivada. Para desactivar la función de autocompletar, es posible utilizar un código similar al siguiente: <*INPUT TYPE*="*password*" *AUTOCOMPLETE*="off">.

3.3.3. Pruebas funcionales

Las pruebas funcionales son aquellas que se aplican a un software determinado, con el objetivo de validar que las funcionalidades implementadas funcionen de acuerdo a las especificaciones de los requisitos definidos con anterioridad. Para la ejecución de este tipo de pruebas, suelen emplearse dos métodos fundamentales: el método de Caja Blanca y el método de Caja Negra. El primero se centra en las pruebas al código de las aplicaciones; mientras que el segundo permite a los probadores enfocar su atención en el funcionamiento de la interfaz, a través del análisis de los datos de entrada y los de salida (Pressman, 2010). A continuación, se muestra un ejemplo de diseño de casos de prueba de aceptación utilizado para detectar errores en la aplicación y mostrar si cumplía con los requisitos. Se describe el caso de prueba **Crear Incidente.**

Caso de prueba *Crear Incidente*.	
Código de caso de prueba: 1	**Nombre de Historia de Usuario:** Crear Incidente
Nombre de la persona que realiza la prueba: Fernando Paciencia Luteiro Palaia	
Descripción de la prueba: Prueba a la funcionalidad crear incidente	
Entrada / Pasos de la ejecución: La entrada consta de la introducción de los datos en los siguientes campos: • **Nombre:** Fernando Paciencia • **usuario:** fpaciencia • **Tipo de Incidente:** Robo de credenciales • **Fecha: 15/05/2020** • **Descripción:** El usuario ha robado las credenciales de su compañero de aula utilizando un programa prohíbo por la dirección de seguridad Informática	
Resultado esperado: Se crea un nuevo incidente en la base de dados	
Evaluación de la prueba: Satisfactoria	

Tabla3.3 Crear Incidente (Elaboración propia).

En total, se planificaron tres iteraciones de prueba. La figura 11 brinda información sobre el total de no conformidades encontradas y las que se resolvieron por cada iteración. Para un total de 34 requisitos funcionales se detectaron 60 no conformidades en la primera iteración y se resolvieron 56, las cuales fueron resueltas satisfactoriamente, y en la segunda iteración se redujo las no conformidades llegando 18 no conformidades, que fueron resueltas en su totalidad, número que se redujo hasta la tercera iteración donde no se obtuvieron no conformidades.

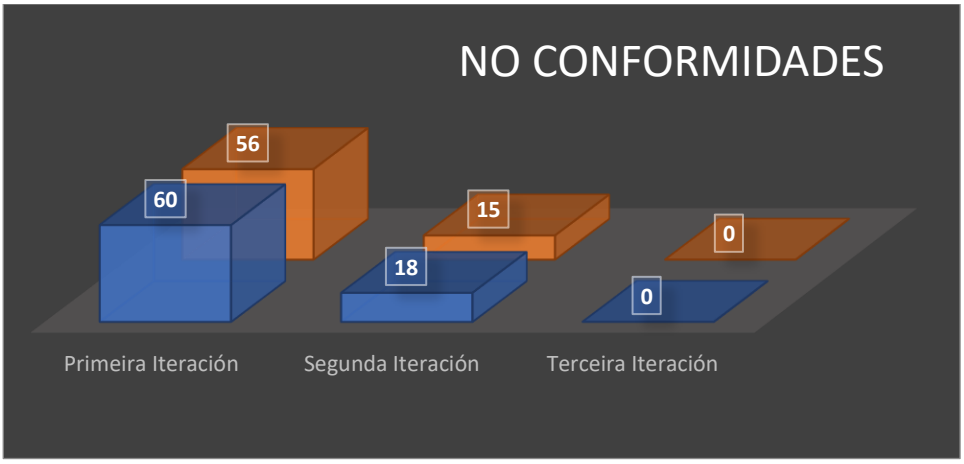


Figura 3.5 Resultado de las pruebas funcionales (Elaboración propia).

Entre las no conformidades detectadas en el proceso de pruebas funcionales se encuentran:

- Opciones que no funcionan
- Inconsistencia con algunas URLs
- Los mensajes presentan problemas de idiomas.
- Operaciones que se podrían realizar sin estar autenticado con un usuario permitido.

En total, se planificaron tres iteraciones de prueba. La figura 12 brinda información sobre el total de no conformidades encontradas y las que se resolvieron por cada iteración. Para un total de 34 requisitos funcionales se detectaron 60 no conformidades en la primera iteración y se resolvieron 56, las cuales fueron resueltas satisfactoriamente, y en la segunda iteración se redujo las no conformidades llegando 18 no conformidades, que fueron resueltas 15, número que se redujo hasta la tercera iteración donde no se obtuvieron no conformidades.

3.3.4. Pruebas de usabilidad

Para garantizar la seguridad de SGSI se realizan pruebas de usabilidad mediante una lista de chequeo aplicable fundamentalmente a SGI y aplicaciones web. Dicha lista establece un conjunto de preguntas formuladas en 9 categorías. El objetivo general de esta lista es evaluar a través de indicadores establecidos por los especialistas del grupo de seguridad del Departamento de Pruebas de Software

(DEPSW) de la UCI la usabilidad de las aplicaciones. A continuación, se muestran los resultados para 2 de las 9 categorías:

Forma de uso

Evaluación: Es la forma de evaluar el indicador en cuestión. El mismo se evalúa de 1 en caso de mal (cuando la respuesta al indicador sea “No”) y 0 en caso que elemento revisado no presente errores (cuando la respuesta al indicador sea “Sí”). NP (No Procede): Se usa para especificar que el indicador a evaluar no se puede aplicar en ese caso.

Visibilidad del sistema	Evaluación	NP
¿La página refleja la identidad de la empresa, logos, compañía...)?	0	
¿Cada pantalla empieza con un título que describe su contenido?	0	
¿Cuándo se selecciona un icono se diferencia de los no seleccionados?	0	
¿Los enlaces del menú se resaltan cuando se seleccionan?	0	
¿Los iconos que aparecen se identifican claramente con lo que representan?	0	
¿El menú de navegación aparece en un lugar destacado?		X
¿No utiliza más de siete opciones principales en el menú de navegación?		X
¿Si la respuesta a una acción se retrasa, aparece un mensaje o indicio como que el sistema está procesando la acción?	0	
¿El sistema le indica al usuario en que parte de la estructura de la aplicación web se encuentra, es decir si muestra 'migas de pan'?		X
¿El nombre de los enlaces es el mismo que el título de la página a la que dirige?	0	
¿El logo de la organización está ubicado en el mismo lugar en todas las páginas, y hacer click en el logo retorna al usuario a la página más lógica (Ejemplo: la página de inicio)?	0	
¿Los títulos de las páginas, tablas e imágenes son descriptivos y distintivos?		X
¿Las etiquetas de las categorías describen con precisión la información de las mismas?	0	
¿Cuándo una tarea involucra documentos fuente, la interfaz es compatible con las características del documento fuente?	0	
¿Las imágenes se muestren con buena resolución?		X
¿No se muestran errores ortográficos?	0	
¿No hay ninguna imagen con información relevante?		X

Tabla 3.4 Indicadores de la categoría. Visibilidad del sistema (Elaboración propia).

Lenguaje común entre sistema y usuario	Evaluación	NP
¿El lenguaje es simple, con un tono adecuado?	0	
¿La información que se presenta en la aplicación es fácil de entender y memorizar?	0	
¿Utiliza los conceptos establecidos para las funciones estándar? (“buscar" para las búsquedas, etc.)	0	
¿Evita el lenguaje técnico: términos informáticos o propios de Internet?	0	
¿Se utiliza siempre la misma nomenclatura para las mismas funciones?	0	
¿Los acrónimos y abreviaturas son definidos al ser usados por primera vez?		X
¿No hace uso de términos extranjeros?	0	
¿Utiliza un texto específico y descriptivo en los vínculos?	0	
¿La información es de rápida lectura, y con una disposición asequible?	0	
¿Los vínculos basados en nombres de la gente, conducen a las biografías cortas o a sus propios blogs, no a un correo electrónico?		X
¿Si se desea incluir un enlace de correo electrónico, se muestra el correo y no el nombre de la persona?		X

Tabla 3.5 Indicadores de la categoría. Lenguaje común entre sistema y usuario. (Elaboración propia)

En las tablas anteriores se puede apreciar un total de 28 indicadores de usabilidad, el sistema de desarrollado utiliza 18 de ellos, el resto (10) no procede para la aplicación. De los 18 necesarios, el sistema cumple con 18 indicadores, cifra que representa el 100% de usabilidad para las funciones presentes.

3.3.5. Prueba de aceptación

El uso de cualquier producto de software tiene que estar justificado por las ventajas que ofrece. Sin embargo, antes de comenzar su explotación es muy difícil determinar si sus ventajas realmente justifican su uso. El mejor instrumento para esta determinación es la llamada prueba de aceptación. En esta prueba

se evalúa el grado de calidad del software con relación a todos los aspectos relevantes para que el uso del producto se justifique (Manuel Cillero, 2018).

La Junta Internacional de Cualificaciones de Pruebas de Software (*ISTQB* por sus siglas en inglés) define la “Aceptación” como: Pruebas formales con respecto a las necesidades del usuario, requerimientos y procesos de negocio, realizadas para determinar si un sistema satisface los criterios de aceptación que permitan que el usuario, cliente u otra entidad autorizada pueda determinar si acepta o no el sistema (Manuel Cillero, 2018)

Una vez concluida la investigación y el período de desarrollo, se entregó la aplicación al cliente para realizar la aceptación del producto y este expresó su criterio mediante una carta de aceptación, a partir de sus consideraciones respecto a las ventajas que ofrece el SGI a la mejora de la gestión de la información para el Departamento de Seguridad Informática en la UCI y las insuficiencias que resuelve.

Criterio de experto

El juicio de expertos es un método de validación útil para verificar la fiabilidad de una investigación que se define como “una opinión informada de personas con trayectoria en el tema, que son reconocidas por otros como expertos cualificados en éste, y que pueden dar información, evidencia, juicios y valoraciones” La evaluación mediante el juicio de expertos, método de validación cada vez más utilizado en la investigación, “consiste, básicamente, en solicitar a una serie de personas la demanda de un juicio hacia un objeto, un instrumento, un material de enseñanza, o su opinión respecto a un aspecto concreto” Se trata de una técnica cuya realización adecuada desde un punto de vista metodológico constituye a veces el único indicador de validez de contenido del instrumento de recogida de datos o de información; de ahí que resulte de gran utilidad en la valoración de aspectos de orden radicalmente cualitativo.

La secuencia establecida para comenzar este proceso es la siguiente:

1. Se establece contacto con los expertos conocedores y se les pide que participen en el panel.
2. Se manda un cuestionario a los miembros del panel y se les pide que den su opinión en los temas de interés.
3. Se analizan las respuestas y se identifican las áreas en que están de acuerdo y en las que difieren.

4. Se manda el análisis resumido de todas las respuestas a los miembros del panel, se les pide que llenen de nuevo el cuestionario y den sus razones respecto a las opiniones que difieren.
5. Se repite el proceso hasta que se estabilizan las respuestas. (Sandra Hurtado de Mendoza Fernández, 2019)

3.3.6 Conclusiones parciales

- La confección del diagrama de componentes permitió observar la integración de los componentes de *software*.
- Aplicar los estándares de codificación permitió obtener en el sistema un código legible, estándar y fácil de comprender lo que asegura la calidad y facilita un futuro mantenimiento.
- Como mecanismo para asegurar la correcta ejecución de las funcionalidades del sistema se realizaron las pruebas de rendimiento, usabilidad, aceptación, funcionalidad y seguridad

- El estudio de los referentes teóricos logró un mejor entendimiento de los conceptos asociados a la evolución de la web, difusión de la información y sistemas de gestión de la información.

- El análisis de las diferentes herramientas y tendencias para la realización de SGI permitió determinar la no existencia de un sistema de gestión de información que responda a las necesidades requeridas por el cliente.

- La implementación del sistema a través de las herramientas y lenguajes seleccionados permitió obtener un sistema de gestión de información de seguridad informática capaz mejorar el proceso de gestión de información en la dirección de seguridad Informática en la Universidad de las Ciencias Informáticas.

- El diseño de la propuesta de solución permitió generar los artefactos más significativos de acuerdo con la metodología de desarrollo de software AUP-UCI tomándose como referencia los requisitos detectados.

- Las definiciones de las necesidades del cliente a través de las historias de usuario, propició el funcionamiento adecuado del sistema.
- Las técnicas de validación aplicadas a la propuesta de solución permitieron la detección y corrección de las no conformidades detectadas y evidenciaron que el sistema constituye una solución funcional.

Una vez concluida la investigación y el desarrollo de la propuesta de solución, los autores del presente trabajo recomiendan:

- Automatizar el proceso para adicionar incidentes a la base de datos del sistema y obtener la información de diversas fuentes como correos electrónicos, archivos de texto y servicios web.
- Para el módulo de auditoría se recomienda la implementación de una funcionalidad que permita realizar auditorías internas a las distintas áreas de la Universidad de forma remota a través de script.
- Implementar un módulo en la aplicación que gestione la información de las pruebas de penetración aplicadas a los portales de la Universidad

  .

1. Alex Mapeli, 2020. NCS. Definitions. Web corporativa. Disponible en: https://www.definitions.net/definition/ncsa+httpd.

2. Andrea Rodríguez, 2016. Que es BSD. Web Corporativa. Disponible en: http://hipertextual.com/archivo/2016/05/que-es-bsd/.

3. Antonio Leyva, 2016. Patrones de diseño de software. Disponible en: https://devexperto.com/patrones-de-diseño-software/.

4. BELLOCH ORTÍ, 2014. Las Tecnologías de la Información y las Comunicación. Web de noticias, TIC.

5. Carlos Fariñas, 2020. CSS3 [Consulta: 11 febrero 2020]. Disponible en: https://www.informatica.org/CSS3. Web educativo.

6. Daniel Suárez, 2020. Que es PHP. Disponible en: https://www.php.net/manual/es/intro-whatis.php. WEB Educativo.

7. Dr César Antonio González Horruitiner, 2017. Por la protección del hombre y su medio. *Qué es la gestión de la Información* Disponible en: https://instituciones.sld.cu. Web corporativa.

8. Eduardo García, 2014. Factory Method. Disponible en: https://refactoring.guru. WEB Educativa. Patrones de diseño

9. Ernesto Sosa, 2019. jQuery. En: Page Versión ID: 120193158, [Consulta: 11 febrero 2020]. Disponible en: https://es. informatica.org/w/index.php?title=JQuery&oldid=120193158.

10. Evelio Morejón, 2020. Bootstrap (framework). En: Page Versión ID: 122653369, [Consulta: 11 febrero 2020]. Disponible en: https://es.wikipedia.org/w/index.php?title=Bootstrap_(framework)&oldid=122653369.

11. Gustavo B, 2019. Descripción completa del sevidor web Apache. *Que es Apache.* Disponible en: https://www.hostinger.es/tutoriales/que-es-apache/.Web Educativa.

12. Ian Sommerville, 2011. *Ingeniería de Software*. 7ma. S.l.: s.n. Pages 641, https://books.google.com.cu/books?id=gQwd49zSut4C&pg=PR17&lpg=PR17&dq

13. Isaura Peralta, 2019. Impacto del uso de patrones de diseño en la industria del software en Costa Rica. Tecnología Vital. WEB Educativa

14. ISO 27001 - Software ISO 27001 de Sistemas de Gestión. 2018. [Consulta: 11 febrero 2020]. Disponible en: https://www.isotools.org/normas/riesgos-y-seguridad/iso-27001/.

15. ISO 27002 - Software ISO 27001 de Sistemas de Gestión. 2018. [Consulta: 11 febrero 2020]. Disponible en: https://www.isotools.org/normas/riesgos-y-seguridad/iso-27001/.

16. Jhon S. Master of Architecture. *MVC Pattern* Disponible en: https://tutorialspoint.com.Web educativa.

17. Jorge Fernández, 2020. HTML. En: Page Versión ID: 123355611, [Consulta: 11 febrero 2020]. Disponible en: https://es. informatica.org/w/index.php?title=HTML&oldid=123355611.

18. Jorge Luis Perdomo Di- Lella, 2019. *Resolución 128*. S.l.: s.n. Reglamento de Seguridad de las Tecnologías de la información y la Comunicación.15 paginas. Sección I y II

19. Jorge Luis Perdomo Di- Lella, 2019. *Resolución 129*. S.l.: s.n. Metodología para la investigación de la Seguridad Informática.16 paginas. Sección II y III

20. José Manuel Rosa, 2020. Que es REST. Conoce su potencia. *Que significa REST* Disponible en: https://openwebinars.net/blog/que-es-rest-conoce-su-potencia.

21. Joseph Somerhaier, 2020. AJAX Introduction. Disponible en: https://www.w3schools.com/xml/ajax_intro.asp. WEB Educativa.

22. KENDALL, G.B., 2019. Que es MySQL. *Explicación detallada para principiantes* Disponible en: https://hostiger.es. WEB Eduactiva.

23. KENDALL, I.J., 2019. Que es UML. Disponible en: http://stadium.unad.edu.co/ovas/10596_9839/qu_es_uml.html. WEB Educativa.

24. Luis M Pérez, 2019. Seguridad Industrial. Disponible en: https://incibe-cert.es. WEB Educativa. SI.

25. Manuel Cillero, 2018. Pruebas de Aceptación. Disponible en: https://manuel.cillero.es/doc/metrica-3/tecnicas/pruebas/aceptacion.WEB Educativa

26. MARÍA JOSEFINA VIDAL LEDO, G.G.P., 2016. La informática y la seguridad. Un tema de importancia para el directivo. Disponible en: https://www.researchgate.net/publicacion/30354168_La_informatica_y_la_seguridad_Un_tema_de_importancia_para_el_directivo.

27. Mario López, 2020. JavaScript. En: Page Versión ID: 123477602, [Consulta: 11 febrero 2020]. Disponible en: https://es.informatica.org/w/index.php?title=JavaScript&oldid=123477602.

28. Martínez A. L, 2018. Sistema de Gestión de Seguridad de la Información para la Secretaría de Economía y Empresas de menor tamaño en Chile.WEB Corporativa.

29. Miguel Díaz-Canel Bermúdez, 2019 *Decreto-Ley No. 370*. S.l.: s.n. Informatización de la Sociedad en Cuba. 37 páginas, Sección I y II

30. NADER-CEBALLOS, E.-R., 2015. Que son las metodologías de desarrollo de software. Disponible en: obsbusiness.school/es/blog-project-management/metodologia-agile/que-son-las-metodologias-de-desarrollo-de-software.

31. Neysi Milagros Arias Santana, 2013. Sistemas de Información. *Importancia de los Sistemas de Información*. WEB Corporativa.

32. Oscar Blancarte, 2018. DAO Pattern. Disponible en: https://scarblancarteblog.com WEB Educativa. Patrones de diseño.

33. Pedro Damián López, 2015. Que es phpMyAdmin. Disponible en: https://hostinet.com. WEB Educativa

34. PÉREZ DEL CERRO, D.R. y D.PROENZA-PUPO, 2015. Oficina de Seguridad de Redes Informática (OSRI). Disponible en: https://mincom.gob.cu.

35. Rodolfo Acosta, 2019. El valor de la Gestión de Datos. *Que es un gestor de datos* Disponible en: https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/que-es-un-gestor-de-datos-y-para-que-sirve.

36. Roger S. Pressman, 2010. *Software Engineering*. 7ma. S.l.: s.n. 810 páginas https://www.pdfdrive.com/ingenieria-de-software-un-enfoque-practico-e58723181.html.

37. ROMERO, Y.F, G., Y.., 2015. Patrón arquitectónico. Disponible en: https://www.desarrollodepaginasweb.com.mx/patrones-de-arquitectura-de-software.

38. ROMERO Y.F, G.Y.., 2012. Patrón Modelo-Vista-Controlador. Revista Telemática. WEB Educativa. Patrones de diseño.

39. ROSERO, N.S.S.E.R.E. y M. DEL C. BENAVIDES, 2015. «Metodología de análisis y evaluación de riesgos aplicados a la seguridad informática y de información bajo la norma ISO/IEC 27001». , no. dic. 2015.

40. Sánchez D.A, 2018. Sistema para la Gestión de la Información de Seguridad Informática en la Universidad de Ciencias Médicas de Holguín. .

41. SÁNCHEZ VERA, R.V. y OCAÑA LA O, 2016. Que es Visual Paradigm. Disponible en: https://www.capterra.es/software/145716/visual-paradigm. WEB Educativa

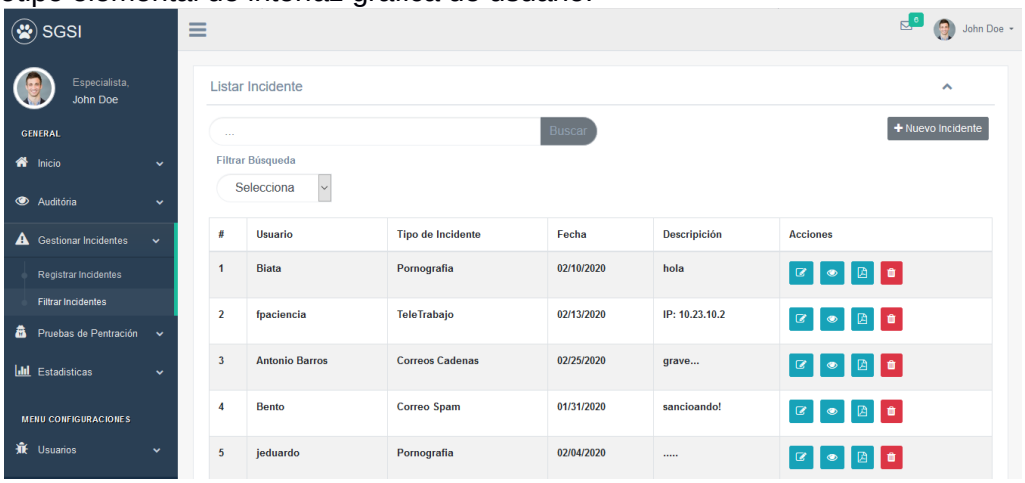
42. Sandra Hurtado de Mendoza Fernández, 2019. Criterio de expertos. Disponible en: http://www.ub.edu/histodidactica/index.php%3Foption%3Dcom_content%26view%3Darticle%26id%3D21:criterio-de-expertos.

43. SOLARTE, R., Ríos, 2018. UML, lenguaje de modelado gráfico. Disponible en: https://www.ionos.es/digitalguide/paginas-ficadoweb/desarrollo-web/uml-lenguaje-unificado-de-modelado-orientado-a-objetos.

44. SYNERGY, 2018. Ventajas del desarrollo a medida para tus proyectos. Proyectos de desarrollo. WEB Educativa

45. TEXCEL RESEARCH, J.R., 2020. Que es el Modelo de Objetos del Documento. Disponible en: http://html.coonclase.net/w3c/dom1-es/introduction.html.

46. VIEITES, Á.G., 2011. *Enciclopedia de la Seguridad Informática. 2ª edición*. S.l.: Grupo Editorial RA-MA. ISBN 978-84-9964-394-6. 240 páginas.

Número: 1. Nombre del requisito: Filtrar Incidente.	
Programador: Fernando Paciencia Luteiro Palaia.	Iteración Asignada: 2ra
Prioridad: Alta	Tiempo Estimado: 4h
Riesgo en Desarrollo: Alta	Tiempo Real: 2h
Descripción: El sistema permite filtrar información referente a un incidente.	
Observaciones: En el caso filtrar incidente el sistema permite que los usuarios con roles de administrador y especialistas en incidentes informáticos filtren los incidentes registrados en el sistema. El sistema permite filtrar las informaciones por los siguientes campos de entrada: ▪ Nombre ▪ Tipo de Incidente ▪ Fecha	
Prototipo elemental de interfaz gráfica de usuario: 	

Anexo 1. Capítulo 2 (Filtrar Incidente) (Elaboración propia)

REST: define un conjunto de principios arquitectónicos mediante los cuales puede diseñar servicios web que se centran en los recursos de un sistema, incluida la forma en que los estados de recursos son abordados y transferidos a través de HTTP por una amplia gama de clientes escritos en diferentes idiomas. Si se mide por el número de servicios web que lo utilizan, REST ha surgido en los últimos años como un modelo de diseño de servicios web predominante (José Manuel Rosa, 2020)

ORM: Es un acrónimo de Mapeo Objeto Relacional.

MVC: Modelo Vista Controladora.

DOM o *Document Object Model*: es un conjunto de utilidades específicamente diseñadas para manipular documentos XML. Por extensión, DOM también se puede utilizar para manipular documentos XHTML y HTML. Técnicamente, DOM es una API de funciones que se pueden utilizar para manipular las páginas XHTML de forma rápida y eficiente (Texcel Research, 2020).

AJAX: Es un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como "JavaScript asíncrono + XML". No es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes (Joseph Somerhaier, 2020).

BSD: *Berkeley Software Distribution* (en español, «distribución de software Berkeley») fue un sistema operativo derivado de Unix que nace a partir de los aportes realizados a ese sistema por la Universidad de California en Berkeley (Andrea Rodríguez, 2016).

NCSA: Era un servidor web desarrollado originalmente en el *National Center for Super computing Applications* por Robert McCool y una lista de colaboradores (Alex Mapeli, 2020).