

**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**

El trabajo de diploma se centra en el objetivo de 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 de la Universidad de las Ciencias Informáticas, por lo que se realiza un estudio exhaustivo del proceso de gestión de incidencias y se hace uso de la Metodología AUP-UCI en su escenario 4, utilizando sus herramientas para modelar el desarrollo de la propuesta de solución. Se tomaron en cuenta características procedentes de un estudio de homólogos que manifiesten un trabajo con incidencias tecnológicas como la forma en que llevan a cabo la realización de reportes y la presencia de una base de conocimientos. Al finalizar el desarrollo de la solución y de aplicar las pruebas correspondientes para su validación, se obtuvieron resultados satisfactorios, logrando resolver e implementar todas las exigencias y requisitos del cliente.

PALABRAS CLAVE

Incidencia, Base de Conocimiento, Reporte, Metodología, Módulo

***ABSTRACT***

*The diploma work focuses on the objective of developing a module for managing technological support incidences based on new requirements and containing new functionalities for strengthening the services of the Support Center of the University of Computer Sciences, so carries out an exhaustive study of the incidences management process and uses the AUP-UCI Methodology in its scenario 4, using its tools to model the development of the solution proposal. Characteristics from a peer study that demonstrate work with technological incidences were taken into account, such as the way they carry out reporting and the presence of a knowledge base. At the end of the development of the solution and applying the corresponding tests for validation, satisfactory results were obtained, managing to resolve and implement all the client's demands and requirements.*

*KEYWORDS*

*Incidence, Knowledge Base, Report, Methodology, Module*

**TABLA DE CONTENIDOS**

[INTRODUCCIÓN 11](#_Toc149385654)

[CAPÍTULO I: Fundamentos y referentes teórico-metodológicos sobre el Módulo de gestión de incidencias tecnológicas 17](#_Toc149385655)

[I.1 Conceptos fundamentales 17](#_Toc149385656)

[I.2 Sistemas de Gestión de Incidencias Tecnológicas nacionales e internacionales: 19](#_Toc149385657)

[I.3 Gestión de Incidencias Tecnológicas: 24](#_Toc149385658)

[I.4 Metodología, herramientas y tecnología 27](#_Toc149385659)

[Conclusiones del capítulo 31](#_Toc149385660)

[CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO 32](#_Toc149385661)

[II.1 Modelado del proceso de gestión de incidencias tecnológicas de soporte 32](#_Toc149385662)

[II.2 Requisitos, análisis y diseño del módulo de gestión de incidencias tecnológicas de soporte 32](#_Toc149385663)

[II.3 Diseño e implementación del módulo de gestión de incidencias tecnológicas de soporte y de la Base de Conocimiento correspondiente. 55](#_Toc149385664)

[II.4 Tratamiento de errores y despliegue del módulo de incidencias tecnológicas de soporte. 55](#_Toc149385665)

[Conclusiones del capítulo 56](#_Toc149385666)

[CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA 57](#_Toc149385667)

[III.1 Verificación y validación del módulo de gestión de incidencias tecnológicas de soporte 57](#_Toc149385668)

[III.2 Impacto del módulo de gestión de incidencias tecnológicas en el Centro de Soporte de la Universidad de las Ciencias Informáticas. 66](#_Toc149385669)

[Conclusiones del capítulo 68](#_Toc149385670)

[CONCLUSIONES FINALES 70](#_Toc149385671)

[RECOMENDACIONES 71](#_Toc149385672)

[REFERENCIAS BIBLIOGRÁFICAS 72](#_Toc149385673)

**ÍNDICE DE TABLAS**

[Tabla 1: Resultados obtenidos de la comparación de sistemas homólogos 24](#_Toc149387943)

[Tabla 2: Requisitos Funcionales 33](#_Toc149387944)

[Tabla 3: Historia de Usuario 3 35](#_Toc149387945)

[Tabla 4: Historia de Usuario 4 36](#_Toc149387946)

[Tabla 5: Historia de Usuario 5 37](#_Toc149387947)

[Tabla 6: Historia de Usuario 6 37](#_Toc149387948)

[Tabla 7: Historia de Usuario 9 39](#_Toc149387949)

[Tabla 8: Historia de Usuario 10 39](#_Toc149387950)

[Tabla 9: Historia de Usuario 11 40](#_Toc149387951)

[Tabla 10: Historia de Usuario 12 41](#_Toc149387952)

[Tabla 11: Descripción de Clases del Caso de Uso <Gestionar Incidencia> 44](#_Toc149387953)

[Tabla 12: Descripción de Clases del Caso de Uso <Gestionar Base de Conocimiento> 46](#_Toc149387954)

[Tabla 13: Casos de Prueba de la Sección Insertar Noticia 58](#_Toc149387955)

[Tabla 14: Casos de Prueba de la Sección Mostrar Noticia 58](#_Toc149387956)

[Tabla 15: Casos de Prueba de la Sección Modificar Incidencia 60](#_Toc149387957)

[Tabla 16: Casos de Prueba de la Sección Eliminar Incidencia 60](#_Toc149387958)

[Tabla 17: Casos de Prueba de la sección Insertar Base de Conocimiento 61](#_Toc149387959)

[Tabla 18: Casos de Prueba de la Sección Mostrar Base de Conocimiento 62](#_Toc149387960)

[Tabla 19: Casos de Prueba de la Sección Modificar Base de Conocimiento 63](#_Toc149387961)

[Tabla 20: Casos de Prueba de la Sección Eliminar Base de Conocimiento 63](#_Toc149387962)

[Tabla 21: Casos de Prueba (Caja Blanca) incidencias “En Espera” 64](#_Toc149387963)

[Tabla 22: Cuadro Lógico de Iadov 67](#_Toc149387964)

[Tabla 23: Niveles de Satisfacción 68](#_Toc149387965)

**ÍNDICE DE FIGURAS**

[Figura 1: Diagrama de Paquetes 42](#_Toc149389616)

[Figura 2: Representación de la arquitectura (MTV), presentando la misma filosofía que (MVC) 42](#_Toc149389617)

[Figura 3: Diagrama de Clases del Caso de Uso <Gestionar Incidencias> 43](#_Toc149389618)

[Figura 4: Diagrama de Clases del Caso de Uso <Gestionar Base de Conocimiento> 45](#_Toc149389619)

[Figura 5: Ejemplo en el módulo del Patrón Experto 47](#_Toc149389620)

[Figura 6: Ejemplo en el módulo del Patrón Creador 47](#_Toc149389621)

[Figura 7: Ejemplo en el módulo del Patrón Alta Cohesión 48](#_Toc149389622)

[Figura 8: Diagrama de Secuencia <Insertar Incidencia> 49](#_Toc149389623)

[Figura 9: Diagrama de Secuencia <Modificar Incidencia> 49](#_Toc149389624)

[Figura 10: Diagrama de Secuencia <Mostrar Incidencia> 50](#_Toc149389625)

[Figura 11: Diagrama de Secuencia <Eliminar Incidencia> 50](#_Toc149389626)

[Figura 12: Diagrama de Secuencia <Insertar Base de Conocimiento> 51](#_Toc149389627)

[Figura 13: Diagrama de Secuencia <Modificar Base de Conocimiento> 51](#_Toc149389628)

[Figura 14: Diagrama de Secuencia <Mostrar Base de Conocimiento> 52](#_Toc149389629)

[Figura 15: Diagrama de Secuencia <Eliminar Base de Conocimiento> 52](#_Toc149389630)

[Figura 16: Diagrama de Clases Persistentes 53](#_Toc149389631)

[Figura 17: Diagrama Entidad-Relación 53](#_Toc149389632)

[Figura 18: Diagrama de Componentes <Gestionar Incidencia> 54](#_Toc149389633)

[Figura 19: Diagrama de Componentes <Gestionar Base de Conocimiento> 54](#_Toc149389634)

[Figura 20: Diagrama de Despliegue 55](#_Toc149389635)

[Figura 21: Método escogido para la técnica de Camino Básico 64](#_Toc149389636)

[Figura 22: Prueba de Fiabilidad utilizando la herramienta Acunetix Trial Edition 65](#_Toc149389637)

[Figura 23: Prueba de Rendimiento (1) utilizando la herramienta Apache JMeter 66](#_Toc149389638)

[Figura 24: Prueba de Rendimiento (2) utilizando la herramienta Apache JMeter 66](#_Toc149389639)

[Figura 25: Prueba de Funcionalidad utilizando la herramienta Selenium IDE 67](#_Toc149389640)

**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.

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 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 de las Ciencias Informáticas.

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.

**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** |

Tabla 1: Resultados obtenidos de la comparación de sistemas homólogos

## 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 responsivas 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:**

DB Browser for SQLite es una aplicación gratuita y de código abierto diseñada para facilitar la creación y administración de las bases de datos con SQLite. Funciona con una interfaz muy clara y sencilla de utilizar, basada en tablas como las que podemos encontrar en Excel de manera que tanto usuarios sin mucha experiencia en la creación y administración de bases de datos, como los desarrolladores más avanzados, puedan trabajar cómodamente con sus bases de datos.

**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:**

SQLite es una herramienta de software libre, que permite almacenar información en dispositivos empotrados de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware, como puede ser una PDA o un teléfono celular. SQLite es uno de los motores de bases de datos SQL gratuito y de código abierto más conocidos y utilizados en toda la red. Este motor se caracteriza principalmente por incluir todo lo necesario para funcionar con unos clics, además de no necesitar un servidor ni complicadas configuraciones para funcionar al 100% y de forma segura.

**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.

**Herramientas de pruebas y validación de la propuesta de solución:**

**I.4.12 Acunetix Trial Edition v11:** Acunetix es el escáner de vulnerabilidades web más maduro del mercado, desarrollado por un equipo especializado desde 2005. El motor está construido con C ++, lo que significa que es mucho más rápido que otras soluciones (*¿Que es Acunetix?*, 2021). Acunetix es una solución de seguridad de aplicaciones web todo en uno, totalmente automatizada que le permite realizar escaneos de aplicaciones web de caja negra, caja gris (IAST), del lado del cliente y fuera de banda (*¿Que es Acunetix?*, 2021).

**I.4.13 Apache Jmeter v5.5:** La aplicación Apache JMeter™ es un software de código abierto, una aplicación Java 100% pura diseñada para cargar pruebas de comportamiento funcional y medir el rendimiento. Fue Diseñado originalmente para probar aplicaciones web, pero tiene desde entonces se expandió a otras funciones de prueba (*Apache JMeter - Apache JMeterTM*, s. f.). Apache JMeter se puede utilizar para probar el rendimiento tanto en modo estático como dinámico. recursos, aplicaciones web dinámicas.

Se puede utilizar para simular una carga pesada en un servidor, grupo de servidores, Red u objeto para probar su resistencia o analizar el rendimiento general. bajo diferentes tipos de carga.

**I.4.14 Selenium IDE:** Selenium es una herramienta de pruebas para aplicaciones web que emplea scripts en cualquier lenguaje de programación para realizar pruebas en el navegador. En pocas palabras, Selenium automatiza el uso del navegador y, con este poder, las posibilidades de uso están abiertas (Hiberus, 2023). El principal objetivo de esta herramienta es la automatización de pruebas sobre aplicaciones web. Su uso se limita a esta actividad, ya que aquellas tareas repetitivas a través del navegador pueden y deberían automatizarse.

## 

## 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

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. .Se realiza un amplio análisis y diseño de los diagramas necesarios para visualizar una correcta implementación de la propuesta de solución donde quedan explícitos los patrones de diseño generales utilizados, se muestran las interfaces gráficas del módulo y el proceso de despliegue del mismo.

## 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 |

Tabla 2: Requisitos Funcionales

**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 utilizará 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:** 0.14 semanas |
| **Descripción:** Se podrá visualizar las diferentes características o parámetros asociados a la Base de Conocimiento.  El procedimiento para poder visualizar una base de conocimiento es el siguiente:   1. Seleccionar en la barra de navegación el link (Base de Conocimiento). 2. Escoger la base de conocimiento a visualizar en el listado de bases de conocimientos que se muestra. 3. Seleccionar el link anexado a la base de conocimiento escogida que tiene por título (Ver).   Luego se podrán visualizar todos los datos que contiene la base de conocimiento escogida. | | |
| **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: 0.14** |
| **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:** 0.14 semanas |
| **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.  El procedimiento para poder eliminar una base de conocimiento es el siguiente:   1. Seleccionar en la barra de navegación el link (Base de Conocimiento). 2. Escoger la base de conocimiento a eliminar en el listado de bases de conocimientos que se muestra. 3. Seleccionar el link anexado a la base de conocimiento escogida que tiene por título (Eliminar).   Luego se podrá eliminar la base de conocimiento escogida. | | |
| **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:** 0.14 semanas |
| **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 7: 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:** 0.14 semanas |
| **Descripción:** Se podrá visualizar todas las características o parámetros que conforman a la incidencia.  El procedimiento para poder visualizar una incidencia es el siguiente:   1. Escoger la incidencia a visualizar en el listado de incidencias que se muestra en la página principal. 2. Seleccionar el link anexado a la incidencia escogida que tiene por título (Ver).   Luego se podrán visualizar todos los datos que contiene la incidencia escogida. | | |
| **Observaciones:** | | |
| **Prototipo de Interfaz gráfica:** | | |

Tabla 8: 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:** 0.14 semanas |
| **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 9: 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:** 0.14 semanas |
| **Descripción:** Se podrán eliminar las incidencias de la Base de Datos junto con sus características asociadas.  El procedimiento para poder eliminar una incidencia es el siguiente:   1. Escoger la incidencia a eliminar en el listado de incidencias que se muestra en la página principal. 2. Seleccionar el link anexado a la incidencia escogida que tiene por título (Eliminar).   Luego se podrá eliminar la incidencia escogida. | | |
| **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 10: Historia de Usuario 12

**II.2.4 Análisis y Diseño.**

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

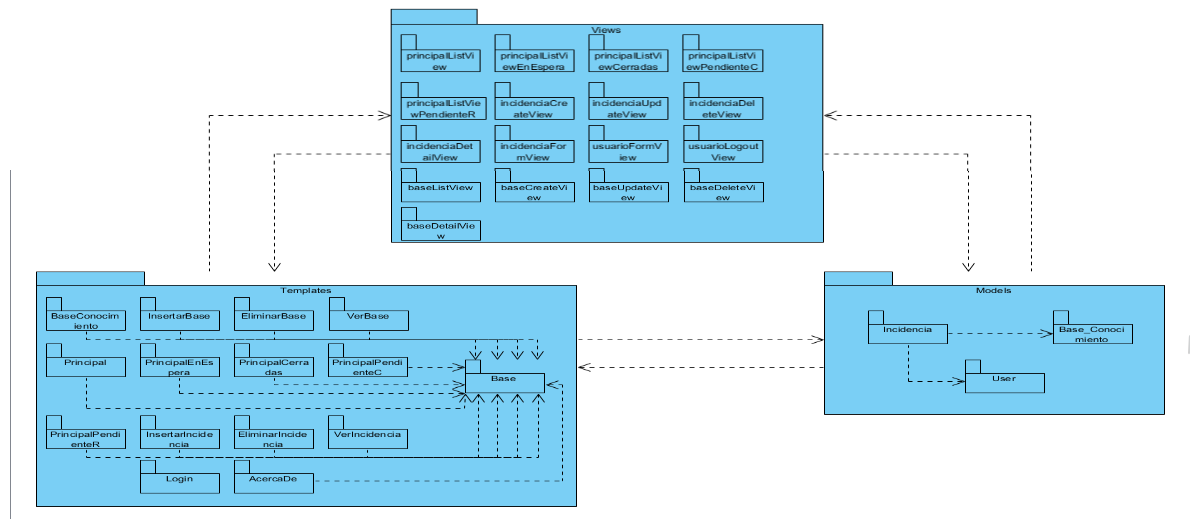
****

Figura 1: Diagrama de Paquetes

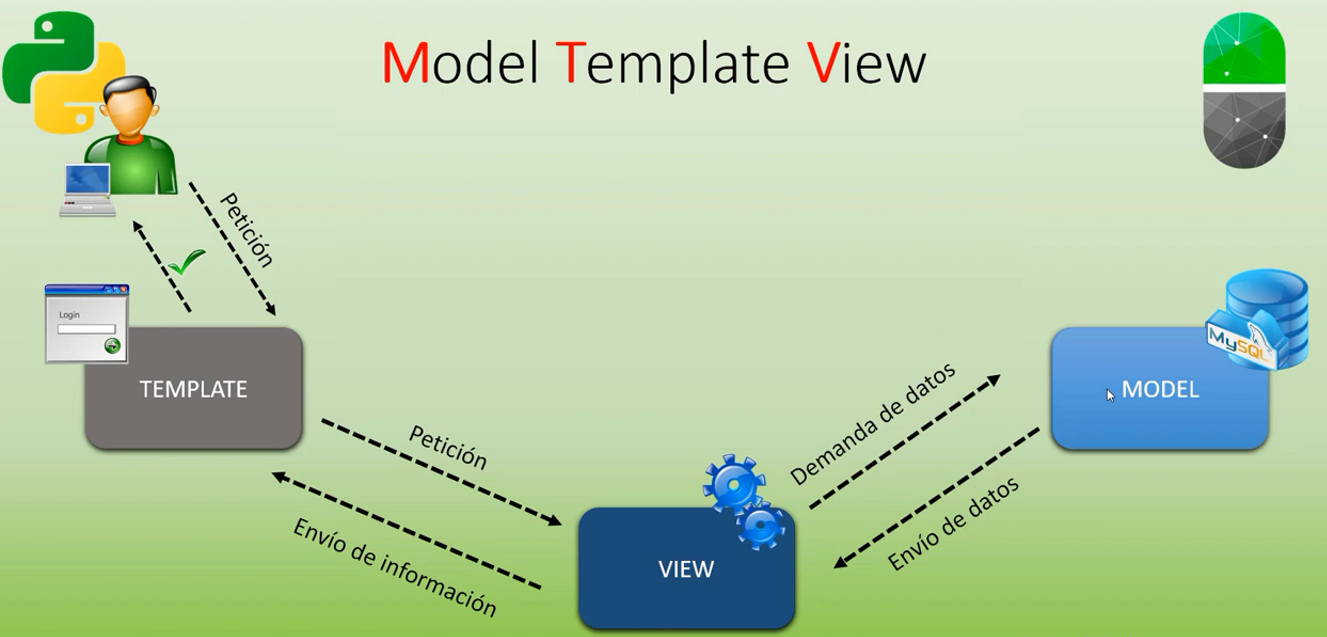
****

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>**

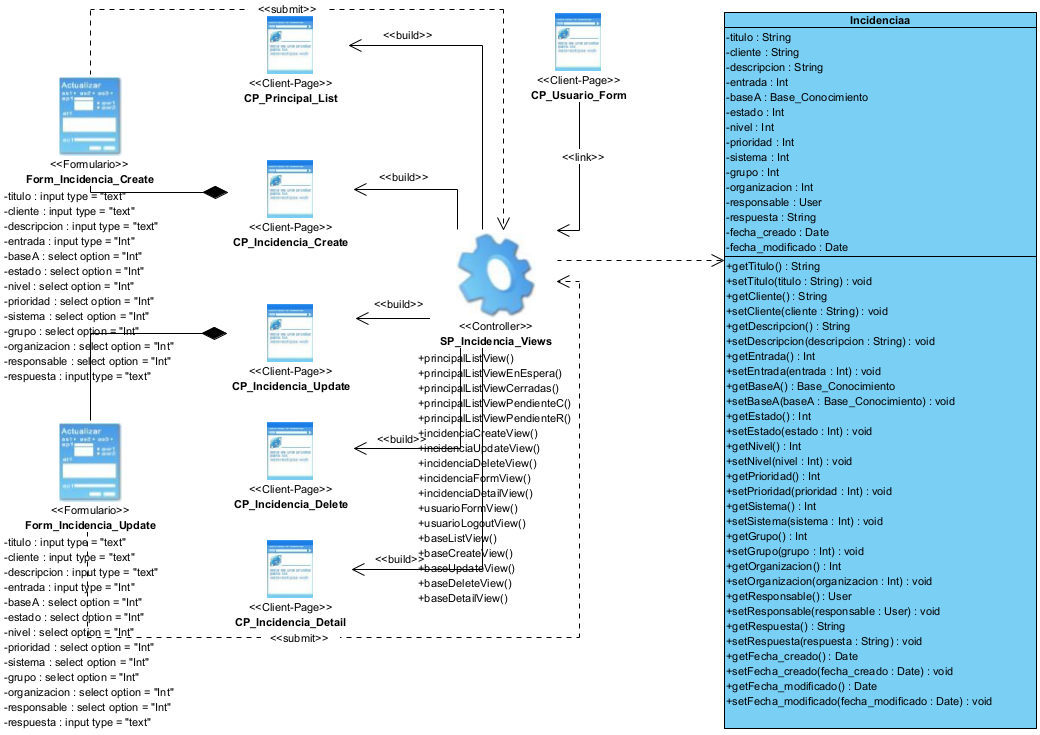
****

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 11: Descripción de Clases del Caso de Uso <Gestionar Incidencia>

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

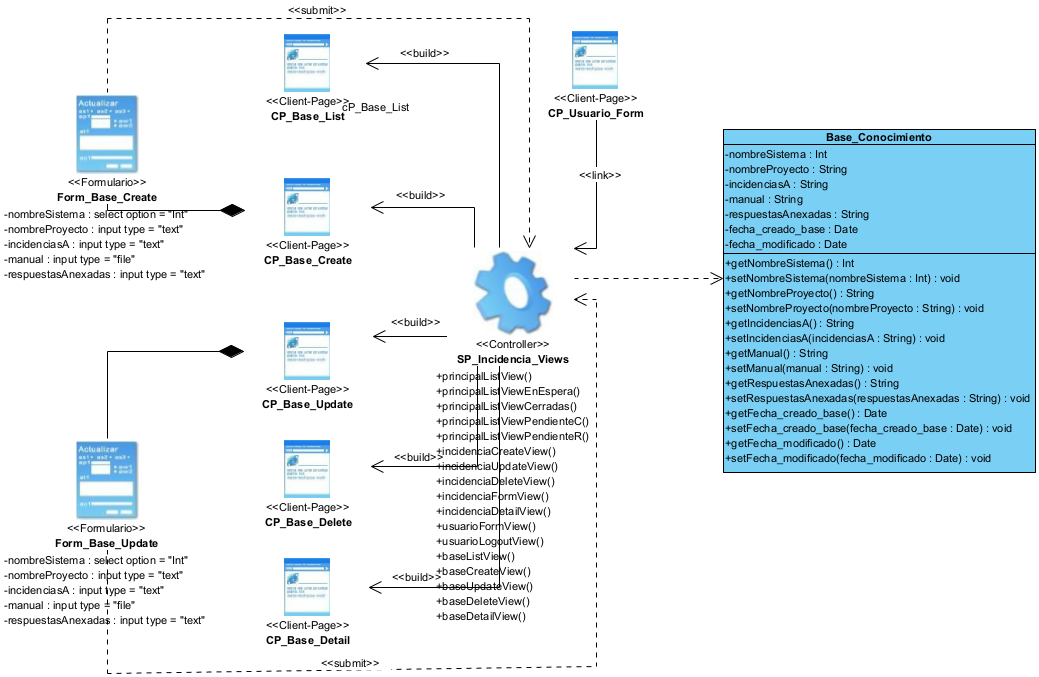
****

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 12: 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.

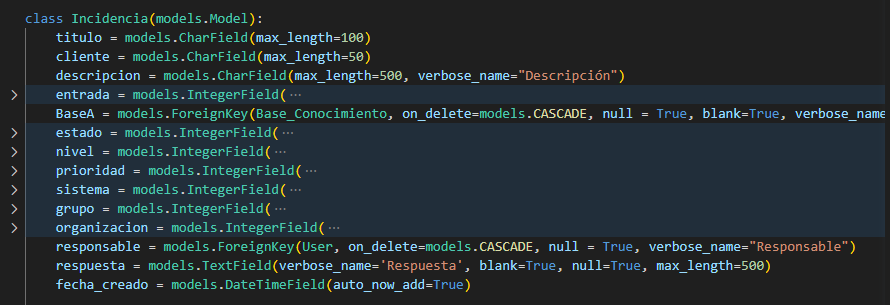
****

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.

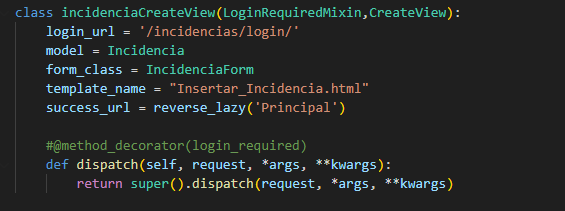
****

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.

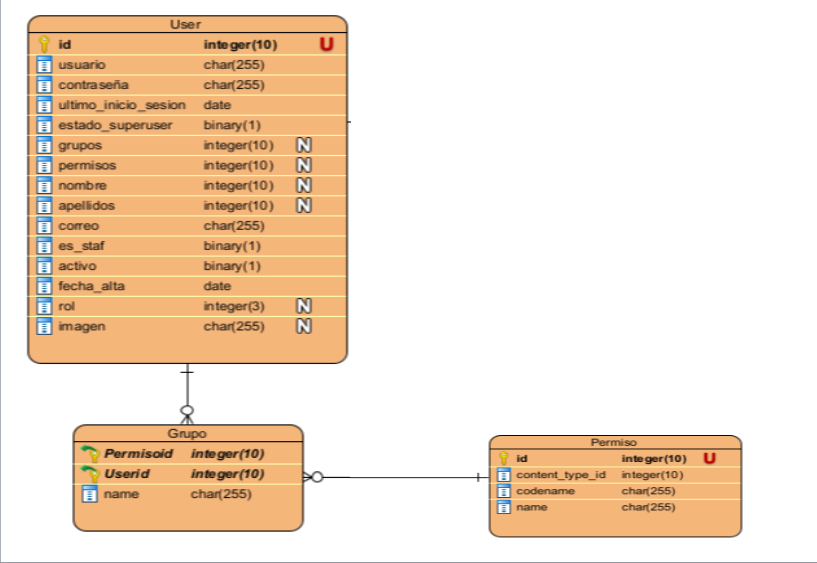
****

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>**

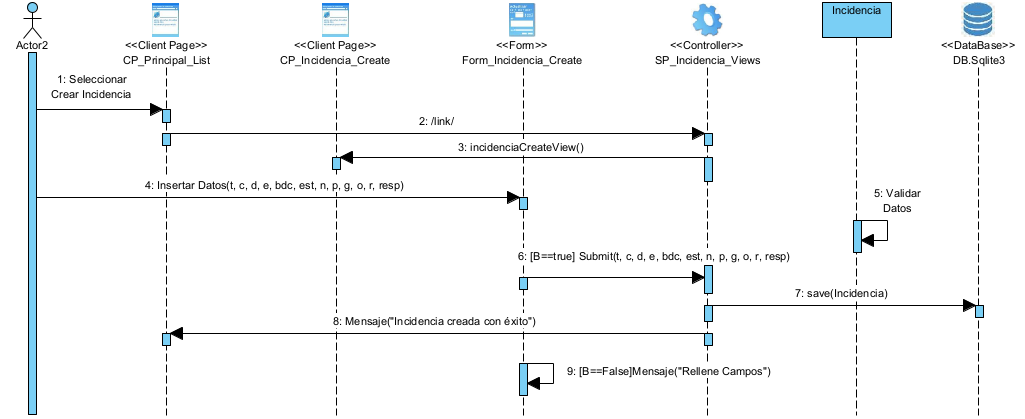
****

Figura 8: Diagrama de Secuencia <Insertar Incidencia>

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

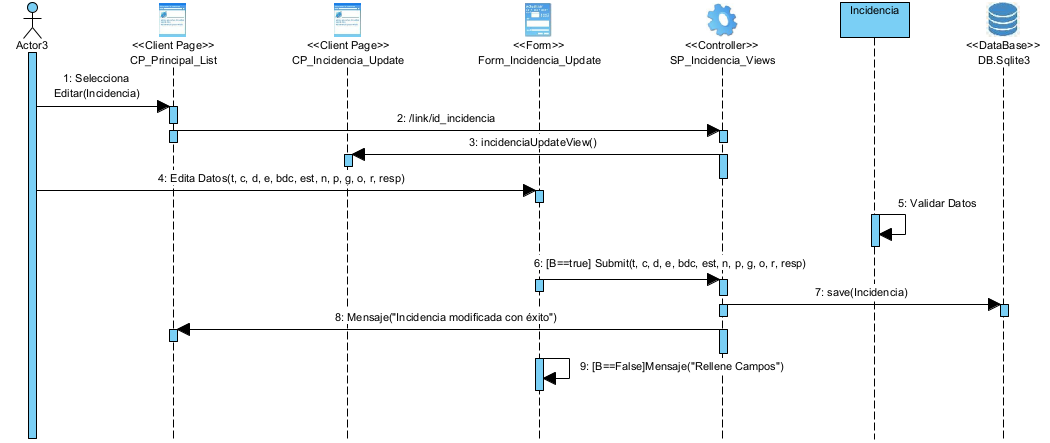
****

Figura 9: Diagrama de Secuencia <Modificar Incidencia>

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

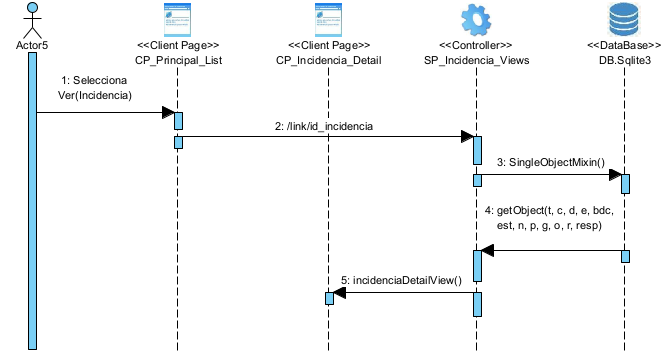
****

Figura : Diagrama de Secuencia <Mostrar Incidencia>

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

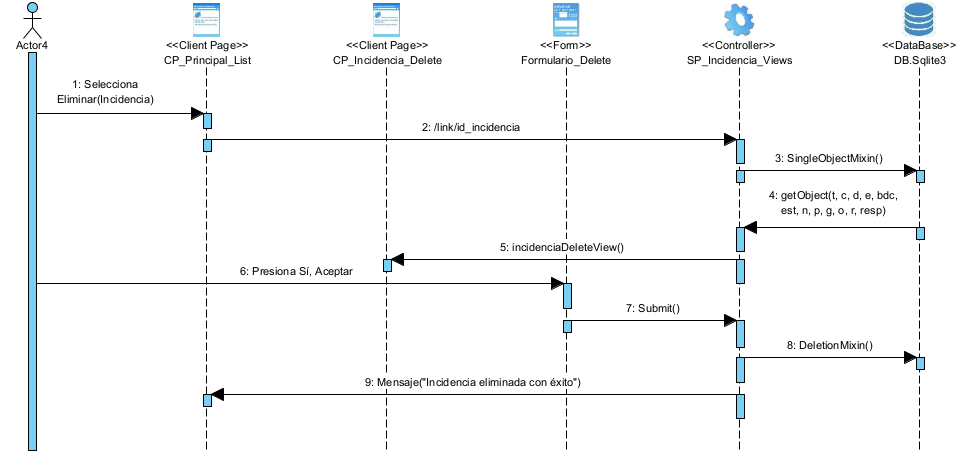
****

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>**

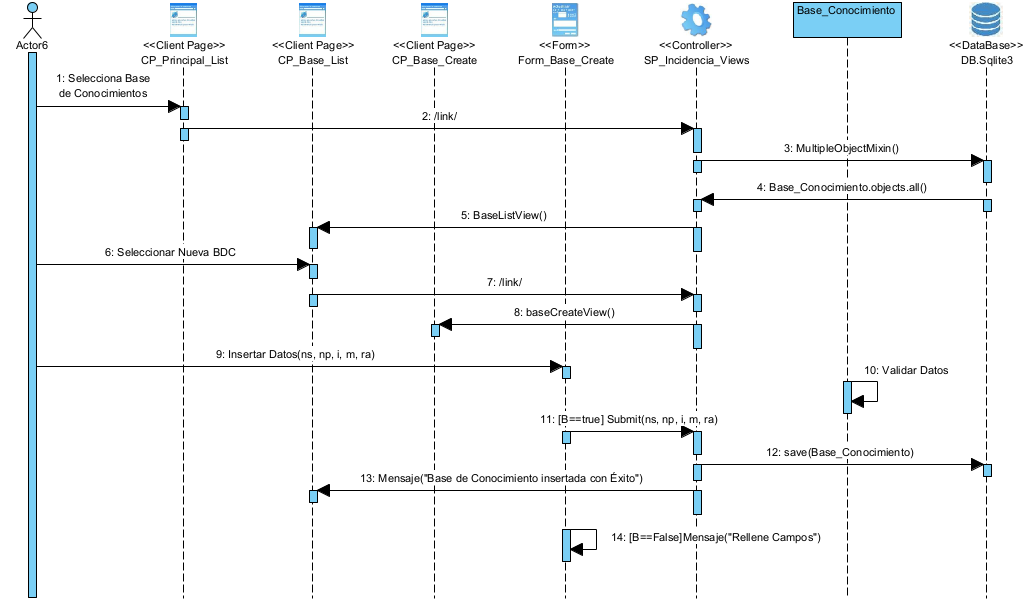
****

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>**

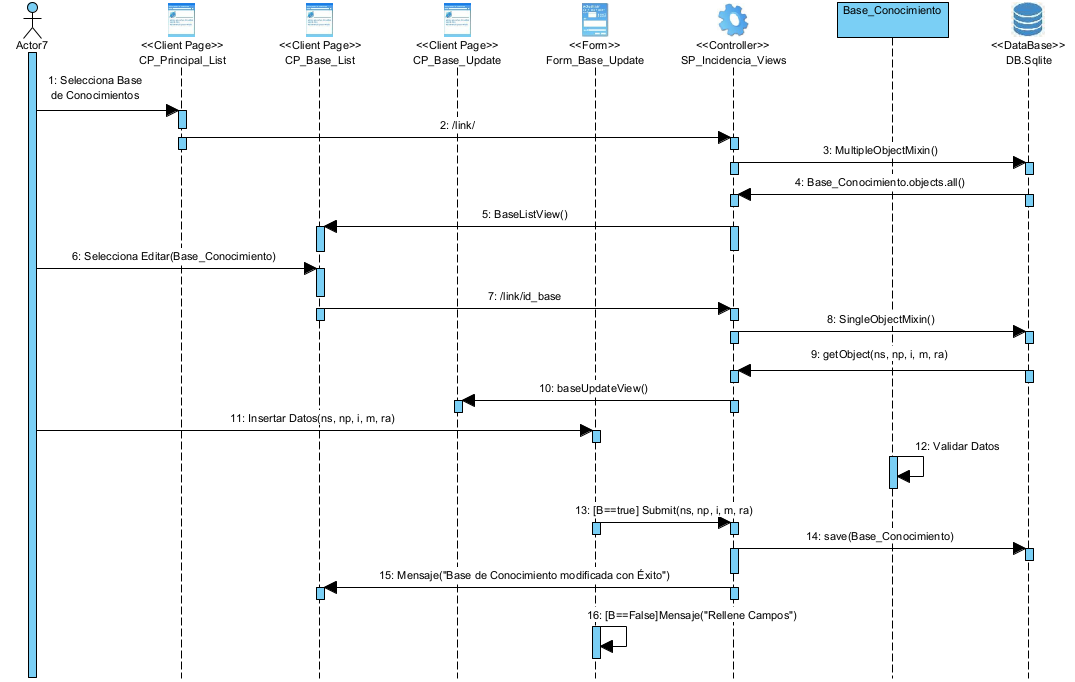
****

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>**

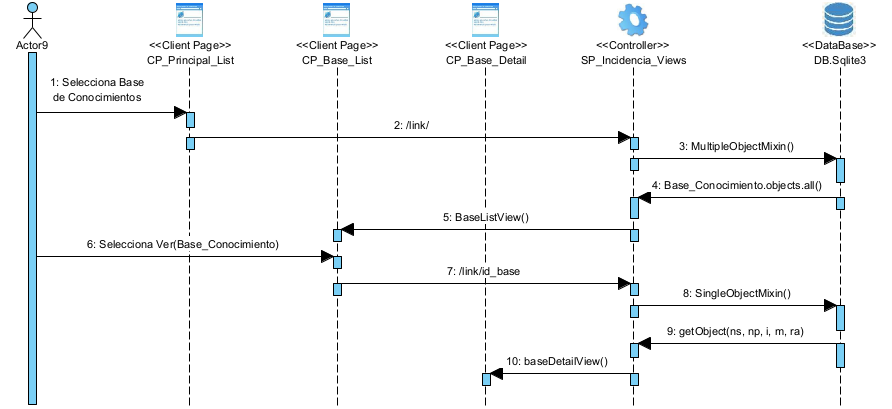
****

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>**

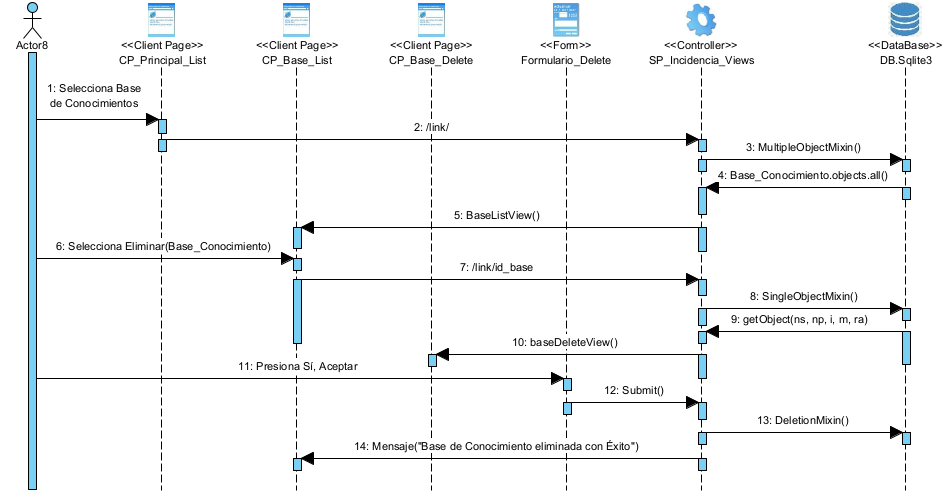
****

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

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

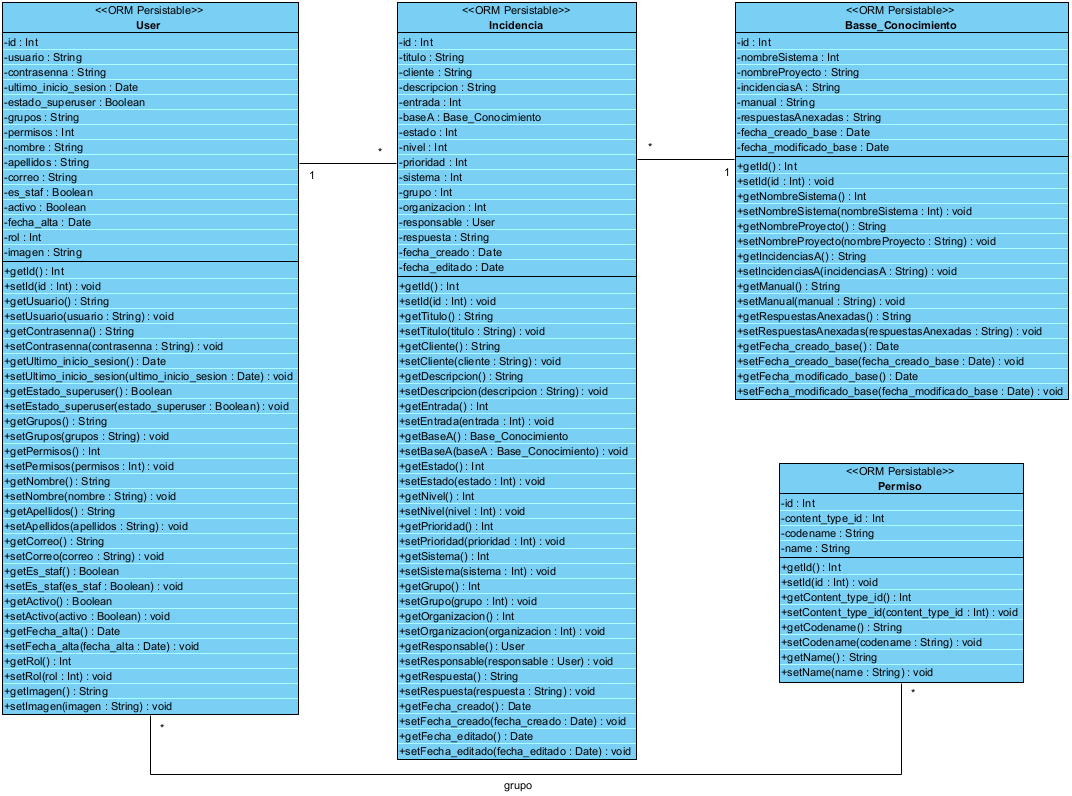
****

Figura 16: Diagrama de Clases Persistentes

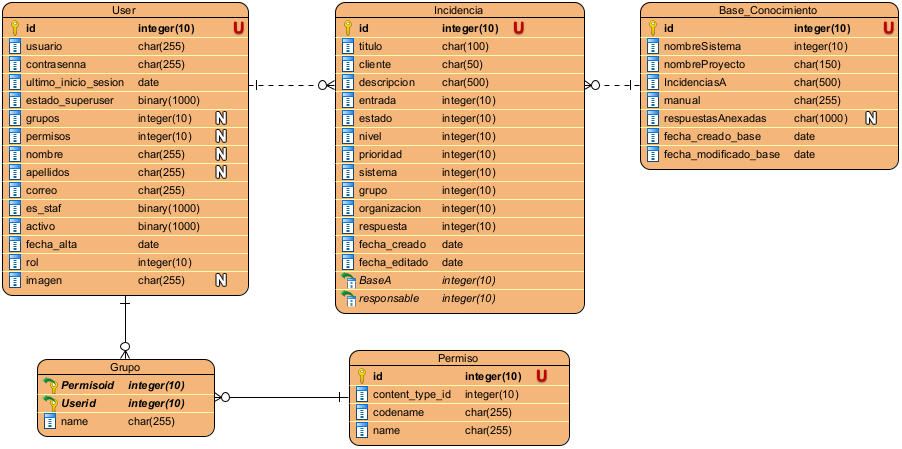
****

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>**

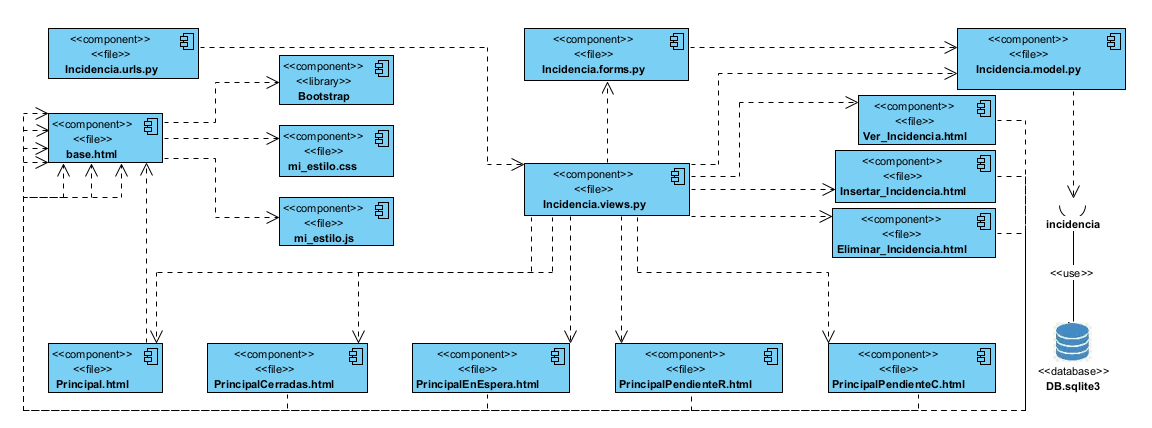
****

Figura 18: Diagrama de Componentes <Gestionar Incidencia>

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

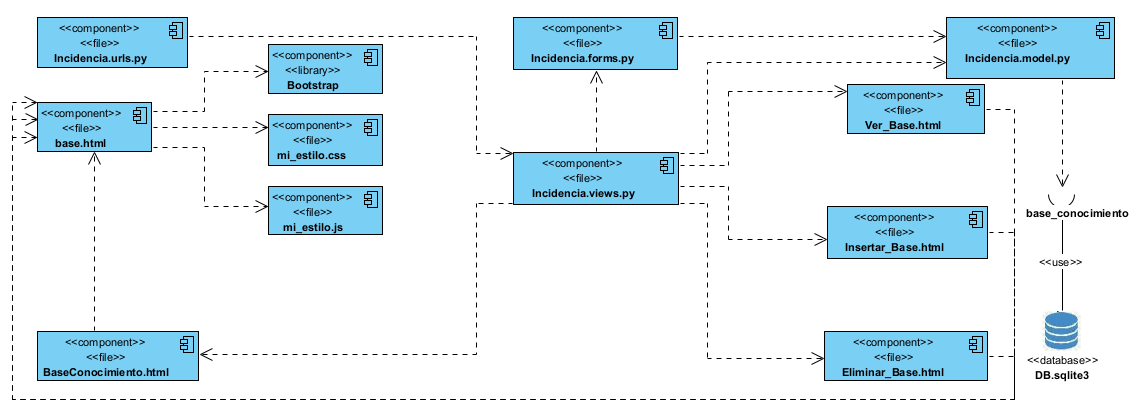
****

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.



Ilustración 1: Interfaz gráfica del Inicio de Sesión del Módulo



Ilustración 2: Página Principal del Módulo



Ilustración 3: Interfaz gráfica de la sección de bases de conocimiento

## 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.

## Conclusiones del capítulo

En general el diseño e implementación del módulo de gestión de incidencias tecnológicas ha sido riguroso y ha seguido buenas prácticas de desarrollo de software pues se basa en un enfoque metodológico sólido. Se utilizaron herramientas de la metodología AUP-UCI en su escenario 4 como diagramas de clases de diseño, complementándose los mismos con la aplicación de los patrones Grasp, mejorando la calidad y mantenibilidad del código, al proporcionar directrices claras para la asignación de responsabilidades y la estructuración del diseño. Los diagramas de paquetes, componentes, secuencia y entidad-relación son otras de las herramientas de la metodología que permitieron una representación clara y estructurada de las diferentes partes del sistema. Gracias a lo planteado se ha facilitado la comprensión y comunicación del diseño del módulo de gestión de incidencias tecnológicas.

# CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

El proceso de verificación implica la revisión de la documentación y el código fuente para asegurarse de que cumple con los requisitos especificados. Por otro lado, la validación implica probar el módulo en un entorno real para asegurarse de que funciona de manera efectiva y cumple con las necesidades del usuario. El capítulo aborda los diseños de casos de pruebas utilizando Caja Negra con la técnica de Partición de Equivalencia y Caja Blanca con la técnica de Camino Básico. Se le realiza al módulo pruebas de Funcionalidad, Fiabilidad (Integridad) y Rendimiento (Carga). Posteriormente se denotan los resultados de la técnica de validación Iadov realizada a los expertos escogidos.

## 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 13: 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 14: 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 15: 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 16: 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 17: 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 18: 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 19: 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 20: Casos de Prueba de la Sección Eliminar Base de Conocimiento

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

Estas pruebas intentan garantizar que se ejecuten al menos una vez todos los caminos independientes que presente el módulo y que todas las estructuras de datos internas sean usadas. Se realizó la prueba a la funcionalidad del módulo que permite saber si existen incidencias que presenten el estado “En Espera”, para lo cual es necesario conocer el número de caminos independientes del algoritmo mediante el cálculo de la complejidad ciclomática.

La complejidad ciclomática, V (G), de un grafo de flujo G se define como: V (G) = A – N + 2, donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.

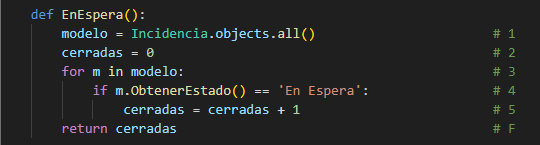


Figura 21: Método escogido para la técnica de Camino Básico

F

V

**V (G) = A – N + 2**

**V (G) = 6 – 6 + 2**

**V (G) = 2**

|  |  |
| --- | --- |
| **Caso de Uso:** | **Gestionar Incidencia** |
| **Caso de Prueba:** | Saber si existen incidencias “En Espera” |
| **Camino#1:** | 1, 2, 3, 4, 5, F |
| **Camino#2:** | 1, 2, 3, 4, F |
| **Entrada para el camino#1:** | m.ObtenerEstado() = “En Espera” |
| **Entrada para el camino#2:** | m.ObtenerEstado() != “En Espera” |
| **Resultados para el camino#1:** | El método devuelve un número distinto de 0 (Cerradas != 0), lo que significaría que se encontraron incidencias que presentan el estado “En Espera”. |
| **Resultados para el camino#2:** | El método devuelve 0 (Cerradas = 0), lo que significaría que no se encontraron incidencias que presentan el estado “En Espera”. |
| **Condiciones:** | 1-El Usuario debe estar autenticado en la plataforma  2-El Usuario debe presentar los permisos requeridos para poder ver el listado de incidencias en Espera. |

Tabla 21: Casos de Prueba (Caja Blanca) incidencias “En Espera”

**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.

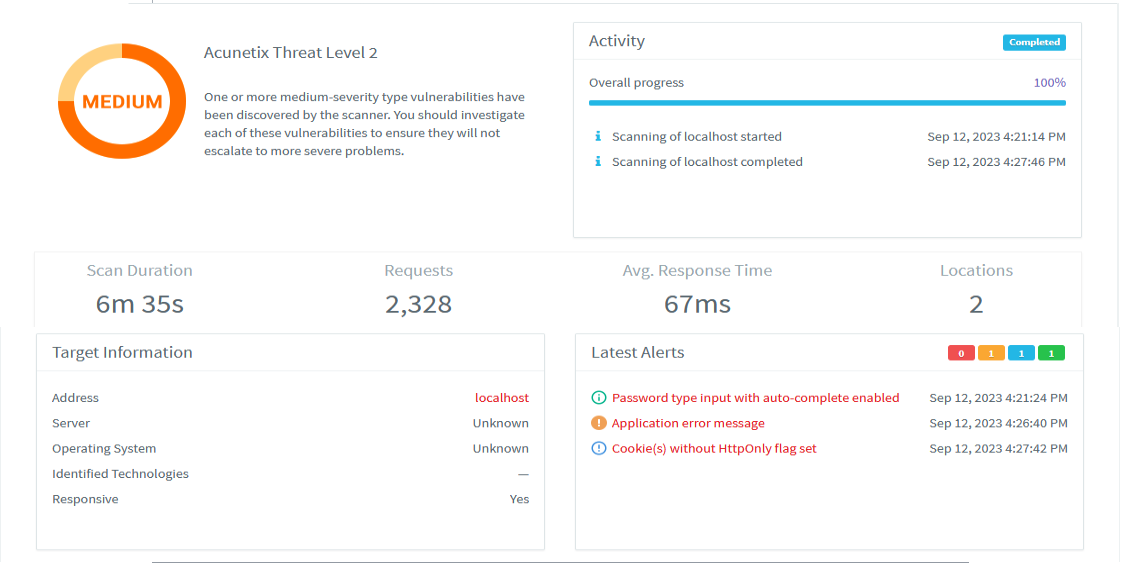


Figura 22: Prueba de Fiabilidad utilizando la herramienta Acunetix Trial Edition

* **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.

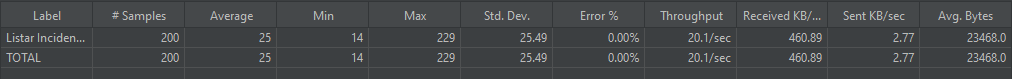


Figura 23: Prueba de Rendimiento (1) utilizando la herramienta Apache JMeter

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.

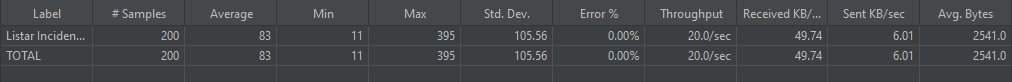


Figura 24: Prueba de Rendimiento (2) utilizando la herramienta Apache JMeter

* **Funcionalidad:**

Se realiza la prueba haciendo uso de la herramienta Selenium IDE para grabar todas las interacciones funcionales con el módulo y verificar errores en las pruebas realizadas.

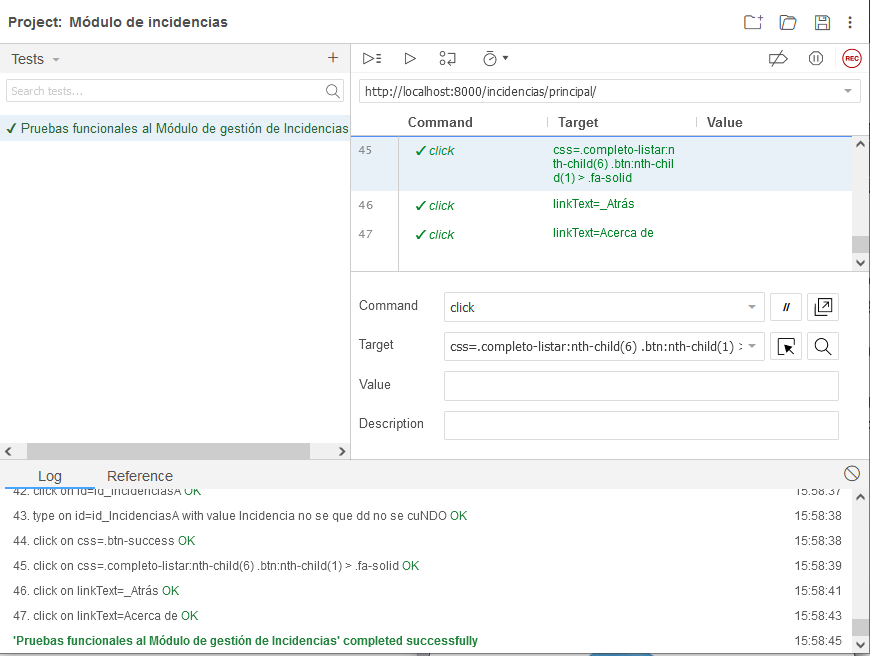


Figura 25: Prueba de Funcionalidad utilizando la herramienta Selenium IDE

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

**Técnica de validación Iadov.**

La técnica de Iadov constituye una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario y cuya relación el sujeto desconoce. Estas tres preguntas se relacionan a través de lo que se denomina el “Cuadro Lógico de Iadov”. El criterio de usuarios, utilizando como procedimiento científico metodológico para el procesamiento de los resultados la Técnica de Iadov, constituye una vía indirecta para el estudio de la satisfacción (Puebla et al., 2016).

Para poder desarrollar esta técnica se aplicó una encuesta, procesando un análisis diferenciado , anteponiendo las preguntas cerradas y posterior las preguntas abiertas.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Considera usted que el módulo realiza una correcta gestión de las incidencias | | | | | | | | |
| Si | | | No Sé | | | No | | |
| Si usted necesitara gestionar incidencias tecnológicas. ¿Usaría el módulo propuesto? | | | | | | | | |
| ¿Le satisface el módulo propuesto para el proceso de gestión de incidencias tecnológicas? | Sí | No  Sé | No | Sí | No  Sé | No | Sí | No  Sé | No |
| Me satisface mucho | 1 | 2 | 6 | 2 | 2 | 6 | 6 | 6 | 6 |
| No me satisface tanto | 2 | 2 | 3 | 2 | 3 | 3 | 6 | 3 | 6 |
| Me da lo mismo | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| Me insatisface más de lo que me satisface | 6 | 3 | 6 | 3 | 4 | 4 | 3 | 4 | 4 |
| No me satisface nada | 6 | 6 | 6 | 6 | 4 | 4 | 6 | 4 | 5 |
| No sé qué decir | 2 | 3 | 6 | 3 | 3 | 3 | 6 | 3 | 4 |

Tabla 22: Cuadro Lógico de Iadov

Iadov presenta una escala de satisfación que permite hacer una valoración de la Satisfacción Individual (SI) de cada experto a partir de sus respuestas a las preguntas cerradas. Al interrelacionar las preguntas cerradas surge un número resultante que indica la posición de cada experto en la escala de satisfacción.

* 1. Clara satisfacción
* 2. Mas satisfecho que insatisfecho
* 3. No definida
* 4. Mas insatisfecho que satisfecho
* 5. Clara insatisfacción
* 6. Contradictoria

A su vez, con esta técnica también se puede obtener el (ISG) Índice de Satisfacción Grupal a travéz de una escala numérica que oscila entre +1 y -1 y el uso de la fórmula a continuación.

ISG = A(+1) + B(+0.5) + C(0) + D(-0.5) + E(-1)

N

En esta fórmula A, B, C, D, E, representan el número de sujetos con índice individual 1; 2; 3 ó 6; 4; 5 y donde N representa el número total de sujetos del grupo. El ISG arroja valores entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0,5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que caen entre 0,5 y 1 indican que existe satisfacción.

|  |  |
| --- | --- |
| **Escala** | **Resultado** |
| +1 | Máximo de satisfacción |
| 0,5 | Más satisfecho que insatisfecho |
| 0 | No definido y contradictorio |
| -0,5 | Más insatisfecho que satisfecho |
| -1 | Máxima insatisfacción |

Tabla 23: Niveles de Satisfacción

## Conclusiones del capítulo

Las pruebas realizadas al módulo de gestión de incidencias tecnológicas fueron exitosas y arrojaron resultados satisfactorios. La prueba de funcionalidad con Selenium Ide permitió verificar que todas las funciones del módulo se ejecutaban correctamente, garantizando su buen funcionamiento. La prueba de integridad con Acunetix trial Edition reveló solo un error de tipo medio lo cual indica que el módulo se encuentra en buen estado y cumple con los estándares de seguridad establecidos. La prueba de Rendimiento con Apache Jmeter demostró que el módulo es capaz de manejar una carga de trabajo adecuada sin experimentar problemas de rendimiento. En resumen, las pruebas realizadas confirmaron la calidad y eficiencia del módulo de gestión de incidencias tecnológicas de soporte.

# CONCLUSIONES FINALES

Después de realizar un estudio exhaustivo de sistemas homólogos y considerar las necesidades del cliente, se puede concluir que la elaboración de un módulo de gestión de incidencias tecnológicas para la plataforma informática del Centro de Soporte de la Universidad de las Ciencias Informáticas es una solución altamente beneficiosa. El sistema anteriormente utilizado en el Centro carecía de una base de conocimiento y una sección de reportes, lo que dificultaba la resolución eficiente de las incidencias tecnológicas. Sin embargo, la propuesta de solución contempla estas funcionalidades, lo que mejorará significativamente la eficiencia y efectividad del Centro de Soporte. La inclusión de una base de conocimiento en el nuevo módulo permitirá almacenar y organizar información relevante sobre las incidencias previamente resueltas, lo que facilitará la búsqueda y aplicación de soluciones en futuros casos similares. Esto ayudará a reducir el tiempo de resolución de incidencias, ya que los técnicos podrán acceder rápidamente a soluciones previamente implementadas. Se escogieron herramientas informáticas, así como lenguajes de programación de gran utilidad para la elaboración de la solución propuesta. Se tuvieron en cuenta en todo momento los requisitos funcionales otorgados por el cliente, concluyendo la creación del módulo con el cumplimiento de los mismos. Para poder llegar a una correcta implementación se diseñaron diagramas necesarios que permitieron avanzar hacia la meta final. Luego de la implementación se le realizaron al módulo pruebas de Integridad, Funcionalidad, Rendimiento, Integración y de validación y aceptación hacia el Usuario final, reafirmando que la elaboración de un módulo de gestión de incidencias tecnológicas con base de conocimiento y sección de reportes es una propuesta sólida que aborda las necesidades del cliente y mejorará significativamente la eficiencia y efectividad del Centro de Soporte de la Universidad de las Ciencias Informáticas.

# RECOMENDACIONES

En primer lugar, me gustaría destacar la importancia de contar con una herramienta eficiente y automatizada para gestionar las incidencias tecnológicas. Por lo que se recomienda la creación de una funcionalidad de gestión automatizada de incidencias tecnológicas que permitirá al Centro de Soporte mejorar la eficiencia en la resolución de problemas. Al automatizar el proceso de registro, asignación y seguimiento de incidencias, se reducirá el tiempo de respuesta y se optimizará la asignación de recursos, lo que resultará en una mejora significativa en la experiencia del usuario. Se agrega también a modo de recomendación lograr la exportación de conocimientos sobre las incidencias filtradas no solo por fecha de creación, sino, por demás características.

# 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.

*Apache JMeter—Apache JMeterTM*. (s. f.). Recuperado 28 de septiembre de 2023, de https://jmeter.apache.org/

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*.

Hiberus. (2023, enero 10). Cómo automatizar pruebas funcionales con Selenium. *Blog de Hiberus Tecnología*. https://www.hiberus.com/crecemos-contigo/como-automatizar-pruebas-funcionales-con-selenium/

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.

Puebla, J. H., Osorio, M. D. L. A., Robaina, F. G., Yunier Díaz Pérez, & Revista Ingeniería Agrícola. (2016). *Grain sorghum (Sorgum vulgare L. Monech) response to irrigation time and nitrogen fertilizer during two plantation dates*. https://doi.org/10.13140/RG.2.2.28326.32325

*¿Que es Acunetix?, Pruebas de Vulnerabilidad Web*. (2021, julio 16). NORTH NETWORKS. https://www.north-networks.com/que-es-acunetix/

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)