ACLARACIONES:

* Las indicaciones encerradas entre <paréntesis angulares> son de obligatorio cumplimiento de no establecerse lo contrario de forma explícita y al completar el texto con lo que se indica deben ser removidas por completo las indicaciones y los paréntesis.
* Se eliminarán los paréntesis que indican plural en caso donde no correspondan al ser singular el sujeto al que hacen referencia.
* Esta página de “ACLARACIONES” debe ser eliminada al realizar la entrega de la memoria escrita.
* En la página “Declaración de autoría” aparecen espacios para dos autores y dos tutores; sin embargo, se utilizarán menos espacios en caso de no se necesario esta cantidad, de forma tal que los autores y tutores queden organizados equitativamente desde el centro de la página y siempre garantizando que los autores estén primero y los tutores debajo. Por ejemplo de ser dos autores y un tutor, se ubicarán los datos de los dos autores uno al lado del otro y en el renglón siguiente el único tutor en el medio de la página.
* Siempre que se escriba el nombre del tutor, co-tutor, consultante o asesor en cualquiera de las partes oficiales de la memoria (excepto en la dedicatoria y agradecimientos), debe hacerse en el formato: categoría docente (Inst., As., P.A., P.T., P.C.), título académico de pregrado (Ing. o Lic.), seguido del nombre completo con sus apellidos y terminando con el grado científico (Dr. Cs., Dr. C.) o título académico de postgrado (Ms. C.). Se suprimirán el grado científico, el título académico de postgrado y la categoría docente, de no poseerse. (Ejemplo: P.A., Ing. Jorge Pérez González, Dr. C.)
* Las tablas se designarán de forma consecutiva comenzando por el número uno (1) en la forma “*Tabla X: Descripción del contenido de la tabla*”, siempre en el borde superior de la tabla y con texto en formato Arial, tamaño 12, interlineado 1.0 punto, justificado y en formato *Italic*. Las filas utilizarán un interlineado sencillo (1.0 punto) y no harán uso de formatos preestablecidos por los editores de texto, sino que siempre serán con fondo blanco y líneas en color negro. Aparecerán siempre en la misma página o la siguiente a donde se les hace mención. Su numeración no cambia al cambiar las secciones de la memoria. Siempre debe declararse en caso de ocurrir si la tabla es tomada o modificada de una referencia bibliográfica, en la forma “[Tomado de/Modificado de: Sánchez (2021)]” a continuación de su designación y en formato Arial, tamaño 12, interlineado 1.0 punto, justificado y en formato Regular. (Ejemplo: *Tabla 1: Descripción del comportamiento de las frecuencias de aparición de errores en la evaluación de la calidad de software de aplicaciones web* [Tomado de: Sánchez (2021)])
* Las figuras se designarán de forma consecutiva comenzando por el número uno (1) en la forma “*Figura X: Descripción del contenido de la figura*”, siempre en el borde inferior de la figura y con texto centrado y en formato Arial, tamaño 12, interlineado 1.0 puntos, justificado y en formato *Italic*. Las figuras que correspondan a gráficos de pasteles, barras u otra forma de representación estadística de información no utilizarán colores para la diferenciación de las series numéricas, sino texturas en puntos, líneas o jeroglíficos y símbolos que puedan ser percibidos y diferenciados al imprimirse el texto en tono de grises. Aparecerán siempre en la misma página o la siguiente a donde se les hace mención. Su numeración no cambia al cambiar las secciones de la memoria. Siempre debe declararse en caso de ocurrir si la figura es tomada o modificada de una referencia bibliográfica, en la forma “[Tomado de/Modificado de: Sánchez (2021)]” a continuación de su designación y en formato Arial, tamaño 12, interlineado 1.0 puntos, centrado y en formato Regular. (Ejemplo: *Figura 1: Gráfica del comportamiento de las frecuencias de aparición de errores en la evaluación de la calidad de software de aplicaciones web* [Tomado de: Sánchez (2021)])
* La memoria se redacta en tiempo pretérito y en tercera persona del singular (el/la) de ser un autor o del plural (los) de ser dos autores y nunca en primera persona; sin el uso de adjetivos no académicos o científicos difíciles de comprobar, demostrar o que no tengan fundamento científico en la literatura o los métodos aplicados.
* Las citas textuales deben realizarse siempre en el propio documento separadas por “*comillas*” o «*paréntesis angulares*» y en letra con formato *Italic,* de forma tal que se distinga claramente del resto del texto.
* Los párrafos no deben exceder los ocho (8) renglones de extensión, y deben ser construidos con oraciones simples de una sola forma verbal, separadas con punto y seguido; limitando al máximo el uso excesivo de comas, puntos y comas y otros signos de puntuación que dificulten la lectura.
* Los párrafos no tienen separación de ningún tipo, excepto en los lugares en los que el formato que se proporciona en esta plantilla así lo establezca.
* Generalmente la distribución de la longitud de cada una de las secciones en este tipo de documento para la socialización de resultados científicos es la que sigue hasta completar un total de ochenta (80) cuartillas:
  + Introducción: hasta 5 cuartillas (≈ 5 %)
  + Capítulo 1 o Sección de Fundamentación teórico-metodológica: hasta 24 cuartillas (≈ 30 %)
  + Capítulo 2 o Sección de Propuesta de solución científica: hasta 36 cuartillas (≈ 45 %)
  + Capítulo 3 o Sección de Validación de resultados científicos: hasta 13 cuartillas (≈ 18 %)
  + Conclusiones: 1 cuartilla (≈ 1 %)
  + Recomendaciones: 1 cuartilla (≈ 1 %)
* Los anexos del documento no deben nunca exceder las diez (10) cuartillas (≈ 12 % del total de ochenta páginas del cuerpo principal de la memoria)
* Las cantidad de asentamientos bibliográficos deben estar en correspondencia con el estado del arte del objeto de estudio y el campo de acción de la investigación. Su calidad no está determinada por la cantidad, sino por otros indicadores bibliométricos como actualidad (al menos el 25 % debe ser de los últimos cinco años y el 50 % de los últimos 8 a 10 años), factor de impacto de las fuentes (de un mínimo de entre 0.3 y 0.5) y distribución de tipos de fuentes (artículos de revistas arbitradas, ponencias en eventos científicos, libros, tesis de pregrado, maestría y doctorado, informes de proyectos de investigación, informes de organismos y asociaciones nacionales, regionales o internacionales, informes técnicos de entidades reconocidas, artículos periodísticos de fuentes oficiales, materiales audiovisuales de fuentes oficiales).
* La presente plantilla está diseñada para una investigación del tipo “Informe de investigación aplicada como resultado de la construcción total o parcial de un producto informático en su ciclo completo de desarrollo”. Por cuanto, la estructura y contenido de los capítulos propuestos puede sufrir cambios para otros tipos de investigación, de acuerdo a lo que consideren el(los) autor(es) y el(los) tutor(es).



**FACULTAD 1**

**Módulo de gestión de Incidencias Tecnológicas de Soporte**

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

**Autor:** Roger Díaz Viamonte

**Tutores:** Ing. Delly Lién González Hernández Ms. C

La Habana, abril de 2023

“Año 65 del triunfo de la Revolución”

**DECLARACIÓN DE AUTORÍA**

|  |
| --- |
| **Roger Díaz Viamonte** |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Firma del autor |

El autor del trabajo de diploma con título ***“Módulo de gestión de Incidencias Tecnológicas de Soporte”*** concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firman la presente a los 26 días del mes de abril del año 2023.

|  |  |
| --- | --- |
|  | **Delly Lién González Hernández** |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Firma del Tutor | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Firma del Tutor |

**DATOS DE CONTACTO**

<Currículum e información de contacto del tutor: nombre y apellidos, títulos académicos, formación de postgrado recibida, lugar de trabajo, responsabilidades laborales asumidas, experiencia profesional, líneas de trabajo y/o investigación, correo electrónico, perfiles en redes profesionales>

<Currículum e información de contacto del asesor: nombre y apellidos, títulos académicos, formación de postgrado recibida, lugar de trabajo, responsabilidades laborales asumidas, experiencia profesional, líneas de trabajo y/o investigación, correo electrónico, perfiles en redes profesionales>

<Currículum e información de contacto del consultante: nombre y apellidos, títulos académicos, formación de postgrado recibida, lugar de trabajo, responsabilidades laborales asumidas, experiencia profesional, líneas de trabajo y/o investigación, correo electrónico, perfiles en redes profesionales>

**AGRADECIMIENTOS**

<Insertar agradecimientos a personas naturales o jurídicas que hayan contribuido de forma directa al desarrollo de la investigación y sin cuya participación no hubiera sido posible su ejecución. No deben confundirse con la sección “Dedicatoria” que tiene otros objetivos. Debe ser breve sin necesidad de argumentar el porqué del agradecimiento; por cuanto se sobre entiende que la mención corresponde al apoyo ofrecido en la realización del trabajo que se presenta. Esta sección es totalmente opcional y de no utilizarse se suprime del documento. Puede utilizarse un formato de letra distinto al que oficialmente se establece para el resto del documento, aunque pudiera esta selección producir un contraste no favorable para la lectura y legibilidad de la obra. No pueden exceder una cuartilla en su extensión>

**DEDICATORIA**

<Insertar dedicatoria a personas naturales o jurídicas a las que se desee dedicar especialmente el trabajo, bien por vínculos afectivos, familiares, o de membresía. No deben confundirse con la sección “Agradecimientos” que tiene otros objetivos. Debe ser breve y de argumentarse la razón de inclusión, debe mantenerse un lenguaje respetuoso y científico. Esta sección es totalmente opcional y de no utilizarse se suprime del documento. Puede utilizarse un formato de letra distinto al que oficialmente se establece para el resto del documento, aunque pudiera esta selección producir un contraste no favorable para la lectura y legibilidad de la obra. No pueden exceder una cuartilla en su extensión>

**RESUMEN**

<Breve explicación de la investigación como un todo en un único párrafo entre 150 y 250 palabras, que refleje el por qué de la investigación, su objetivo y objeto de estudio, principales métodos o estrategia metodológica en su ejecución, principales hallazgos y resultados obtenidos y principales conclusiones. Debe lograrse al leer el resumen, obtener una panorámica general de la investigación tanto en su carácter teórico como metodológico y práctico>

PALABRAS CLAVE

<Términos simples o compuestos que permitan indexar el trabajo de acuerdo a las temáticas principales que aborda. No deben exceder la cantidad de 4 ó 5 términos>

***ABSTRACT***

<*Se escribe en idioma inglés la traducción del texto en el resumen. Se escribe en letra con formato “Italic” para diferenciarlo visualmente de su antecesor en idioma español*>

*KEYWORDS*

*<Se escriben en idioma inglés la traducción de las palabras clave en español. Igual se hace en letra con formato “Italic” para diferenciarlas visualmente de sus antecesoras en idioma español>*

**TABLA DE CONTENIDOS**

[​](#__RefHeading___Toc30362_1777028182) INTRODUCCIÓN 1

​ CAPÍTULO I: Fundamentos y referentes teórico-metodológicos sobre el objeto de estudio 2

​ I.1 Nombre del Epígrafe I.1 2

​ I.2 Nombre del Epígrafe I.2 2

​ I.3 Nombre del Epígrafe I.3 2

​ I.3 Nombre del Epígrafe I.4 2

​ Conclusiones del capítulo 3

​ CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO 4

​ II.1 Nombre del Epígrafe II.1 4

​ II.2 Nombre del Epígrafe II.2 4

​ II.3 Nombre del Epígrafe II.3 4

​ II.4 Nombre del Epígrafe II.4 4

​ Conclusiones del capítulo 4

​ CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA 6

​ III.1 Nombre del Epígrafe III.1 6

​ III.2 Nombre del Epígrafe III.2 6

​ III.3 Nombre del Epígrafe III.3 6

​ Conclusiones del capítulo 6

​ CONCLUSIONES FINALES 8

​ RECOMENDACIONES 9

​ REFERENCIAS BIBLIOGRÁFICAS 10

​ ANEXOS 12

**ÍNDICE DE TABLAS**

<Listado de las tablas en orden de aparición en la memoria>

**ÍNDICE DE FIGURAS**

<Listado de las figuras en orden de aparición en la memoria>

**OPINIÓN DEL(OS) TUTOR(ES)**

<Contenido de la opinión de los tutores>

**AVAL DEL CLIENTE**

<Contenido del aval del cliente sobre la solución desarrollada>

# INTRODUCCIÓN

Los avances Tecnológicos en las aplicaciones web han sido uno de los mayores impulsores del cambio en la forma en que interactuamos con el mundo digital. En los últimos años, se observa una evolución constante en la tecnología web, que ha llevado a una mayor eficiencia, seguridad y funcionalidad en las aplicaciones. El desarrollo de la tecnología web ha permitido la creación de aplicaciones más complejas y sofisticadas, que antes eran imposibles de desarrollar. Dichas aplicaciones modernas son altamente interactivas y ofrecen una experiencia de usuario fluida e intuitiva (Paredes Colmenar, 2020). Por consiguiente, la tecnología web ha brindado la creación de aplicaciones móviles, lo que ha llevado a una mayor accesibilidad y comodidad para los usuarios. Se pueden observar varios avances significativos en el ámbito de aplicaciones web, entre estos se pueden encontrar (Luna et al., 2018):

El desarrollo de HTML5. Esta nueva versión del lenguaje de marcado ha permitido la creación de aplicaciones más ricas en contenido multimedia como video y audio. También ha mejorado la experiencia con el usuario al brindar la creación de aplicaciones más rápidas y responsivas. Otro avance importante en la tecnología web es la introducción de CSS3[[1]](#footnote-1). Este lenguaje de estilo ha posibilitado la creación de diseños más sofisticados y personalizados para las aplicaciones web. Junto con HTML5[[2]](#footnote-2) brinda la creación de diseños responsivos que se adaptan a diferentes tamaños de pantallas. Se nota un gran avance también en la seguridad de dichas aplicaciones. Los desarrolladores en la actualidad pueden usar HTTPS[[3]](#footnote-3) para cifrar las comunicaciones entre los servidores y el navegador del usuario, lo que garantiza que los datos transmitidos sean seguros y privados.

En la actualidad el desarrollo web se ha vuelto una tarea cada vez más compleja debido a la gran cantidad de funcionalidades que se requieren para satisfacer las necesidades de los usuarios. Es por ello que el uso de módulos se ha vuelto imprescindible para el desarrollo de aplicaciones web eficientes y escalables. Un módulo es una unidad de software o colección de definiciones de variables que se utiliza para agregar una funcionalidad específica a una aplicación (Cassá & Ufano, 2012). Los módulos permiten la reutilización de código y la modularidad en el diseño de software, lo que facilita su mantenimiento y evolución. En el contexto de las aplicaciones web, los módulos se utilizan para agregar funcionalidades como autenticación, manejo de Bases de Datos, integración con servicios externos, entre otros (Luna et al., 2018).

Por lo tanto, se puede decir que los módulos posibilitan a los desarrolladores enfocarse en la implementación de funcionalidades específicas sin tener que preocuparse por el desarrollo de funcionalidades básicas anteriores que ya han sido resueltas por otros módulos. Esto acelera el proceso y mejora la calidad del software. A modo de aprendizaje se puede tomar como un ejemplo de módulo: el desarrollo de un módulo de integración de redes sociales, el cual permite a los usuarios autenticarse en una aplicación utilizando sus cuentas de redes sociales y compartir contenido en sus perfiles (Paredes Colmenar, 2020).

El soporte tecnológico es una de las áreas más importantes en cualquier organización que utiliza tecnología para sus operaciones diarias. La rápida evolución de la tecnología ha llevado a una mayor complejidad en los sistemas y en las aplicaciones, lo que ha aumentado la cantidad de incidencias que pueden surgir.

Estas incidencias influyen en la obtención de un impacto significativo en la productividad y la eficiencia de la organización, por lo que es fundamental contar con un equipo de soporte tecnológico competente y eficiente. En este sentido el soporte tecnológico se ha vuelto cada vez más importante en la actualidad debido a la creciente dependencia de las organizaciones de la tecnología para sus operaciones diarias. Las incidencias tecnológicas pueden variar desde problemas menores, como la configuración de una impresora, hasta problemas de mayor envergadura, como una falla en el servidor que afecta a toda la organización (Vilchez Velasquez, 2022). A modo de ejemplo, como incidencias tecnológicas más comunes se puede encontrar:

* La seguridad de la información (con la creciente cantidad de datos sensibles que se manejan en las organizaciones, la seguridad de la información se ha vuelto una preocupación crítica).
* Los equipos de soporte técnico deben asegurarse de que los sistemas y aplicaciones estén debidamente protegidos contra ataques cibernéticos y vulnerabilidades de seguridad, falta de compatibilidad entre sistemas y aplicaciones. Las organizaciones utilizan una variedad de sistemas y aplicaciones diferentes para sus operaciones diarias, y, a menudo, estos sistemas no están diseñados para trabajar en conjunto. Esto puede llevar a problemas de integración y dificultades en el intercambio de datos entre sistemas.
* La disponibilidad y el rendimiento. Los sistemas y aplicaciones deben estar disponibles en todo momento para garantizar la continuidad del negocio. Además, deben tener un rendimiento óptimo para garantizar la eficiencia y la productividad de la organización.

En la Universidad de las Ciencias Informáticas (UCI) existe un Centro de Soporte Tecnológico, cuya misión es “*Brindar el servicio de soporte técnico a las aplicaciones y servicios informáticos desarrollados por la Universidad de las Ciencias Informáticas con calidad y eficiencia a partir de una correcta gestión de los mismos*” (https://soporte.uci.cu/nosotros). El Centro de Soporte Tecnológico se inserta en el proceso de desarrollo de software de la UCI luego de terminar el proceso de calidad ejecutados por la Dirección de Calidad subordinada a la Vicerrectoría de Producción. (Tengo duda sobre cómo elaborar la parte anterior. Recuerdo que me dijo que separara algo así como el servicio de soporte de los centros de la UCI, no me acuerdo muy bien)

El Centro de Soporte contaba anteriormente con la plataforma ServiceDesk. Esta plataforma es privativa y la licencia de su uso expiró hace algún tiempo sin posibilidad de nuevo pago para mantenerla. Ante tal situación, se decidió migrar para Xamac(No se preocupe por esto, soy yo que tengo que buscar bien como se escribe el nombre de la plataforma), una plataforma gratuita y enfocada en software libre pero limitada en la gestión de incidencias tecnológicas por falta de requisitos que si presentaba la plataforma inicial.

Actualmente el Centro cuenta, en dicha plataforma, dentro de sus tecnologías y aplicaciones, con un módulo de gestión de incidencias para lograr, mediante las opiniones y aclaraciones del cliente, la obtención de los distintos defectos y errores de los disímiles softwares desarrollados por la Universidad. Dicho módulo carece de una de las herramientas que mejor uso e importancia presenta para responder la mejor solución a las incidencias, la Base de conocimientos. Esta herramienta garantizaría, mediante un Manual, entre otras opciones, que el centro de desarrollo (que creó el sistema o aplicación) brinda al Centro de Soporte Tecnológico, la obtención de conocimientos de todo el sistema al que se le está realizando el Soporte. Conocimientos como (Proceso de instalación del sistema), (Funcionalidad de algún botón o apartado), etc. Por consiguiente, tampoco se observa una adición de las distintas soluciones a los diferentes problemas a la base de datos o Base de Conocimientos elaborada, siendo esto prioridad, pues garantizaría dar respuesta a problemas similares antes ocasionados con una mayor rapidez y eficacia.

La gestión de incidencias no debe confundirse con la gestión de problemas, pues a diferencia de esta última, no se preocupa de encontrar y analizar las causas subyacentes a un determinado incidente sino exclusivamente a restaurar el servicio (López Vargas & Vázquez Chávez, 2016). Resulta ser que a medida que pasa el tiempo también aumentan las tecnologías y cambian los requisitos que se necesita brindar al cliente. Por lo que daría paso a la importancia de realizar una actualización de dicho módulo y adaptarlos a los nuevos requisitos.

Por lo anteriormente planteado se puede expresar como **problema de investigación** ¿Cómo contribuir al fortalecimiento de los servicios que brinda el Centro de Soporte Tecnológico de la Universidad a partir de la gestión de incidencias?

Se define como **objeto de estudio** el proceso de gestión de incidencias tecnológicas de soporte. Señalándose a su vez como **Campo de Acción** la gestión de incidencias tecnológicas en el Centro de Soporte de la Universidad de las Ciencias Informáticas.

Teniendo en cuenta el problema de investigación anteriormente señalado se enmarca como **objetivo general:** Desarrollar un módulo de gestión de incidencias tecnológicas de soporte basado en nuevos requisitos y conteniendo nuevas funcionalidades para el fortalecimiento de los servicios del Centro de Soporte Tecnológico de la Universidad.

Para la realización de la solución, se consideran las siguientes **preguntas científicas:**

1. ¿Cuáles son los referentes teóricos fundamentales que sustentan la investigación relacionada con el proceso de gestión de incidencias tecnológicas de soporte?
2. ¿Cuáles son las características y funcionalidades que debe cumplir el módulo de gestión de incidencias tecnológicas de soporte en el sistema del Centro de Soporte de la Universidad de las Ciencias Informáticas?
3. ¿Cómo validar el funcionamiento del módulo de gestión de incidencias tecnológicas de soporte para el Centro de Soporte de la Universidad de las Ciencias Informáticas?

Las principales **Tareas de Investigación** que se proponen para poder cumplir con el objetivo general son:

* Sistematización de los referentes teóricos relevantes que sustentan la investigación relacionada con el proceso de gestión de incidencias tecnológicas de soporte para el Centro de Soporte de la Universidad de las Ciencias Informáticas.
* Implementación de las funcionalidades del módulo de gestión de incidencias tecnológicas de soporte para el Centro de Soporte de la Universidad de las Ciencias Informáticas.
* Validación de las funcionalidades del módulo de gestión de incidencias tecnológicas de soporte para el Centro de Soporte de la Universidad de las Ciencias Informáticas.

Los **métodos de investigación** que se utilizaron para lograr cumplir con los objetivos específicos y las tareas de investigación propuestas fueron:

**Teóricos**

Analítico-Sintético: Se utiliza el mismo para lograr un análisis y una más eficiente comprensión sobre la documentación de la gestión de incidencias tecnológicas, permitiendo una extracción de los elementos fundamentales del objeto de estudio.

Histórico-Lógico: Presenta un gran auge pues se hizo uso de este método para la comprensión de los antecedentes de la gestión de incidencias y de otros temas en relación con este. Comparándolas con las tendencias actuales.

Preguntar a la tutora cual era el otro método que quería que adicionara, el cual vamos a utilizar luego de hecho el módulo para ver ci cumple con lo que se quiere

**Empíricos**

Entrevista: Se utilizó este método para la realización de entrevistas a especialistas y directivos del Centro de Soporte de la Universidad de Ciencias Informáticas, los cuales aportaron elementos de gran auge en la investigación.

Observación: Gracias a su utilización se logró percibir como se manifiesta todo el proceso por el que está compuesto el objeto de estudio. Elaboración del diagnóstico acerca de la realización del módulo propuesto.

# CAPÍTULO I: Fundamentos y referentes teórico-metodológicos sobre el Módulo de gestión de incidencias tecnológicas

En este capítulo se expresarán toda la fundamentación teórica del módulo de gestión de incidencias tecnológicas. Lo primero será visualizar los diferentes conceptos esenciales asociados al objeto de estudio, para tener un conocimiento más enmarcado sobre el mismo. Se deja a resaltar un estudio de homólogos nacionales e internacionales por el cual se realizará una comparación sobre sus tecnologías y procesos en acción. Se dejará plasmado el estado actual del objeto de estudio de la investigación, sobresaltando el porqué es necesario realizar una actualización del módulo de incidencias tecnológicas. Quedará expuesta la metodología de desarrollo de software por la cual será apoyada el proyecto. A partir de esto se plantearán las diferentes tecnologías y características de las mismas para llevar a cabo desde un inicio la realización del módulo propuesto.

## I.1 Conceptos fundamentales

**Incidencia:**

Una incidencia se refiere a un evento o situación que interrumpe o afecta el funcionamiento normal de una actividad o sistema. En el contexto de la gestión de servicios de tecnología de la información, una incidencia se define como una interrupción no planificada de un servicio, o reducción en la calidad de un servicio (TEAM, s. f.)

En otros contextos, como en el ámbito de la salud, una incidencia puede referirse a un evento adverso o una compilación relacionada con la atención médica. (Fernández-Montalvo & Echeburúa, 2003).

A partir de estos conceptos, en el trabajo de diploma se tomará a una incidencia como un error, evento contradictorio o acontecimiento de desacierto en el código o funcionalidad de un servicio que traiga consigo consecuencias graves en la calidad. En resumen, el término incidencia se utiliza en diferentes contextos para referirse a situaciones o eventos que afectan negativamente el funcionamiento normal de una actividad o sistema, y es importante identificar y gestionar las incidencias de manera efectiva para minimizar su impacto y restaurar el servicio lo antes posible.

**Módulo:**

Un módulo puede definirse como una unidad funcional o de procesamiento independiente que forma parte de un sistema más grave (Dumas et al., 2018).

Los módulos se utilizan comúnmente en la informática y la ingeniería de sistemas para diseñar sistemas complejos y modularizar el proceso de desarrollo (Dumas et al., 2018). Un módulo también puede referirse a una unidad de medida o un bloque de información dentro de un sistema, como en el procesamiento digital de señales (Gómez, 2017). Visto desde el contexto de programación, un módulo (o biblioteca) es una colección de definiciones de variables, funciones y tipos (entre otras cosas) que pueden ser importadas para ser usadas desde cualquier programa(Sanz, 2016). Del concepto de módulo se deriva el de modularidad**.** Se conoce a esto, como la capacidad de un sistema para ser atendido como la unión de varios elementos que se vinculan entre sí y que resultan solidarios (cada uno cumple con una tarea en pos de un objetivo en común) (Vidal-Silva et al., 2019).

A modo de resumen el término módulo se utiliza para referirse a una unidad funcional o de procesamiento independiente dentro de un sistema más grande, y se utiliza comúnmente en la informática y la ingeniería de sistemas para modularizar el proceso de desarrollo y diseñar sistemas complejos.

**Soporte:**

En términos generales, el soporte se refiere a la ayuda o asistencia que se brinda a una persona, organización o sistema para resolver problemas o alcanzar objetivos. En el contexto de la informática y la tecnología de la información, el soporte se refiere a la asistencia técnica que se brinda a los usuarios para ayudarles a resolver problemas con sus dispositivos, software o servicio de TI (López Vargas & Vázquez Chávez, 2016). El soporte también puede incluir aspectos emocionales y psicológicos, como el soporte emocional que se brinda a los pacientes con enfermedades crónicas (Juanes et al., 2000). A partir de estos conceptos, y enfocándose en el objeto de estudio se llega a la conclusión de que el soporte es la asistencia que se le otorga a un sistema de software para resolver errores y problemas o alcanzar objetivos.

Se puede decir que el concepto de **soporte técnico** se deriva del mismo soporte, pero en un enfoque más empresarial. Este estipula ser el servicio que provee una empresa a sus clientes, con el fin que reciban ayuda acerca del uso de sus productos, ya sean físicos o digitales (López Vargas & Vázquez Chávez, 2016). Puede estar directamente proporcionado por la compañía o por agentes externos especializados.

Cabe resaltar las importantes funciones que realiza el soporte técnico:

* Identificar problemas de software o hardware, así como realizar instalaciones y configuraciones exitosas de los mismos.
* Dar mantenimiento regular a los productos, sobre todo cuando hay una garantía de por medio.
* Realizar diagnósticos de las fallas en un producto o un servidor.
* Hablar a los clientes de la forma adecuada para que comprendan cuales son los pasos a seguir.
* Documentar el procedimiento del soporte por medio de una bitácora.
* Avisar a los clientes del estado de resolución de sus peticiones.
* Recibir comentarios acerca de la calidad del servicio y poner en prácticas las mejoras que surjan de esto.

## I.2 Sistemas de Gestión de Incidencias Tecnológicas nacionales e internacionales:

Actualmente existen varias opciones de sistemas homólogos internacionales de módulos de gestión de incidencias tecnológicas que pueden ser utilizados para mejorar la gestión de incidencias en una organización. Cada una de estas plataformas tiene sus propias características por lo que es importante evaluarlas cuidadosamente.

**Homólogos Internacionales:**

**ServiceNow:** ServiceNow es una plataforma de gestión de servicios empresariales en la nube que incluye un módulo de gestión de incidencias tecnológicas. Permite la gestión colaborativa de incidencias, seguimiento de incidencias, automatización de procesos y análisis de datos para la toma de decisiones.

**Jira Service Management:** Jira Service Management es un software de gestión de servicios empresariales que también incluye un módulo de gestión de incidencias. Dentro de sus características se encuentra permitir la gestión de incidencias, solicitudes de servicio y problemas, la automatización de flujos de trabajo y la integración con otras herramientas de desarrollo de software.

**BMC Helix ITSM:** BMC Helix ITSM es una plataforma de gestión de servicios empresariales que incluye un módulo de gestión de incidencias tecnológicas. Permite la gestión de incidencias, problemas, cambios y activos, también resalta la automatización de flujos de trabajo y la integración con otras herramientas de gestión de servicios.

**Zendesk:** Zendesk es una plataforma de gestión de servicios empresariales que incluye un módulo de gestión de incidencias tecnológicas. Y como las anteriores también permite la gestión de incidencias, solicitudes de servicios y problemas, presenta una automatización de flujo de trabajo y una de las características más común hasta ahora es la integración con otras herramientas de gestión de servicios.

**Homólogos Nacionales:**

Aunque la información sobre sistemas empresariales que utilizan un módulo de gestión de incidencias tecnológicas en Cuba es limitada, se tuvo en cuenta la investigación de homólogos también nacionalmente, logrando llegar al resultado de tener en cuenta 3 sistemas que llevan a cabo dicho proceso.

**Infomed****:**  Infomed es una red de información de salud en Cuba que ofrece una amplia gama de servicios en línea como información médica y científica, publicaciones especializadas, bases de datos, herramientas de búsquedas, cursos en línea, foros de discusión y redes sociales a profesionales de la salud, científicos y estudiantes. Fue creada en 1992 como una iniciativa del Ministerio de Salud Pública de Cuba para mejorar la comunicación y el intercambio de información en el sector de la salud en Cuba. Su objetivo principal es mejorar la calidad de la atención médica en Cuba y fomentar la investigación y la educación en el campo de la salud. Presenta en su plataforma un sistema de gestión de incidencias tecnológicas para administrar problemas técnicos y reportes de errores en la misma. Esto ayuda a garantizar que los usuarios tengan acceso sin interrupciones a los servicios y la información que necesitan.

**CubaTronik:** CubaTronik es una empresa de tecnología en Cuba que ofrece soluciones tecnológicas personalizadas a empresas y organizaciones. Fue creada en 2014 como una iniciativa del Ministerio de la Informática y las Comunicaciones de Cuba. Presenta como objetivos principales desarrollar soluciones tecnológicas personalizadas para empresas y organizaciones, fomentar la innovación y el desarrollo en el campo de la tecnología en Cuba, y mejorar la calidad de vida de los ciudadanos cubanos a través de la tecnología. Dentro de sus servicios se incluye el desarrollo de software, diseño de sitios web, análisis de datos, consultoría en tecnología, soluciones de seguridad y soluciones de comunicación para fomentar el desarrollo de la tecnología en Cuba. Cuenta con un sistema de gestión de incidencias tecnológicas para administrar problemas técnicos y reportes de errores en sus soluciones. Esto proporciona una ayuda para garantizar que los clientes puedan usar sus soluciones sin interrupciones y que se resuelvan rápidamente los problemas técnicos que puedan surgir.

**Etecsa:** Etecsa es la empresa de telecomunicaciones de Cuba y es responsable de proporcionar servicios de telefonía, internet y otras tecnologías de la información en Cuba. Fue creada en 1994 como una empresa estatal de telecomunicaciones. Dentro de sus objetivos principales se encuentran: Proporcionar servicios de telecomunicaciones de alta calidad en Cuba, fomentar la innovación y el desarrollo en el campo de las telecomunicaciones, y mejorar la conectividad y el acceso de la información en Cuba. Dentro de sus servicios principales están enmarcados la telefonía móvil, telefonía fija, internet, correo electrónico, mensajería y servicios de valor agregado. Sus servicios son accesibles para todos los ciudadanos cubanos y los visitantes extranjeros también. Tal y como es tendencia entre las anteriores mencionadas, también presenta un sistema de gestión de incidencias tecnológicas para administrar problemas técnicos y reporte de errores en sus servicios. Hace uso de la aplicación web AvilaQuid desarrollada en la provincia de Ciego de Ávila por la empresa DESOFT para toda esta gestión de incidencias. Garantizando así que los usuarios puedan usar sus servicios sin interrupciones y que se resuelvan rápidamente los problemas técnicos que puedan surgir.

**I.2.1 Resultados de comparación obtenida por el estudio de homólogos:**

En este epígrafe se elabora una comparación entre los diferentes sistemas homólogos tanto nacionales como internacionales, que realizan una gestión de incidencias dentro de sus procesos, teniendo en cuenta la guía de aspectos que son necesidades del cliente. Las principales características por las que se hacen la comparación son:

Presentar un carácter de software libre en caso de ser un sistema desarrollador de software o utilizar software libre en la mayoría de sus aplicaciones, o específicamente, en la gestión de incidencias tecnológicas.

Otro de los puntos a tener en cuenta es: Presentar una actualización de las tecnologías en las que se apoya el sistema. Muy necesario este aspecto pues con el desarrollo de las tecnologías, es relevante adquirir una actualización de las mismas para permitir una mayor integración, eficacia y eficiencia de las aplicaciones y servicios desarrollados.

Se tendrá en cuenta también que contengan una base de conocimientos. Significa específicamente en este caso, que el proceso de gestión de incidencias que presenta el sistema homólogo permita que, mediante un Manual, que es requerido para cada sistema se le pueda otorgar respuesta a las incidencias. Puede tener anexado a la base de datos soluciones ya antes dadas por las que pudiera guiarse también.

Por último, se observará como característica que presenten uso de la metodología ITIL todos los procesos de Soporte y más inmiscuidos dentro de la gestión de incidencias tecnológicas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Homólogos/ Necesidades del cliente** | Carácter de software libre | Tecnologías actualizadas a partir de 2021 | Contención de una Base de Conocimientos | Uso de la Metodología ITIL |
| ServiceNow | **Carácter privativo** | **Tecnologías actualizadas** | **Si** | **Hace uso de la metodología** |
| Jira Service Management | **Carácter privativo** | **Tecnologías actualizadas** | **Si** | **No soporta el flujo adecuado de ITIL 2011 a las incidencias** |
| BMC Helix ITSM | **Carácter privativo** | **Tecnologías actualizadas** | **Si** | **Hace uso de la metodología** |
| Zendesk | **Carácter privativo** | **Tecnologías actualizadas** | **Si** | **Hace uso de la metodología** |
| Infomed | **El código fuente es de acceso público pero su uso y distribución están limitados por las políticas del gobierno de Cuba.** | **Hasta hace 2 años aún se usaban PHP.v7 y Java.v8, actualmente están parcialmente actualizadas** | **No** | **Hace uso de la metodología** |
| CubaTronik | **No publica su código fuente de manera abierta y no permite la libre distribución y modificación de sus soluciones,** | **Tecnologías actualizadas** | **Si** | **Hace uso de la metodología** |
| ETECSA (AvilaQuid) | **Utiliza software libre en algunos de sus sistemas y aplicaciones, por lo que el código fuente de esos sistemas se encuentra disponible.** | **Tecnologías actualizadas** | **Si** | **No soporta los procesos de ITIL 2011** |

## 

## I.3 Gestión de Incidencias Tecnológicas:

La gestión de incidencias tecnológicas se refiere al proceso de identificar, registrar, priorizar, investigar y resolver los problemas relacionados con los servicios de Tecnología de la Información (ITIL, 2019).

Presenta como objetivo principal: Minimizar el impacto negativo de los incidentes de las Tecnologías de la Información (TI) en el negocio y en los usuarios finales (ManageEngine, 2021).

Se pueden observar numerosas ventajas con respecto a la gestión de Incidencias. De estas se pueden mencionar:

1. Mejora la eficiencia y la rapidez en la resolución de incidentes.
2. Reduce el impacto negativo de los incidentes en el negocio.
3. Mejora la satisfacción del usuario final al proporcionar una comunicación clara y transparente.
4. Identifica y aborda las causas subyacentes de los incidentes para prevenir su recurrencia.

Dejando enmarcadas las ventajas, se debe tener noción también, que esta gestión de incidencias conlleva consecuencias en varias ocasiones. Ejemplos de estas desventajas serían:

* Costos asociados con la implementación y el mantenimiento de un sistema de gestión de incidentes.
* Curva de aprendizaje para los usuarios, técnicos y administradores que utilizan el sistema.
* Puede requerir la colaboración de múltiples departamentos y equipos, lo que puede ser un desafío en organizaciones grandes y complejas.

En la actualidad, la Universidad de Ciencias Informáticas cuenta con un Centro de Soporte, el cual brinda el servicio de soporte técnico a las aplicaciones y servicios informáticos desarrollados. El Centro trabajaba con la plataforma privativa ServiceDesk, pero como es de pago, culmino el rango de tiempo de la licencia y tuvieron que cambiar para Xamac, siendo esta la escogida por sus ventajas de ser gratuita y enfocada en software libre. Dicho Centro cuenta dentro de su plataforma con un módulo de gestión de incidencias tecnológicas de soporte, con el objetivo, de administrar problemas contenidos dentro de las disímiles aplicaciones expandidas a la población y gestionar el reporte de errores dentro de sus servicios.

Como modelo de referencia, mediante entrevistas realizadas a directivos del Centro de Soporte, se logró acreditar que la gestión de incidencias tecnológicas en el Centro de Soporte de la Universidad de las Ciencias Informáticas funciona de la siguiente manera:

* Se registra como primer paso el Registro de la Incidencia: En este paso, los usuarios pueden reportar una incidencia a través de una plataforma de soporte en línea, un correo electrónico o una llamada telefónica. El Centro de Soporte de la UCI registra la incidencia y le asigna un número de seguimiento único.
* Posteriormente se avanza a la Priorización de la Incidencia: El Centro de Soporte de la UCI utiliza un sistema de priorización para determinar la gravedad de la incidencia y la urgencia de su resolución. Esto ayuda a los encargados de la gestión de incidencias a determinar cuáles deben ser atendidas primero.
* Luego se avanza a la Asignación de la Incidencia: Al cumplirse el paso anterior, el Centro de Soporte de la UCI asigna la incidencia automáticamente al equipo de Soporte correspondiente. Se puede tener en cuenta la notificación al usuario que reportó la incidencia, del estado actual y del equipo de soporte que se encuentra desarrollando la solución.
* Posteriormente se realiza un Análisis y Diagnóstico: El equipo previamente asignado comienza a trabajar en la incidencia y realiza un análisis y diagnóstico detallado para determinar la causa raíz del problema.
* De manera siguiente se procede a la Resolución de la Incidencia: Una vez que se ha identificado la causa raíz del problema, el equipo de soporte correspondiente trabaja para resolver la incidencia y realizar las pruebas necesarias para asegurarse de que el problema ha sido solucionado.
* Como paso final se encuentra el Cierre de la Incidencia: Una vez que se ha resuelto la incidencia, el equipo de soporte cierra el registro de la misma en la plataforma de soporte en línea. También se notifica al usuario que reportó la incidencia de su resolución, y se le solicita que proporcione comentarios sobre su experiencia con los nuevos resultados.

Luego de verificar la información obtenida por la investigación a los sistemas homólogos y teniendo en cuenta que el módulo de gestión de incidencias del Centro de Soporte no presenta dentro de sus requisitos una base de conocimientos, siendo la misma una muy buena práctica para incidir positivamente en la resolución de las incidencias, se hace necesario una actualización de dicho módulo.

Tampoco se evidencia la adición de las diferentes soluciones a distintos sistemas, a su respectiva base de conocimientos para así tener mayor información al respecto y por consiguiente brindar la mejor solución a la incidencia. Queda plasmado el porqué es necesario realizar una actualización del módulo de incidencias tecnológicas, pues garantizaría el cumplimiento de todo el proceso descrito en líneas anteriores de forma eficiente, que, a su vez, proporcionaría mayor seguridad y eficacia a la hora de expandir aplicaciones desarrolladas en la universidad con una correcta gestión de errores y con calidad.

## I.4 Metodología, herramientas y tecnología

**I.4.1 Metodología de desarrollo a utilizar: AUP-UCI: Escenario 4**

La Universidad de las Ciencias informáticas desarrolló una versión de la metodología de desarrollo de software AUP (Proceso Ágil Unificado), con el fin de crear una metodología que se adapte al ciclo de vida definido por la actividad productiva de la Universidad. Esta versión decide mantener para el ciclo de vida de los proyectos la fase de Inicio, pero modificando el objetivo de la misma y se unifican las restantes fases de la metodología de desarrollo de software AUP en una sola, nombrada Ejecución y agregándose también una nueva fase llamada Cierre (Alfonso Benítez, 2017).

La metodología AUP-UCI propone 3 variantes para el modelado del negocio, CUN (Casos de Uso del Negocio), DPN (Descripción del Proceso de Negocio) y MC (Modelo Conceptual), y contiene 3 formas de encapsular los requisitos, CUS (Casos de Uso del Sistema), HU (Historias de Usuarios), DRP (Descripción de Requisitos por Procesos). Por estas características está dividida la metodología en 4 escenarios (Tamara, 2015). En este trabajo de diploma se hará uso de la metodología AUP-UCI en su escenario 4, pues el mismo se utiliza en proyectos que no modelen el negocio y se utiliza para el modelado del sistema HU (Historia de Usuario).

**I.4.2 Lenguaje de Modelado:**

**UML v2.5.1:**

UML (Lenguaje Unificado de Modelado) se trata de un estándar que se ha adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollos de software. Es una notación de modelado visual, que utiliza diagramas para mostrar distintos aspectos de un sistema (Fontela, 2012). UML es apto para cualquier sistema, pero su mayor difusión y sus principales virtudes se advierten en el campo de los sistemas de software.

**I.4.3 Herramienta CASE:**

**Visual Paradigm v17.0:**

Visual Paradigm es una herramienta de modelado de software que se utiliza para crear diagramas UML, diagramas de flujo de datos, diagramas de entidad-relación y otros tipos de diagramas de diseño. Presenta importantes características que benefician a los desarrolladores como lo es: contener una amplia gama de plantillas, colaboración en tiempo real, generación de código e integración con otras herramientas como lo son Eclipse, NetBeans y Microsoft Office. Visual Paradigm se utiliza principalmente en el desarrollo de software para ayudar a los equipos a visualizar y diseñar sistemas complejos. Presenta auge también en la planificación y gestión de proyectos.

**I.4.4 Visual Studio Code v1.76:**

Visual Studio Code es un editor de código fuente desarrollado por Microsoft que se utiliza principalmente para la programación de software. Algunas de sus principales características son: Posee una amplia gama de extensiones para diferentes lenguajes de programación y tecnologías, presenta una depuración integrada, está en relación integrada con Git, presenta un soporte multiplataforma, pues está disponible para Windows, macOS y Linux. Visual Studio Code se utiliza principalmente en el desarrollo de software para ayudar a los programadores a escribir, depurar y mantener su código.

**I.4.5 Django v4.2.5:**

Django es un popular framework de desarrollo web en Python, siendo el mismo flexible, potente y seguro. Presenta como características:

* Utiliza el patrón de arquitectura MVC (Modelo-Vista-Controlador) para separar la lógica de negocio de la interfaz de usuario y la gestión de datos.
* Incluye un ORM (Object-Relational Mapping) que permite trabajar con bases de datos de manera más sencilla y eficiente.
* Django presenta un enrutador de URL incorporado que puede manejar las solicitudes del cliente y dirigirlas a las vistas apropiadas.
* Django proporciona un administrador web incorporado que permite gestionar fácilmente los datos de la aplicación.
* Incorpora medidas de seguridad para proteger las aplicaciones web contra vulnerabilidades comunes, como XSS y CSRF.
* Cuenta con una gran comunidad de desarrolladores que contribuyen con su código y brindan soporte a otros desarrolladores.

**I.4.7 Bootstrap v5.1:**

Bootstrap es un framework de diseño web de código abierto que se utiliza para crear sitios web y aplicaciones móviles responsivos y con un diseño atractivo. Sus características principales son: Diseño responsivo, cuenta con una amplia variedad de componentes predefinidos, permite su personalización por el desarrollador en cuestión, cuenta con una documentación completa y detallada y es compatible con una amplia variedad de navegadores. Bootstrap es una herramienta muy útil para los desarrolladores web que buscan crear sitios web modernos y atractivos de manera rápida y eficiente.

**I.4.8 DB Browser (SQLite) v3.12.2:**

**Lenguajes de programación:**

**I.4.9 Python v3.11.4:**

Python es un lenguaje de programación de código abierto y gratuito que se utiliza principalmente para crear aplicaciones web dinámicas y sitios web interactivos. Sus principales características son: Es un lenguaje de programación del lado del servidor, relativamente fácil de aprender y utilizar, presenta gran cantidad de recursos disponibles y se integra fácilmente con otras tecnologías web como HTML, CSS, JavaScript y bases de datos. Por todas las características antes mencionadas es un lenguaje de programación popular entre los desarrolladores.

**I.4.10 SQLite v3.35.5:**

MySQL es un sistema de gestión de Bases de Datos de código abierto y gratuito que se utiliza para almacenar y gestionar datos en línea. Algunas de sus características principales son: Es altamente escalable y puede gestionar grandes volúmenes y tráfico de datos, es conocido por su velocidad y eficiencia en la gestión y recuperación de datos, es muy confiable y se utiliza en una amplia variedad de aplicaciones empresariales y de misión crítica. MySQL se utiliza principalmente para almacenar y gestionar datos en línea.

**I.4.11 JavaScript:**

JavaScript es un lenguaje de programación interpretado, lo que significa, que no necesita ser compilado antes de ser ejecutado. Concede la creación de objetos y la definición de clases. Una de sus principales características es ser un lenguaje dinámico, permitiendo que las variables puedan cambiar su tipo en tiempo de ejecución. Puede ser usado en diferentes sistemas operativos y navegadores web otorgando una visión multiplataforma. Es utilizado principalmente en el desarrollo web.

## 

## Conclusiones del capítulo

Queda plasmado en este capítulo la situación problemática por la que se encuentra la gestión de incidencias en el Centro de Soporte de la Universidad de las Ciencias Informáticas que, a modo de resumen, el principal problema que se observa es la falta de una Base de Conocimientos, siendo una de las principales funcionalidades por las que se apoyan los módulos investigados mediante el estudio de homólogos realizado. Se absorbieron características referentes a los mismos, que presentan cierto grado de importancia y que se tendrán en cuenta para la actualización del módulo propuesto. Igual, se escogieron las herramientas necesarias para la creación de la respuesta a desarrollar para el problema planteado, haciendo uso de las mismas siempre teniendo en cuenta las características de la metodología escogida, que, en este caso, tal y como está enmarcado en el capítulo, se hará uso de AUP-UCI en su escenario 4.

# CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO

<Introducción del capítulo 2 con una breve explicación del objetivo que persigue el capítulo, los principales contenidos que aborda, la estructura que puede encontrar el lector en su composición y un breve texto introductorio a las temáticas principales que aborda el capítulo>

Este capítulo persigue como objetivo visualizar y diseñar el módulo de gestión de incidencias tecnológicas a través de la utilización de la metodología AUP-UCI en su escenario 4. Se especifican los requisitos funcionales y no funcionales de la propuesta de solución, así como las historias de usuarios correspondientes.

## II.1 Modelado del proceso de gestión de incidencias tecnológicas de soporte

<Generalmente este epígrafe se dedica al modelado de los procesos y subprocesos que componen el objeto de estudio, por lo que debe ser consecuencia del epígrafe del capítulo 1 donde se realizó la descripción de dicho proceso en términos textuales>

## II.2 Requisitos, análisis y diseño del módulo de gestión de incidencias tecnológicas de soporte

**II.2.1 Requisitos Funcionales**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nombre** | **Descripción** | **Prioridad** | **Complejidad** |
| Rf-1 | Autenticar Usuarios |  | Alta | Alta |
| Rf-2 | Manejar Usuarios |  | Media | Media |
| Rf-3 | Crear Base de Conocimientos |  | Alta | Alta |
| Rf-4 | Visualizar Base de Conocimiento |  | Alta | Alta |
| Rf-5 | Modificar Base de Conocimiento |  | Alta | Alta |
| Rf-6 | Eliminar Base de Conocimientos |  | Alta | Alta |
| Rf-7 | Visualizar estados de las incidencias |  | Alta | Alta |
| Rf-8 | Listar las incidencias por su estado asociado |  | Alta | Alta |
| Rf-9 | Crear las incidencias |  | Alta | Alta |
| Rf-10 | Visualizar las incidencias |  | Alta | Alta |
| Rf-11 | Modificar las incidencias |  | Alta | Alta |
| Rf-12 | Eliminar las incidencias |  | Alta | Alta |
| Rf-13 | Cerrar incidencias |  | Alta | Media |
| Rf-14 | Permitir el adjunto de archivos a las Base de Conocimientos |  | Alta | Media |
| Rf-15 | Visualizar el nivel de las incidencias |  | Media | Media |
| Rf-16 | Gestionar la prioridad de las incidencias. |  | Media | Media |
| Rf-17 | Gestionar la organización y sistema asociado a la incidencia |  | Media | Media |
| Rf-18 | Escalar las incidencias a diferente nivel |  | Media | Baja |
| Rf-19 | Asociar las Bases de Conocimientos a las Incidencias |  | Alta | Media |
| Rf-20 | Gestionar el modo de entrada de las incidencias |  | Baja | Baja |

**II.2.2 Requisitos no Funcionales.**

**RnF-1 Usabilidad:**

RnF-1.1 La propuesta de solución debe brindar un acceso fácil y rápido.

RnF-1.2 La interfaz debe ser amigable y sencilla para la comodidad del usuario.

RnF-1.3 Las opciones más usadas serán cómodas de invocarse y presentarán accesos directos.

**RnF-2 Software:**

RnF-2.1 Para la visualización en la Pc cliente y su correcto funcionamiento será necesario Firefox, Google Crome, Opera o Microsoft Edge.

**RnF-3 Hardware:**

RnF-3.1 Será necesario para un funcionamiento óptimo, una Pc con propiedades iguales o superiores a las siguientes:

Procesador: Intel® Celeron® CPU N3050 @ 1.60GHz

Memoria RAM: 512MB

**RnF-4 Restricción del Diseño e Implementación:**

RnF-4.1 Se utilizará Visual Paradigm en su versión 17.0 como herramienta de modelado

RnF-4.2 Se DB Browser (SQLite) en su versión como gestor de Base de Datos.

RnF-4.3 Se utilizará de Django en su versión 4.2.3 o superior para la realización de la solución propuesta.

RnF-4.4 Se utilizará Visual Studio Code en su versión 1.79.2 como IDE para el desarrollo de la propuesta de solución.

**Rnf-5 Seguridad:**

* Confidencialidad:

RnF-5.1 Se encontrarán protegidas las diferentes áreas del módulo contra el acceso no autorizado, dado el uso de roles y grupos de usuarios.

* Integridad:

RnF-5.2 La información solo podrá ser modificada por el personal autorizado.

RnF-5.3 Se tomarán en cuenta validaciones del lado del servidor para evitar ataques de inyección SQL.

**RnF-6 Interfaz de usuario:**

RnF-6.1 El diseño de la interfaz responderá a la ejecución de acciones de forma rápida, realizando los procesos en pocos pasos.

RnF-6.2 La entrada incorrecta de datos será mostrada al usuario resaltando el campo donde se encuentra el error y un texto explicativo de los datos correctos a validar.

RnF-6.3 Las interfaces contendrán los datos estructurados permitiendo una correcta interpretación de la información.

**II.2.3 Historias de Usuarios.**

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **No:** HU-3 | **Nombre del Requisito:** Crear Base de Conocimientos | |
| **Programador:** Roger Díaz Viamonte | | **Iteración Asignada:** 1 |
| **Prioridad:** Alta | | **Tiempo Estimado:** 1dia |
| **Riesgo en Desarrollo:** Alta | | **Tiempo Real:** 0.14 semanas |
| **Descripción:** Cada Incidencia contará, para lograr ser resuelta de forma eficiente y con mayor rapidez, con una base de conocimientos que dentro de sus características está: la de poseer un manual individual para el proyecto en específico en el que se evidencia la incidencia conteniendo información esencial para el uso del software y de sus respectivas funcionalidades. Y la opción de visualizar las soluciones dadas anteriormente para la guía a la hora de dar respuesta a las incidencias a la que está asociada la base de conocimiento.  Para la creación de las Bases de Conocimientos se deben rellenar los siguientes campos:   * **nombreSistema:** Obligatorio. Lista desplegable. Valores (Lista de sistemas). Valores por defecto (1). * **nombreProyecto:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima de 150 caracteres. * **IncidenciasA:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima de 500 caracteres. * **Manual:** Obligatorio. campo de tipo archivo. * **respuestasAnexadas:** Obligatorio. Campo de texto. Permite todos los caracteres. | | |
| **Observaciones:** | | |
| **Prototipo de Interfaz gráfica:** | | |

***Tabla 3: Historia de Usuario 3***

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **No:** HU-4 | **Nombre del Requisito:** Visualizar Base de Conocimiento | |
| **Programador:** Roger Díaz Viamonte | | **Iteración Asignada:** 1 |
| **Prioridad:** Alta | | **Tiempo Estimado:** 1 |
| **Riesgo en Desarrollo:** Alta | | **Tiempo Real:** 1 |
| **Descripción:** Se podrá visualizar las diferentes características o parámetros asociados a la Base de Conocimiento.   * **nombreSistema:** Obligatorio. Lista desplegable. Valores (Lista de sistemas). Valores por defecto (1). * **nombreProyecto:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima de 150 caracteres. * **IncidenciasA:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima de 500 caracteres. * **Manual:** Obligatorio. campo de tipo archivo. * **respuestasAnexadas:** Obligatorio. Campo de texto. Permite todos los caracteres. | | |
| **Observaciones:** | | |
| **Prototipo de Interfaz gráfica:** | | |

***Tabla 4: Historia de Usuario 4***

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **No:** HU-5 | **Nombre del Requisito:** Modificar Base de Conocimiento | |
| **Programador:** Roger Díaz Viamonte | | **Iteración Asignada:** 1 |
| **Prioridad:** Alta | | **Tiempo Estimado:** 1 |
| **Riesgo en Desarrollo:** Alta | | **Tiempo Real:** 1 |
| **Descripción:** Se podrá modificar las características o parámetros asociados a la Base de Conocimiento que haya sido creada con antelación.   * **nombreSistema:** Obligatorio. Lista desplegable. Valores (Lista de sistemas). Valores por defecto (1). * **nombreProyecto:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima de 150 caracteres. * **IncidenciasA:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima de 500 caracteres. * **Manual:** Obligatorio. campo de tipo archivo. * **respuestasAnexadas:** Obligatorio. Campo de texto. Permite todos los caracteres. | | |
| **Observaciones:** | | |
| **Prototipo de Interfaz gráfica:** | | |

***Tabla 5: Historia de Usuario 5***

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **No:** HU-6 | **Nombre del Requisito:** Eliminar Base de Conocimiento | |
| **Programador:** Roger Díaz Viamonte | | **Iteración Asignada:** 1 |
| **Prioridad:** Alta | | **Tiempo Estimado:** 1 |
| **Riesgo en Desarrollo:** Alta | | **Tiempo Real:** 1 |
| **Descripción:** Se podrá eliminar la Base de Conocimiento creada con antelación, eliminándose por consiguiente todas las incidencias a las se le haya anexado esa Base de Conocimiento. | | |
| **Observaciones:** | | |
| **Prototipo de Interfaz gráfica:** | | |

***Tabla 6: Historia de Usuario 6***

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **No:** HU-9 | **Nombre del Requisito:** Crear las Incidencias | |
| **Programador:** Roger Díaz Viamonte | | **Iteración Asignada:** 1 |
| **Prioridad:** Alta | | **Tiempo Estimado:** 1 |
| **Riesgo en Desarrollo:** Alta | | **Tiempo Real:** 1 |
| **Descripción:** El usuario autenticado en la plataforma podrá crear la incidencia y hacerle corresponder las diferentes características que la conforman.   * **Título:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima de 100 caracteres. * **Cliente:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima de 50 caracteres. * **Descripción:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima de 500 caracteres. * **Entrada:** Obligatorio. Lista desplegable. Valores (Tipos de entradas). Valores por defecto (1). * **BaseA:** Lista desplegable. Valores (Lista de bases de conocimientos). * **Estado:** Obligatorio. Lista desplegable. Valores (Tipos de estado). Valores por defecto (1). * **Nivel:** Obligatorio. Lista desplegable. Valores (Tipos de nivel). Valores por defecto (1). * **Prioridad:** Obligatorio. Lista desplegable. Valores (Tipos de prioridad). Valores por defecto (1). * **Sistema:** Obligatorio. Lista desplegable. Valores (Tipos de sistema). Valores por defecto (1). * **Grupo:** Obligatorio. Lista desplegable. Valores (Tipos de grupos). Valores por defecto (1). * **Organización:** Obligatorio. Lista desplegable. Valores (Tipos de organización). Valores por defecto (1). * **Responsable:** Lista desplegable. Valores (Lista de bases de conocimientos). * **Respuesta:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima de 500 caracteres. | | |
| **Observaciones:** | | |
| **Prototipo de Interfaz gráfica:** | | |

***Tabla 9: Historia de Usuario 9***

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **No:** HU-10 | **Nombre del Requisito:** Visualizar las Incidencias | |
| **Programador:** Roger Díaz Viamonte | | **Iteración Asignada:** 1 |
| **Prioridad:** Alta | | **Tiempo Estimado:** 1 |
| **Riesgo en Desarrollo:** Alta | | **Tiempo Real:** 1 |
| **Descripción:** Se podrá visualizar todas las características o parámetros que conforman a la incidencia.   * **Título:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima de 100 caracteres. * **Cliente:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima de 50 caracteres. * **Descripción:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima de 500 caracteres. * **Entrada:** Obligatorio. Lista desplegable. Valores (Tipos de entradas). Valores por defecto (1). * **BaseA:** Lista desplegable. Valores (Lista de bases de conocimientos). * **Estado:** Obligatorio. Lista desplegable. Valores (Tipos de estado). Valores por defecto (1). * **Nivel:** Obligatorio. Lista desplegable. Valores (Tipos de nivel). Valores por defecto (1). * **Prioridad:** Obligatorio. Lista desplegable. Valores (Tipos de prioridad). Valores por defecto (1). * **Sistema:** Obligatorio. Lista desplegable. Valores (Tipos de sistema). Valores por defecto (1). * **Grupo:** Obligatorio. Lista desplegable. Valores (Tipos de grupos). Valores por defecto (1). * **Organización:** Obligatorio. Lista desplegable. Valores (Tipos de organización). Valores por defecto (1). * **Responsable:** Lista desplegable. Valores (Lista de bases de conocimientos). * **Respuesta:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima de 500 caracteres. | | |
| **Observaciones:** | | |
| **Prototipo de Interfaz gráfica:** | | |

***Tabla 10: Historia de Usuario 10***

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **No:** HU-11 | **Nombre del Requisito:** Modificar las incidencias | |
| **Programador:** Roger Díaz Viamonte | | **Iteración Asignada:** 1 |
| **Prioridad:** Alta | | **Tiempo Estimado:** 1 |
| **Riesgo en Desarrollo:** Alta | | **Tiempo Real:** 1 |
| **Descripción:** Se podrán modificar las diferentes características o parámetros asociados a las incidencias creadas con antelación.   * **Título:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima de 100 caracteres. * **Cliente:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima de 50 caracteres. * **Descripción:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima de 500 caracteres. * **Entrada:** Obligatorio. Lista desplegable. Valores (Tipos de entradas). Valores por defecto (1). * **BaseA:** Lista desplegable. Valores (Lista de bases de conocimientos). * **Estado:** Obligatorio. Lista desplegable. Valores (Tipos de estado). Valores por defecto (1). * **Nivel:** Obligatorio. Lista desplegable. Valores (Tipos de nivel). Valores por defecto (1). * **Prioridad:** Obligatorio. Lista desplegable. Valores (Tipos de prioridad). Valores por defecto (1). * **Sistema:** Obligatorio. Lista desplegable. Valores (Tipos de sistema). Valores por defecto (1). * **Grupo:** Obligatorio. Lista desplegable. Valores (Tipos de grupos). Valores por defecto (1). * **Organización:** Obligatorio. Lista desplegable. Valores (Tipos de organización). Valores por defecto (1). * **Responsable:** Lista desplegable. Valores (Lista de bases de conocimientos). * **Respuesta:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima de 500 caracteres. | | |
| **Observaciones:** | | |
| **Prototipo de Interfaz gráfica:** | | |

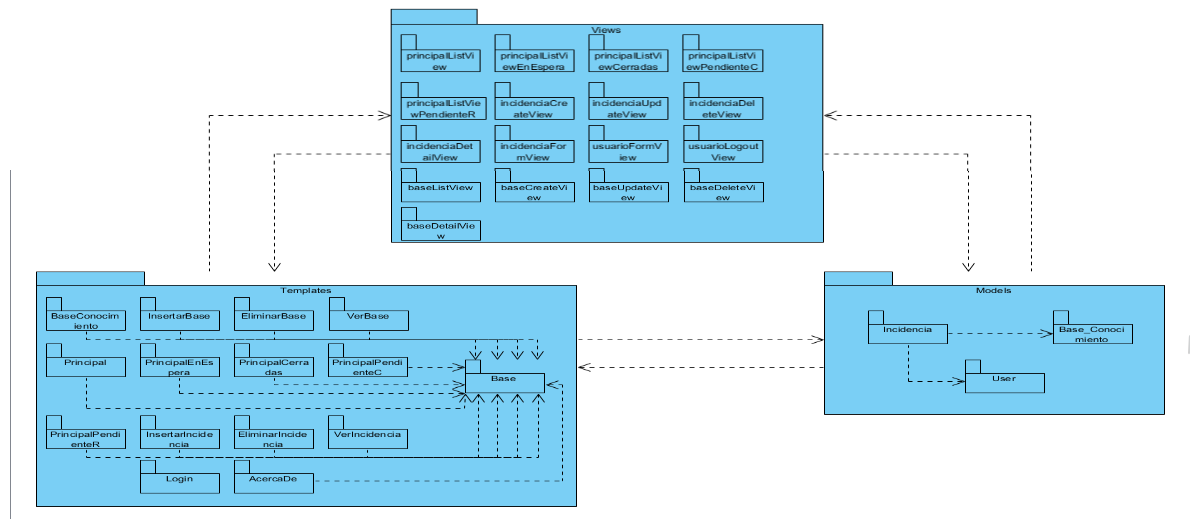
***Tabla 11: Historia de Usuario 11***

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **No:** HU-12 | **Nombre del Requisito:** Eliminar las Incidencias | |
| **Programador:** Roger Díaz Viamonte | | **Iteración Asignada:** 1 |
| **Prioridad:** Alta | | **Tiempo Estimado:** 1 |
| **Riesgo en Desarrollo:** Alta | | **Tiempo Real:** 1 |
| **Descripción:** Se podrán eliminar las incidencias de la Base de Datos junto con sus características asociadas. | | |
| **Observaciones:** La eliminación de incidencias no es lo mismo que el cierre de incidencias, pues la segunda solo requiere de cambiar el estado de la incidencia a (Cerrada). | | |
| **Prototipo de Interfaz gráfica:** | | |

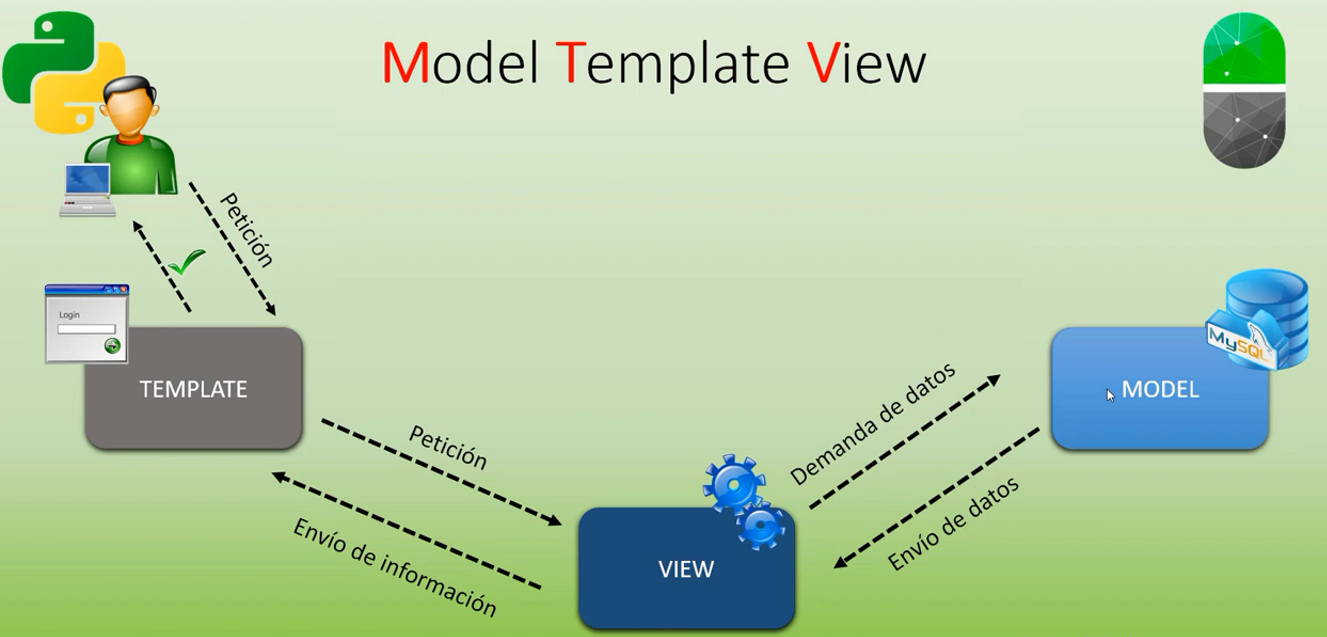
***Tabla 12: Historia de Usuario 12***

**II.2.4 Análisis y Diseño.**

**II.2.4.1 Descripción y representación de la arquitectura:**

****

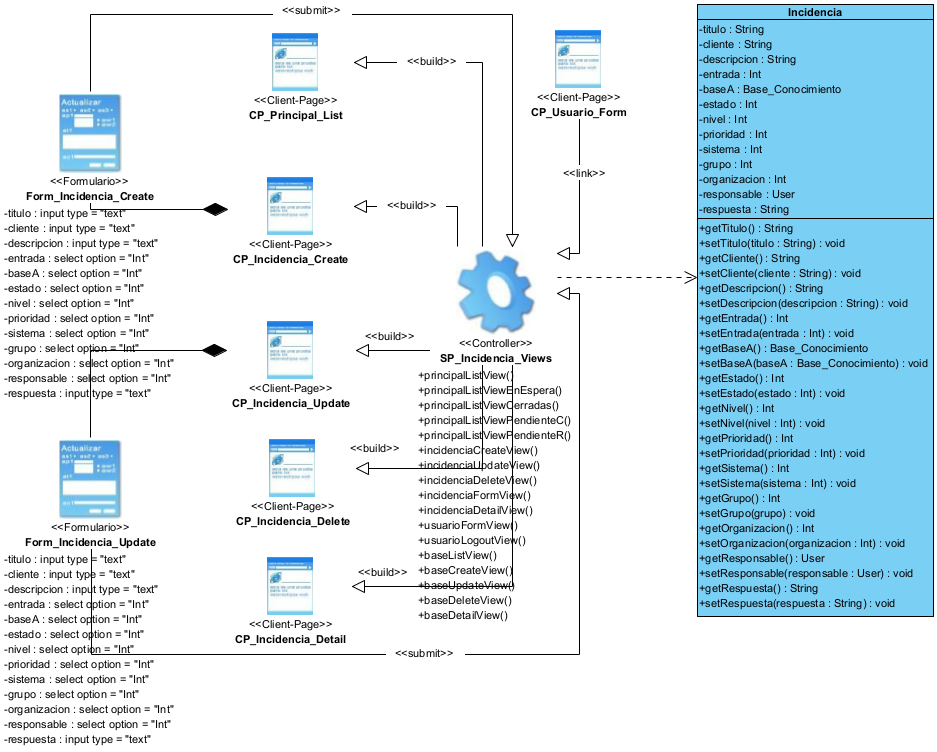
***Figura 1: Diagrama de Paquetes***

****

***Figura 2: Representación de la arquitectura (MTV), presentando la misma filosofía que (MVC)***

**II.2.4.2 Diagrama de Clases del Diseño:**

* **2.4.2.1 Diagrama de Clases del Caso de Uso <Gestionar Incidencia>**

****

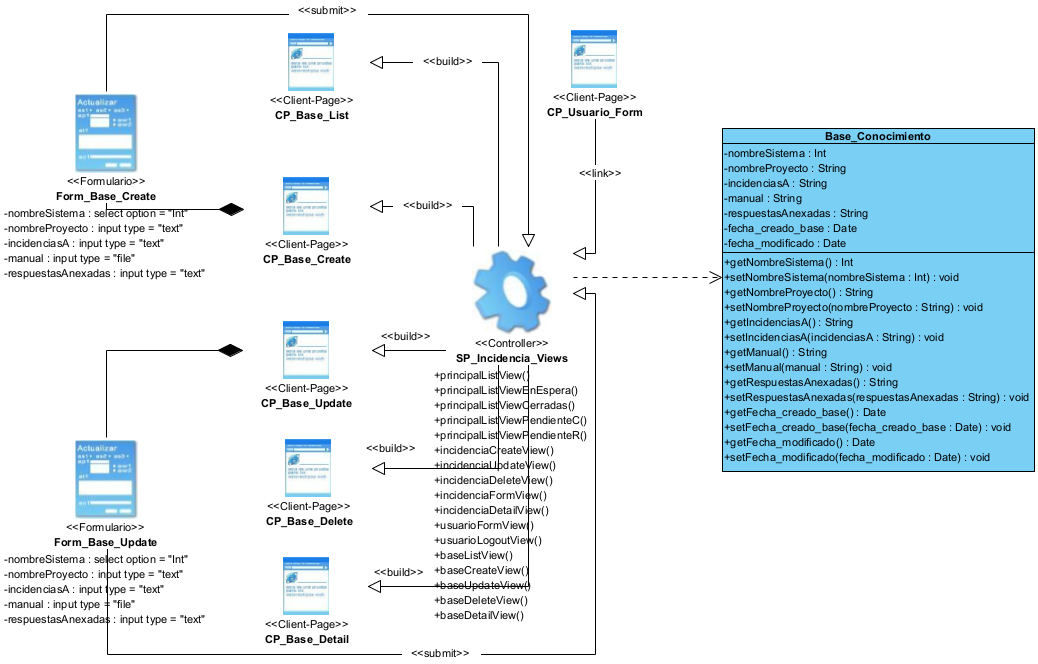
***Figura 3: Diagrama de Clases del Caso de Uso <Gestionar Incidencias>***

**Descripción de Clases del Caso de Uso <Gestionar Incidencias>**

|  |  |  |
| --- | --- | --- |
| **No.** | **Nombre de la Clase** | **Descripción** |
| **1** | **<<Client Page>>**  **CP\_Usuario\_Form** | Esta clase describe la vista que tiene el usuario del login que posee el módulo, por el que se debe autenticar para poder acceder a la gestión de Incidencias y derivados. |
| **2** | **<<Client Page>>**  **CP\_Principal\_List** | Esta clase describe la vista que tiene el usuario del listado de incidencias creadas hasta el momento. |
| **3** | **<<Client Page>>**  **CP\_Incidencia\_Create** | Esta clase describe la vista que tiene el usuario cuando solicita crear una Nueva Incidencia. |
| **4** | **<<Client Page>>**  **CP\_Incidencia\_Update** | Esta clase describe la vista que tiene el usuario cuando solicita Editar una incidencia específica. |
| **5** | **<<Client Page>>**  **CP\_Incidencia\_Detail** | Esta clase describe la vista que tiene el usuario cuando solicita Ver una incidencia específica. |
| **6** | **<<Client Page>>**  **CP\_Incidencia\_Delete** | Esta clase describe la vista que tiene el usuario cuando solicita Eliminar una incidencia específica. |
| **7** | **<<Controller>>**  **SP\_IncidenciaView** | Esta clase describe las funcionalidades que permiten el control de datos de la Incidencias y a la vez contiene los métodos necesarios para llevar a cabo la gestión de las mismas. |
| **8** | **<<Form>>**  **Form\_Incidencia\_Create** | Esta clase describe los campos que debe rellenar el usuario para crear una Nueva Incidencia |
| **9** | **<<Form>>**  **Form\_Incidencia\_Update** | Esta clase describe los campos que debe rellenar el usuario para Editar una incidencia específica. |
| **10** | **Incidencia** | Esta clase contiene todos los datos de las Incidencias. |

***Tabla 21: Descripción de Clases del Caso de Uso <Gestionar Incidencia>***

* **2.4.2.2 Diagrama de Clases del Caso de Uso <Gestionar Base de Conocimiento>**

****

***Figura 4: Diagrama de Clases del Caso de Uso <Gestionar Base de Conocimiento>***

**Descripción de Clases del Caso de Uso <Gestionar Base de Conocimiento>**

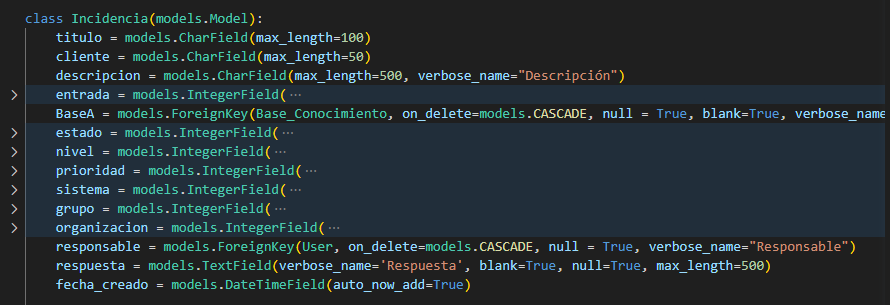
|  |  |  |
| --- | --- | --- |
| **No.** | **Nombre de la Clase** | **Descripción** |
| **1** | **<<Client Page>>**  **CP\_Usuario\_Form** | Esta clase describe la vista que tiene el usuario del login que posee el módulo, por el que se debe autenticar para poder acceder a la gestión de Incidencias y derivados. |
| **2** | **<<Client Page>>**  **CP\_Base\_List** | Esta clase describe la vista que tiene el usuario del listado de Bases de Conocimiento creadas hasta el momento. |
| **3** | **<<Client Page>>**  **CP\_Base\_Create** | Esta clase describe la vista que tiene el usuario cuando solicita crear una Nueva Base de Conocimiento. |
| **4** | **<<Client Page>>**  **CP\_Base\_Update** | Esta clase describe la vista que tiene el usuario cuando solicita Editar una Base de Conocimiento específica. |
| **5** | **<<Client Page>>**  **CP\_Base\_Detail** | Esta clase describe la vista que tiene el usuario cuando solicita Ver una Base de Conocimiento específica. |
| **6** | **<<Client Page>>**  **CP\_Base\_Delete** | Esta clase describe la vista que tiene el usuario cuando solicita Eliminar una Base de Conocimiento específica. |
| **7** | **<<Controller>>**  **SP\_IncidenciaView** | Esta clase describe las funcionalidades que permiten el control de datos de las Bases de Conocimientos y a la vez contiene los métodos necesarios para llevar a cabo la gestión de las mismas. |
| **8** | **<<Form>>**  **Form\_Base\_Create** | Esta clase describe los campos que debe rellenar el usuario para crear una Nueva Base de Conocimiento. |
| **9** | **<<Form>>**  **Form\_Base\_Update** | Esta clase describe los campos que debe rellenar el usuario para Editar una Base de Conocimiento específica. |
| **10** | **Base\_Conocimiento** | Esta clase contiene todos los datos de las Bases de Conocimiento. |

***Tabla 22: Descripción de Clases del Caso de Uso <Gestionar Base de Conocimiento>***

**II.2.4.2.3 Patrones de Diseño**

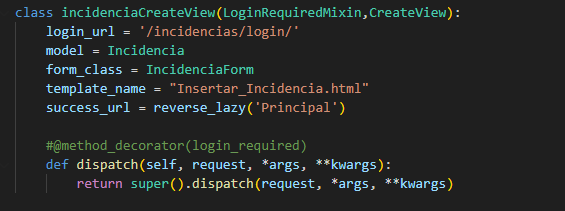
* **Patrones GRASP:** Los Patrones Generales de Software para Asignar Responsabilidades (GRASP, por sus siglas en inglés) son una serie de "buenas prácticas" de aplicación recomendable en el diseño de software.

1. **Patrón Experto:** Es el principio básico de asignación de responsabilidades en diseño Orientado a Objetos. Se encarga de “asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad” (Tabares, 2010). En la propuesta de solución se evidencia en la clase Incidencia, encargada de almacenar todos los atributos que puede tener una Incidencia.

****

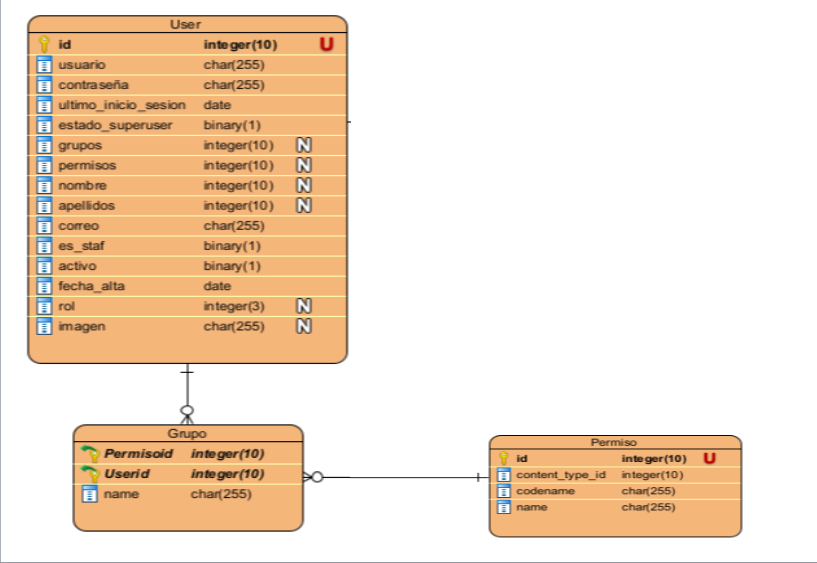
***Figura 5: Ejemplo en el módulo del Patrón Experto***

1. **Patrón Creador:** Guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. Es el responsable de asignarle a la clase B la responsabilidad de crear una instancia de clase A, o sea que B es un creador de los objetos A (Tabares, 2010). En la propuesta de solución se evidencia en la clase incidenciaCreateView, que se encarga de crear una instancia de la clase Incidencia.

****

***Figura 6: Ejemplo en el módulo del Patrón Creador***

1. **Patrón Alta-Cohesión:** “En la perspectiva del diseño orientado a objetos, la cohesión (o, más exactamente, la cohesión funcional) es una medida de cuán relacionadas y enfocadas están, las responsabilidades de una clase”(Tabares, 2010). En la propuesta de solución se evidencia en la colaboración de User y Permiso para crear Grupo.

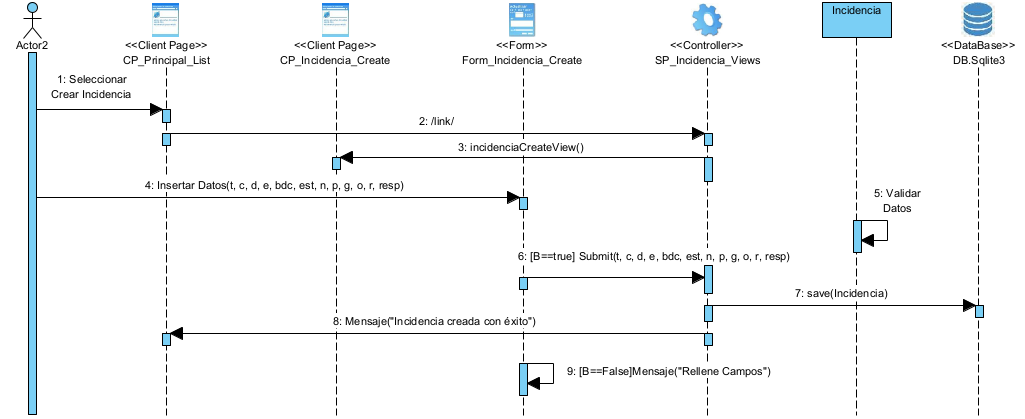
****

***Figura 7: Ejemplo en el módulo del Patrón Alta Cohesión***

1. **Patrón Controlador:** Sugiere que la lógica de negocios debe estar separada de la capa de presentación, para aumentar la reutilización de código y a la vez tener un mayor control. Este patrón se evidencia en la propuesta de solución con cada clase controladora.

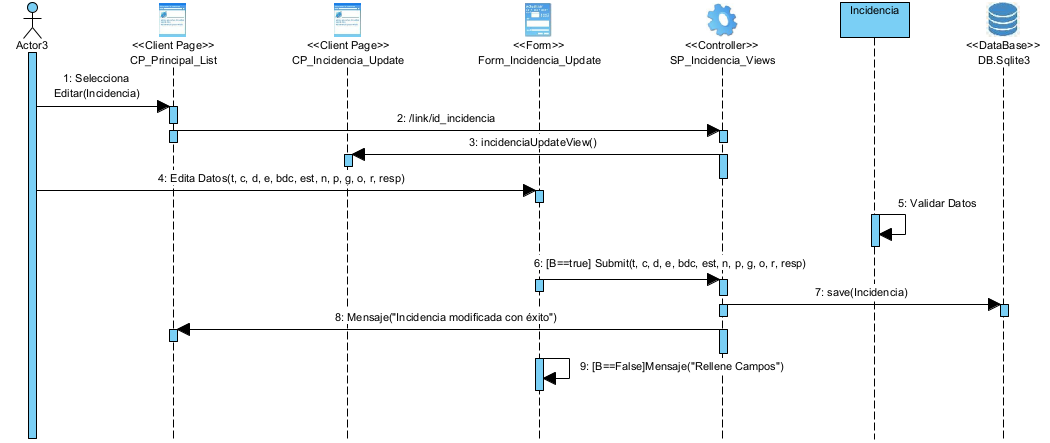
**II.2.4.3 Diagramas de Secuencias:**

* **2.4.3.1 Diagrama de Secuencia del Caso de Uso <Gestionar Incidencia>. <Insertar Incidencia>**

****

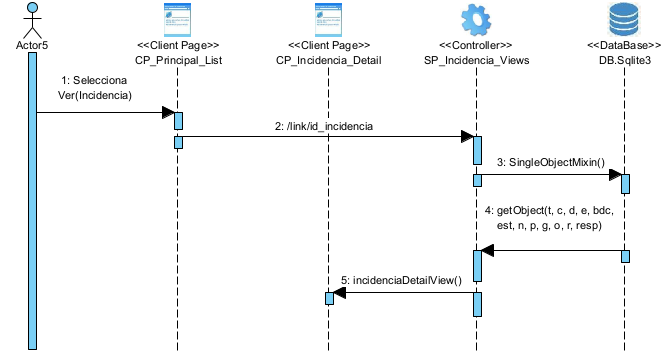
***Figura 8: Diagrama de Secuencia <Insertar Incidencia>***

* **2.4.3.1 Diagrama de Secuencia del Caso de Uso <Gestionar Incidencia>. <Modificar Incidencia>**

****

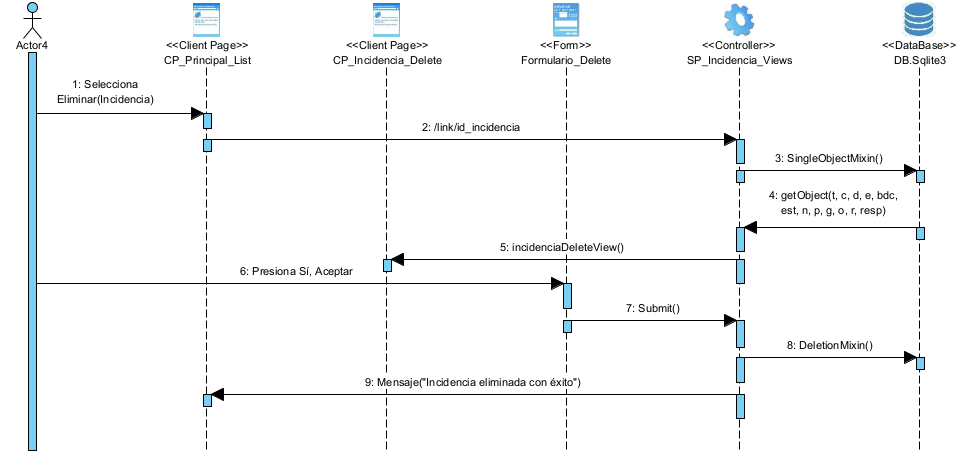
***Figura 9: Diagrama de Secuencia <Modificar Incidencia>***

* **2.4.3.1 Diagrama de Secuencia del Caso de Uso <Gestionar Incidencia>. <Mostrar Incidencia>**

****

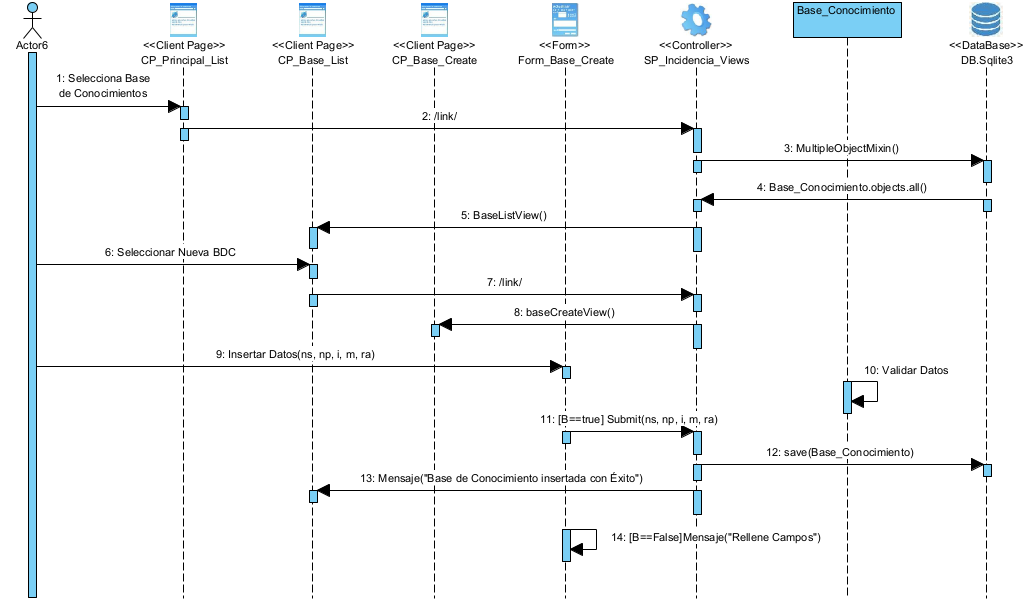
***Figura 10: Diagrama de Secuencia <Mostrar Incidencia>***

* **2.4.3.1 Diagrama de Secuencia del Caso de Uso <Gestionar Incidencia>. <Eliminar Incidencia>**

****

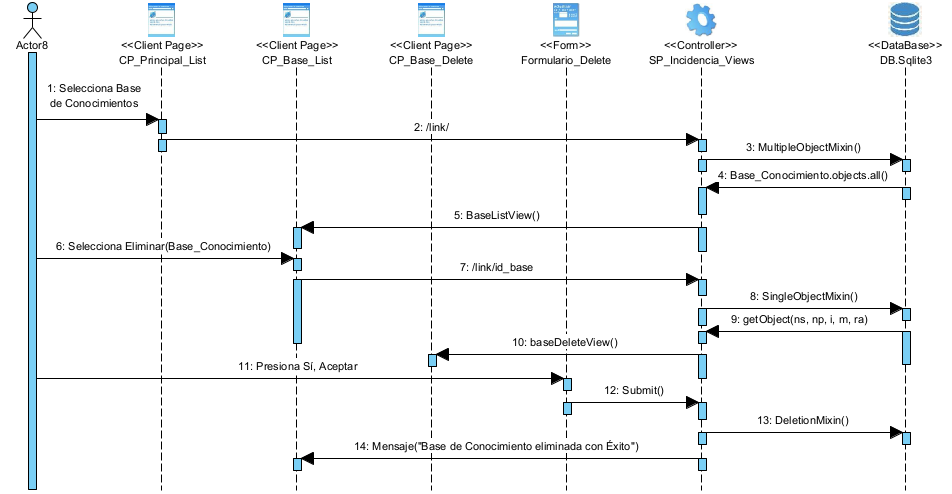
***Figura 11: Diagrama de Secuencia <Eliminar Incidencia>***

* **2.4.3.1 Diagrama de Secuencia del Caso de Uso <Gestionar Base de Conocimiento>. <Insertar Base de Conocimiento>**

****

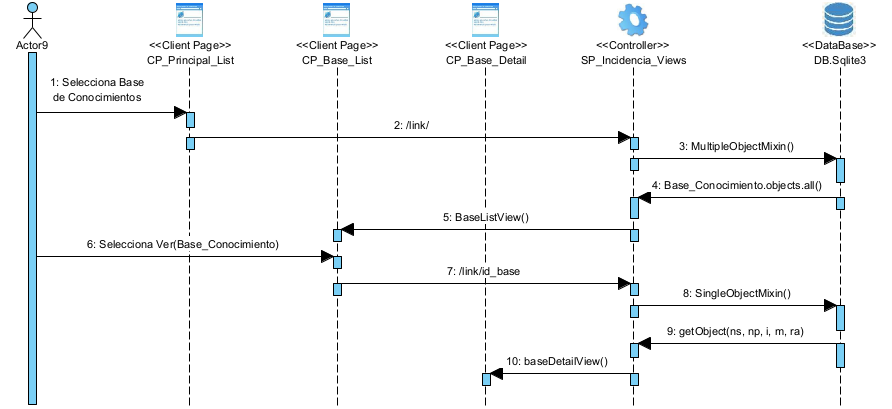
***Figura 12: Diagrama de Secuencia <Insertar Base de Conocimiento>***

* **2.4.3.1 Diagrama de Secuencia del Caso de Uso <Gestionar Base de Conocimiento>. <Modificar Base de Conocimiento>**

****

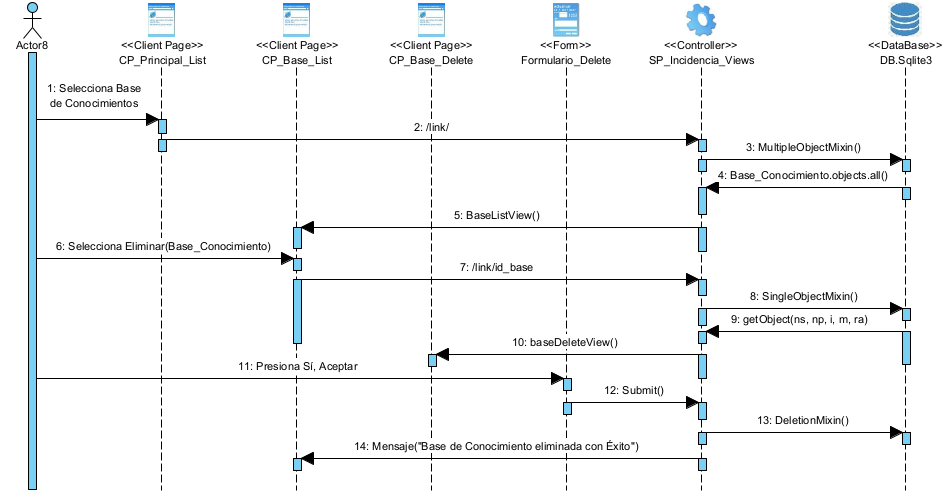
***Figura 13: Diagrama de Secuencia <Modificar Base de Conocimiento>***

* **2.4.3.1 Diagrama de Secuencia del Caso de Uso <Gestionar Base de Conocimiento>. <Mostrar Base de Conocimiento>**

****

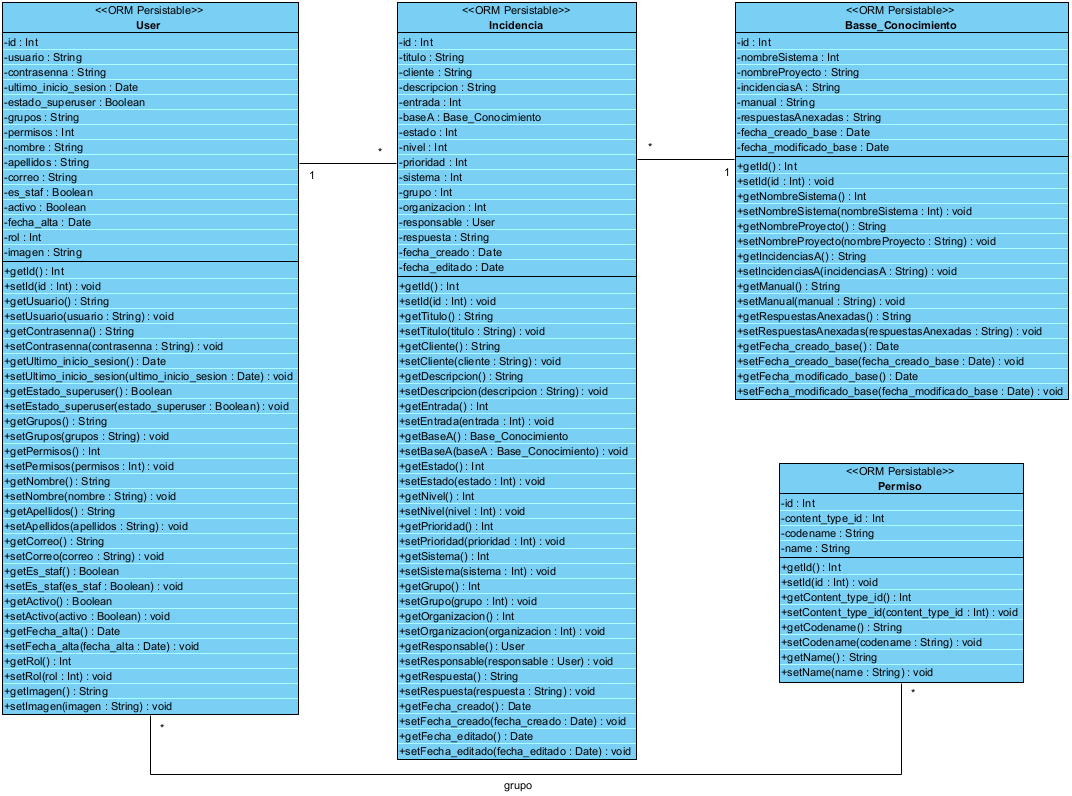
***Figura 14: Diagrama de Secuencia <Mostrar Base de Conocimiento>***

* **2.4.3.1 Diagrama de Secuencia del Caso de Uso <Gestionar Base de Conocimiento>. <Eliminar Base de Conocimiento>**

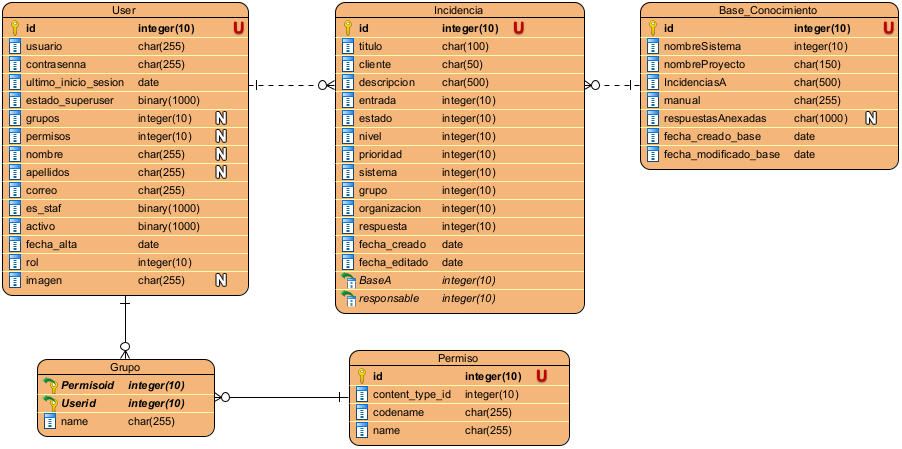
****

***Figura 15: Diagrama de Secuencia <Eliminar Base de Conocimiento>***

**II.2.4.4 Diagrama de Entidad-Relación:**

****

***Figura 16: Diagrama de Clases Persistentes***

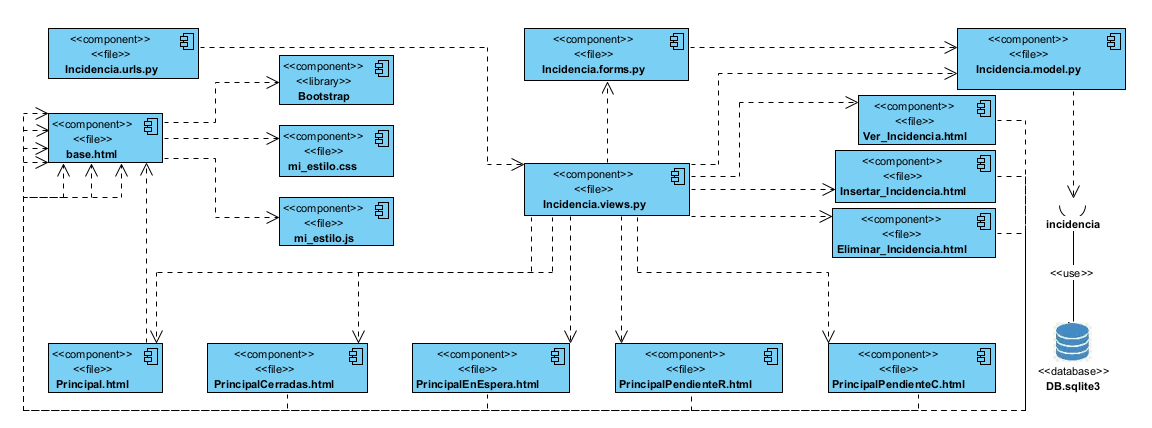
****

***Figura 17: Diagrama Entidad-Relación***

**II.2.5 Implementación.**

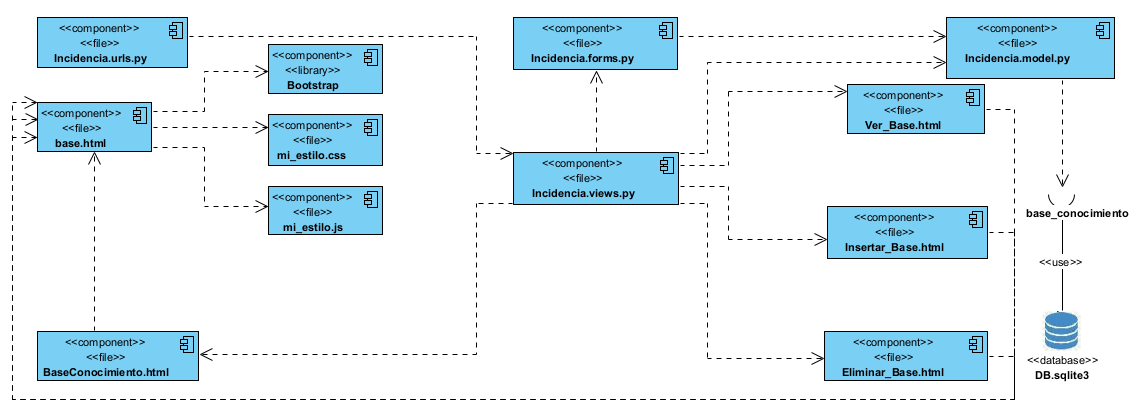
**II.2.5.1 Diagrama de Componentes:**

* **2.5.1.1 Diagrama de Componentes del caso de uso <Gestionar Incidencia>**

****

***Figura 18: Diagrama de Componentes <Gestionar Incidencia>***

* **2.5.1.2 Diagrama de Componentes del caso de uso <Gestionar Base de Conocimiento>**

****

***Figura 19: Diagrama de Componentes <Gestionar Base de Conocimiento>***

## II.3 Diseño e implementación del módulo de gestión de incidencias tecnológicas de soporte y de la Base de Conocimiento correspondiente.

<Por lo general este epígrafe presenta el diseño de los mecanismos para el almacenamiento, procesamiento y transmisión de los datos; así como ejemplos de la implementación de estos mecanismos y de las interfaces gráficas de usuario de la solución propuesta>

## II.4 Tratamiento de errores y despliegue del módulo de incidencias tecnológicas de soporte.



***Figura 20: Diagrama de Despliegue***

* **II.4.1 PC-Cliente:** Computador Cliente en la que los usuarios hacen uso del sistema. Donde se intercambia información a través de peticiones y respuestas entre usuario y sistema. No requiere de grandes capacidades de cómputo. Solamente un procesador Intel® Celeron® CPU N3050 @ 1.60GHz o mayor, memoria RAM de 512mb y un navegador cualquiera instalado, ya sea Opera, Chrome, Microsoft Edge o Mozilla Firefox.
* **II.4.2 Web-Server:** Computador donde estará montado el servidor de la Página web. Debe poseer uno de los sistemas operativos entre Linux y Windows. Debe tener instalado el Software Apache y poseer al menos 2 GB de RAM, un CPU de 64 bits a 2 núcleos, una tarjeta de interfaz de red y más de 120 GB de almacenamiento.
* **II.4.3 Data-Base-Server:** Manejo de grandes y complejos volúmenes de datos, al tiempo en que se requiere compartir la información con un conjunto de clientes, sean aplicaciones o usuarios.
* **II.4.4 <<https:443>>:** A través de este protocolo y puerto se establecerá la comunicación entre los PC clientes y el servidor de forma segura.
* **II.4.5 <<ip:3306>>:** Provee la conectividad de extremo a extremo especificando cómo los datos deberían ser transmitidos, enrutados, formateados, direccionados y recibidos por el destinatario. El puerto 3306 hace que dicho protocolo se especifique para una base de datos MySQL.

<Por lo general este epígrafe presenta el diseño del tratamiento de los errores, así como el diseño del despliegue de la solución propuesta>

## Conclusiones del capítulo

<La lista de conclusiones en este capítulo por lo general van dirigidas a establecer los argumentos [posición científica que adopta el autor] a partir de la aplicación del enfoque, metodología y métodos de Ingeniería de Software aplicados para lograr cumplir el objetivo, resolver el problema científico planteado y obtener un resultado práctico; destacando la pertinencia de las decisiones tomadas a lo largo del proceso de desarrollo de software descrito>

<EJEMPLO DE ESTRUCTURA Y CONTENIDO DE ESTE CAPÍTULO:

Título del Capítulo: SoftCalTest: Una aplicación web para la evaluación de la calidad de software en la empresa SOFT-PLUS+

I.1 Modelado del proceso de evaluación de la calidad de software en la empresa SOFT-PLUS+

I.2 Requisitos, análisis y diseño de la aplicación web SoftCalTest

I.3 Diseño e implementación del almacenamiento, procesamiento y transmisión de los datos en la aplicación web SoftCalTest

I.4 Tratamiento de errores y despliegue de la aplicación web SoftCalTest

Conclusiones del capítulo>

# CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

<Introducción del capítulo 3 con una breve explicación del objetivo que persigue el capítulo, los principales contenidos que aborda, la estructura que puede encontrar el lector en su composición y un breve texto introductorio a las temáticas principales que aborda el capítulo>

## III.1 Verificación y validación del módulo de gestión de incidencias tecnológicas de soporte

**III.1.1 Diseños de Casos de Pruebas:**

* **Caja Negra (Técnica de partición de equivalencia)**

**Caso de Uso <Gestionar Incidencia>**

**Descripción General:**

Se diseñarán los casos de prueba por secciones, cada sección será una funcionalidad del Caso de Uso <Gestionar Incidencia>. Siendo la primera sección la funcionalidad de Insertar Incidencia.

**Condiciones de ejecución:**

El usuario tiene que estar autenticado al módulo y presentar los permisos necesarios para poder ejecutar las funcionalidades que presenta el Caso de Uso.

**Sección 1: Insertar Incidencia**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Escenario** | **Descripción** | **titulo** | **cliente** | **descripción** | **Respuesta** | **Respuesta del Sistema** | **Flujo Central** |
| **Campos Vacíos** | **Al rellenar el formulario se dejarán algunos campos vacíos** | **V** | **I** | **I** | **V** | **Muestra que es necesario rellenar todos los campos vacíos, los señala y no permite que se ejecute la acción** | **1.Ir a la vista de IncidenciaCreateView**  **2.Rellenar el formulario y dejar campos vacíos**  **3.Presionar Guardar** |
| **‘ejemplot’** | **Vacío** | **Vacío** | **‘ejemplor’** |
| **Datos Incorrectos** | **Al rellenar el formulario se pondrán algunos campos con datos incorrectos** | **V** | **I** | **V** | **V** | **Muestra un alerta con todos los errores del formulario y no permite que se ejecute la acción** | **1.Ir a la vista de IncidenciaCreateView**  **2.Rellenar el formulario y poner datos incorrectos**  **3.Presionar Guardar** |
| **‘ejemplot’** | **‘Zapote&&!ªª’** | **‘ejemplod’** | **‘ejemplor’** |
| **Datos Correctos** | **Se rellenará el formulario sin errores en sus campos** | **V** | **V** | **V** | **V** | **Permite que se ejecute la acción, muestra un mensaje de Inserción exitosa y redirige la vista hacia la página principal** | **1.Ir a la vista de IncidenciaCreateView**  **2.Rellenar el formulario sin errores en los datos**  **3.Presionar Guardar** |
| **‘ejemplot’** | **‘ejemploc’** | **‘ejemplod’** | **‘ejemplor’** |

***Tabla 24: Casos de Prueba de la Sección Insertar Noticia***

**Sección 2: Mostrar Incidencia**

|  |  |  |  |
| --- | --- | --- | --- |
| **Escenario** | **Descripción** | **Respuesta del Sistema** | **Flujo Central** |
| **Mostrar Datos de una Incidencia** | **Se deben mostrar todos los datos que posea la incidencia escogida** | **Se muestran todos los datos que presenta la incidencia escogida** | **1.Ir a la vista de IncidenciaDetailView** |

***Tabla 25: Casos de Prueba de la Sección Mostrar Noticia***

**Sección 3: Modificar Incidencia**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Escenario** | **Descripción** | **titulo** | **cliente** | **descripción** | **Respuesta** | **Respuesta del Sistema** | **Flujo Central** |
| **Campos Vacíos** | **Al modificar el formulario se dejarán algunos campos vacíos** | **V** | **I** | **I** | **V** | **Muestra que es necesario rellenar todos los campos vacíos, los señala y no permite que se ejecute la acción** | **1.Ir a la vista de IncidenciaUpdateView**  **2.Rellenar el formulario y dejar campos vacíos**  **3.Presionar Guardar** |
| **‘ejemplot modificado’** | **Vacío** | **Vacío** | **‘ejemplor modificado’** |
| **Datos Incorrectos** | **Al modificar el formulario se pondrán algunos campos con datos incorrectos** | **V** | **I** | **V** | **V** | **Muestra un alerta con todos los errores del formulario y no permite que se ejecute la acción** | **1.Ir a la vista de IncidenciaUpdateView**  **2.Rellenar el formulario y poner datos incorrectos**  **3.Presionar Guardar** |
| **‘ejemplot**  **modificado’** | **‘Zapote&&!ªª’** | **‘ejemplod**  **modificado’** | **‘ejemplor**  **modificado’** |
| **Datos Correctos** | **Se modificará el formulario sin errores en sus campos** | **V** | **V** | **V** | **V** | **Permite que se ejecute la acción, muestra un mensaje de Inserción exitosa y redirige la vista hacia la página principal** | **1.Ir a la vista de IncidenciaUpdateView**  **2.Rellenar el formulario sin errores en los datos**  **3.Presionar Guardar** |
| **‘ejemplot**  **modificado’** | **‘ejemploc modificado’** | **‘ejemplod modificado’** | **‘ejemplor modificado’** |

***Tabla 26: Casos de Prueba de la Sección Modificar Incidencia***

**Sección 4: Eliminar Incidencia**

|  |  |  |  |
| --- | --- | --- | --- |
| **Escenario** | **Descripción** | **Respuesta del Sistema** | **Flujo Central** |
| **Eliminar una incidencia escogida** | **Se debe eliminar la incidencia escogida y sus datos** | **Se elimina la incidencia de la base de datos exitosamente** | **1.Ir a la vista de IncidenciaDeletelView**  **2.Seleccionar Si\_Aceptar** |

***Tabla 27: Casos de Prueba de la Sección Eliminar Incidencia***

**Caso de Uso <Gestionar Base de Conocimiento>**

**Descripción General:**

Se diseñarán los casos de prueba por secciones, cada sección será una funcionalidad del Caso de Uso <Gestionar Base de Conocimiento>. Siendo la primera sección la funcionalidad de Insertar Base de Conocimiento.

**Condiciones de ejecución:**

El usuario tiene que estar autenticado al módulo y presentar los permisos necesarios para poder ejecutar las funcionalidades que presenta el Caso de Uso.

**Sección 1: Insertar Base de Conocimiento**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Escenario** | **Descripción** | **proyecto** | **Incidencias** | **Respuesta** | **Respuesta del Sistema** | **Flujo Central** |
| **Campos Vacíos** | **Al rellenar el formulario se dejarán algunos campos vacíos** | **V** | **I** | **V** | **Muestra que es necesario rellenar todos los campos vacíos, los señala y no permite que se ejecute la acción** | **1.Ir a la vista de BaseCreateView**  **2.Rellenar el formulario y dejar campos vacíos**  **3.Presionar Guardar** |
| **‘ejemplop’** | **Vacío** | **‘ejemplor’** |
| **Datos Incorrectos** | **Al rellenar el formulario se pondrán algunos campos con datos incorrectos** | **V** | **I** | **V** | **Muestra un alerta con todos los errores del formulario y no permite que se ejecute la acción** | **1.Ir a la vista de BaseCreateView**  **2.Rellenar el formulario y poner datos incorrectos**  **3.Presionar Guardar** |
| **‘ejemplop’** | **‘Zapote&&!ªª’** | **‘ejemplor’** |
| **Datos Correctos** | **Se rellenará el formulario sin errores en sus campos** | **V** | **V** | **V** | **Permite que se ejecute la acción, muestra un mensaje de Inserción exitosa y redirige la vista hacia BaseListView** | **1.Ir a la vista de BaseCreateView**  **2.Rellenar el formulario sin errores en los datos**  **3.Presionar Guardar** |
| **‘ejemplop’** | **‘ejemploi’** | **‘ejemplor’** |

***Tabla 28: Casos de Prueba de la sección Insertar Base de Conocimiento***

**Sección 2: Mostrar Base de Conocimiento**

|  |  |  |  |
| --- | --- | --- | --- |
| **Escenario** | **Descripción** | **Respuesta del Sistema** | **Flujo Central** |
| **Mostrar Datos de una Base de Conocimiento** | **Se deben mostrar todos los datos que posea la Base de Conocimiento escogida** | **Se muestran todos los datos que presenta la Base de Conocimiento escogida** | **1.Ir a la vista de BaseDetailView** |

***Tabla 29: Casos de Prueba de la Sección Mostrar Base de Conocimiento***

**Sección 3: Modificar Base de Conocimiento**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Escenario** | **Descripción** | **proyecto** | **incidencias** | **Respuesta** | **Respuesta del Sistema** | **Flujo Central** |
| **Campos Vacíos** | **Al modificar el formulario se dejarán algunos campos vacíos** | **V** | **I** | **V** | **Muestra que es necesario rellenar todos los campos vacíos, los señala y no permite que se ejecute la acción** | **1.Ir a la vista de BaseUpdateView**  **2.Rellenar el formulario y dejar campos vacíos**  **3.Presionar Guardar** |
| **‘ejemplop modificado’** | **Vacío** | **‘ejemplor modificado’** |
| **Datos Incorrectos** | **Al modificar el formulario se pondrán algunos campos con datos incorrectos** | **V** | **I** | **V** | **Muestra un alerta con todos los errores del formulario y no permite que se ejecute la acción** | **1.Ir a la vista de BaseUpdateView**  **2.Rellenar el formulario y poner datos incorrectos**  **3.Presionar Guardar** |
| **‘ejemplop**  **modificado’** | **‘Zapote&&!ªª’** | **‘ejemplor**  **modificado’** |
| **Datos Correctos** | **Se modificará el formulario sin errores en sus campos** | **V** | **V** | **V** | **Permite que se ejecute la acción, muestra un mensaje de Inserción exitosa y redirige la vista hacia BaseListView** | **1.Ir a la vista de BaseUpdateView**  **2.Rellenar el formulario sin errores en los datos**  **3.Presionar Guardar** |
| **‘ejemplop**  **modificado’** | **‘ejemploc modificado’** | **‘ejemplor modificado’** |

***Tabla 30: Casos de Prueba de la Sección Modificar Base de Conocimiento***

**Sección 4: Eliminar Base de Conocimiento**

|  |  |  |  |
| --- | --- | --- | --- |
| **Escenario** | **Descripción** | **Respuesta del Sistema** | **Flujo Central** |
| **Eliminar una Base de Conocimiento escogida** | **Se debe eliminar la Base de Conocimiento escogida y sus datos** | **Se elimina la Base de Conocimiento de la base de datos exitosamente** | **1.Ir a la vista de BaseDeletelView**  **2.Seleccionar Si\_Aceptar** |

***Tabla 31: Casos de Prueba de la Sección Eliminar Base de Conocimiento***

* **Caja Blanca (Técnica de camino básico)**

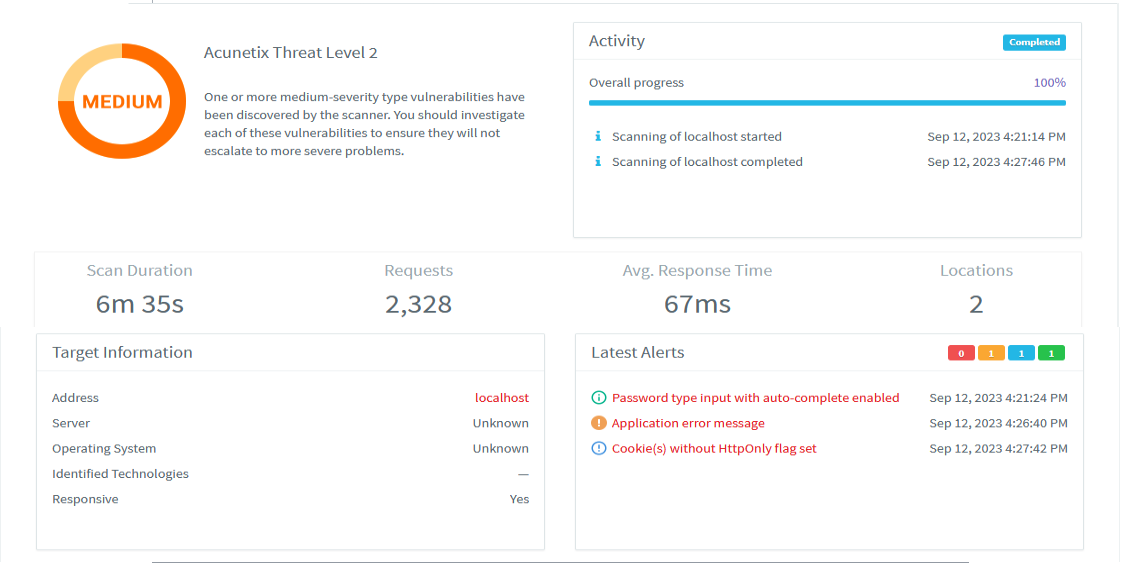
|  |  |
| --- | --- |
| **Caso de Uso:** | **Gestionar Incidencia** |
| **Caso de Prueba:** | Insertar Incidencia |
| **Camino:** |  |
| **Entrada:** | Datos de la incidencia necesarios en el formulario para lograr la inserción. |
| **Resultados:** | Se inserta la incidencia en la base de datos y se agrega al listado de las incidencias. |
| **Condiciones:** | 1-El Usuario debe estar autenticado en la plataforma  2-Los campos del formulario deben cumplir con los requisitos validados. |

***Tabla 32: Casos de Prueba (Caja Blanca) Insertar Incidencia***

**III.1.2 Tipos de Prueba realizadas al módulo de gestión de incidencias tecnológicas**

* **Fiabilidad (Integridad):**

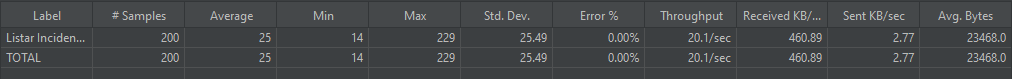
Se realiza la prueba haciendo uso de la herramienta Acunetix Trial Edition, permitiendo un full-scan mediante la asignación de la url del login del módulo de gestión de incidencias realizado. Al finalizar la prueba en un tiempo de 6 minutos con 35 segundos se ven reflejadas tres vulnerabilidades, las cuales se tratarán de manera directa para su resolución e impedir que escalen y se transformen en un problema futuro. Como la mayor vulnerabilidad es de nivel medio, la prueba lanza que la aplicación solo presenta un nivel medio de riesgo en la integridad.

***Figura 21: Prueba de Integridad utilizando la herramienta Acunetix Trial Edition***

* **Rendimiento (Carga):**

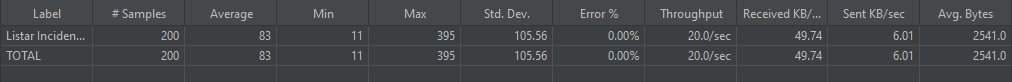
Se realiza la prueba haciendo uso de la herramienta Apache JMeter a partir de 2 iteraciones.

1. La primera con un total de 200 usuarios solicitando el listado de incidencias mediante el método GET durante 10 segundos, lo que significa que realizan esta petición 20 usuarios por segundo. Esta acción se realiza sin haber implementado las validaciones de autenticación de usuarios.



***Figura 22: Prueba de Rendimiento (1) utilizando la herramienta Apache JMeter***

1. La segunda con un total de 200 usuarios solicitando el listado de incidencias mediante el método GET durante 10 segundos, lo que significa que realizan esta petición 20 usuarios por segundo. Esta acción se realiza luego de implementadas las validaciones de autenticación de usuarios.



***Figura 23: Prueba de Rendimiento (2) utilizando la herramienta Apache JMeter***

<En sentido general este epígrafe presenta el diseño de los mecanismos utilizados para la verificación y validación de la solución propuesta, su ejecución y los resultados obtenidos>

## III.2 Impacto del módulo de gestión de incidencias tecnológicas en el Centro de Soporte de la Universidad de las Ciencias Informáticas.

<Por lo general este epígrafe contiene la aplicación de los métodos y técnicas científicos que demuestran la transformación lograda por la solución propuesta en el objeto de estudio, es decir los datos que demuestren el tránsito del estado actual descrito en el capítulo 1 al estado deseado de dicho objeto>

## III.3 Estudio de factibilidad del desarrollo de software del módulo de gestión de incidencias tecnológicas.

<Generalmente este epígrafe contiene el estudio de factibilidad para la realización de la solución propuesta que igualmente demuestra la viabilidad de la solución desarrollada>

## Conclusiones del capítulo

<La lista de conclusiones en este capítulo por lo general van dirigidas a establecer los argumentos y resultados que demuestran la veracidad, factibilidad y fiabilidad de la solución propuesta en términos de los datos obtenidos al aplicar técnicas y métodos de verificación y validación de software, técnicas y métodos de validación científica de la transformación o impacto sobre el objeto de estudio; así como la factibilidad económica de la solución propuesta>

<EJEMPLO DE ESTRUCTURA Y CONTENIDO DE ESTE CAPÍTULO:

Título del Capítulo: Validación de SoftCalTest como aplicación web para la evaluación de la calidad de software en la empresa SOFT-PLUS+

I.1 Verificación y validación de la aplicación web SoftCalTe

I.2 Impacto de la aplicación web SoftCalTest en la evaluación de la calidad de software en la empresa SOFT-PLUS+

I.3 Estudio de factibilidad del desarrollo de software de de la aplicación web SoftCalTest

Conclusiones del capítulo>

# CONCLUSIONES FINALES

<La lista de conclusiones finales por lo general van dirigidas a establecer los argumentos y resultados a los que se arribó en lo siguientes aspectos: (1) sistematización del estado del arte referido al objeto de estudio y el campo de acción, (2) diagnóstico del estado actual del objeto de estudio, (3) principales aspectos del análisis, diseño e implementación de la solución, (4) principales resultados de la validación de la solución propuesta. Deben apoyarse en los resultados obtenidos y descritos en la memoria y no en datos que no aparezcan en este documento. No pueden exceder una cuartilla en su extensión>

# RECOMENDACIONES

<La lista de conclusiones por lo general van dirigidas a establecer aquellos aspectos en los cuales la investigación puede continuar para su perfeccionamiento, mantenimiento o evolución en el tiempo. No deben constituir acciones no realizadas por omisión de etapas del proceso investigativo o ingenieril; ni ser demasiadas en número que cuestionen la completitud y pertinencia de la investigación realizada. No pueden exceder una cuartilla en su extensión>

# REFERENCIAS BIBLIOGRÁFICAS

Alfonso Benítez, D. (2017). *Herramienta para generar productos de trabajos de la metodología variación AUP-UCI* [B.S. thesis]. Universidad de las Ciencias Informáticas. Facultad 3.

Cassá, J. M. O., & Ufano, P. A. O. D. (2012). *Programación web en Java*. Ministerio de Educación.

Dumas, M., La Rosa, M., Mendling, J., Reijers, H. A., Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2018). Introduction to business process management. *Fundamentals of business process management*, 1-33.

Fontela, C. (2012). *UML: Modelado de software para profesionales*. Alpha Editorial.

Gómez, M. (2017). Procesamiento digital de señales. *Estadística, probabilidad y ruido*.

Juanes, V. G., Villar, M. C., González, S. F., Gómez, J. A., Alcántara, M. C., de Marino Gómez-Sandoval, M. A., & Caldentey, C. V. (2000). Análisis del consumo de medicamentos utilizando indicadores de calidad en la prescripción. *Atención primaria*, *25*(9), 618-624.

López Vargas, Y., & Vázquez Chávez, A. (2016). La Gestión de Servicios de soporte técnico en el ciclo de vida del desarrollo de software. *Revista Cubana de Ciencias Informáticas*, *10*, 46-60.

Luna, F., Millahual, C. P., & Iacono, M. (2018). *PROGRAMACION WEB Full Stack 24-Salida laboral: Desarrollo frontend y backend-Curso visual y práctico* (Vol. 24). RedUsers.

Paredes Colmenar, M. del P. (2020). *Aplicaciones web*. Sintesis.

Sanz, M. L. (2016). *MF0491\_3 Programación web en el Entorno Cliente.* Ra-Ma Editorial.

Tabares, R. B. (2010). Patrones Grasp y Anti-Patrones: Un Enfoque Orientado a Objetos desde Lógica de Programación. *Entre Ciencia e Ingeniería*, *4*(8), Article 8.

TEAM, A. (s. f.). *Metodología ITIL: Gestión de incidencias y objetivos*. Recuperado 17 de mayo de 2023, de https://www.ambit-bst.com/blog/metodología-itil-gestión-de-incidencias-y-objetivos

Vidal-Silva, C. L., Pham, T. T., Sepúlveda, S. M., & Carter, L. E. (2019). En Búsqueda de un Procedimiento de Desarrollo de Software Modular. Simbiosis entre Programación Orientada a la Característica y Programación Orientada a Aspectos JPI. *Información tecnológica*, *30*(3), 95-104.

Vilchez Velasquez, D. M. (2022). *Metodología ITIL v3 en la gestión de incidencias de la oficina de soporte técnico de una universidad privada, Lima 2022*.

ANEXOS

<Contenido de los anexos con igual tipo de fuente Arial, pero a tamaño 11 puntos e interlineado 1.0 puntos. Debe tratar de sólo utilizarse aquellos anexos imprescindibles para complementar lo presentado en la memoria escrita y que no excedan las ocho (8) o diez (10 páginas). Deben aparecer uno a continuación del otro sin necesidad de saltos de página entre estos>

1. CSS3 (Cascading Style Sheet) [↑](#footnote-ref-1)
2. HTML5 (Hyper Text Markup Language) [↑](#footnote-ref-2)
3. HTTPS (Hyper Text Transfer Protocol Secure) [↑](#footnote-ref-3)