



Universidad de las Ciencias
Informáticas

Facultad 1

Módulo de Comisiones Disciplinarias del Sistema de Gestión de Trabajo Educativo de la Facultad 1.

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

Autor(es): Leslie de las Mercedes Prieto Ricardo

Tutor: M. Sc PA. Aylin Estrada Velazco

Ing. Michel Pedrera Suen

La Habana, diciembre 2022

Año 63 de la Revolución

“La Educación es el pasaporte hacia el futuro, el mañana pertenece a aquellos que se preparan para él en el día de hoy “

Malcolm X

“El mayor estímulo para esforzarnos en el estudio y en la vida es el placer del mismo trabajo, el placer de los resultados y la conciencia del valor de los resultados para esa comunidad “

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo, Leslie de las Mercedes Prieto Ricardo con carné de identidad 00010368573, soy la autora del trabajo de diploma con título “Módulo de Comisiones Disciplinarias del Sistema de Gestión de Trabajo Educativo de la Facultad 1”, concedo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. Para que así conste firmo la presente en La Habana, a los ____ días del mes de _____ del año 2022.

Nombre del autor

Leslie de las Mercedes Prieto Ricardo

Firma del Autor

Nombre del tutor

Aylin Estrada Velazco

Firma del Tutor

Nombre del tutor

Michel Pedrera Suen

Firma del Tutor

AGRADECIMIENTOS

A mis padres Belkis y Orlando y mi novio Richard porque gracias a su apoyo incondicional estoy finalizando con éxito mi carrera universitaria. Muchas gracias a mis padres por ser tan buenos y maravillosos, a mi mamá gracias por su amor, preocupación, apoyo y dedicación, a mi papá gracias por su amor y apoyo incondicional.

A mi novio Richard por estar ahí conmigo desde primer año ayudándome en todo, luchando codo a codo conmigo, por sus consejos, por su amor, comprensión y dedicación.

A mis tutores Aylín y Michel y todos los profesores que me han ayudado en este proceso de culminación de estudios, a la profesora Yaili, la profesora Disnayle y el profesor Yaniel por ser tan amables y aclararme todas las dudas siempre que los necesitaba.

A Neiser por su explicación detallada y los materiales que me brindó sobre el tema de tesis.

A los profesores que me han impartido clases en el transcurso de la carrera, aportando sus conocimientos para contribuir a mi formación como profesional, en especial profesores dedicados que recuerdo con mucho cariño de los primeros años Saydanis, José Hilario, Yor Alex, Serguey, Ariel, Juan Manuel, Jorge Sergio.

A todos mis compañeros desde primer año por su gran apoyo y solidaridad, en especial a mi compañera Elizabeth Hidalgo Colomé por su apoyo incondicional siempre y su gran amistad.

A mis compañeros de equipo y de diferentes aulas en diferentes años de la carrera Daniel Peñalver, Richard Alejandro, Eddie Delgado, Pedro Luis, Héctor Andy, Gabriel Antonio, Duanny Jesús, Osman Piedra, Ariel Champagne, Daríel Torres, Roberto Lázaro, Claudia Mulet, Claudia Rodríguez, Gian Michel, Neiser Hernández, Rachel Marichal, Amanda, Beatriz, Sergio Lim que todos de una forma u otra me han ayudado en el transcurso de la carrera siempre que los he necesitado.

A todas mis compañeras de apartamento desde primer año por la agradable convivencia.

Gracias a todos.

DEDICATORIA

A mis padres queridos, a mi novio y mis abuelos, los amo y agradezco su cariño, apoyo, comprensión en todo momento.

RESUMEN

La presente investigación se realizó con el propósito de informatizar el proceso de comisiones disciplinarias para mejorar el trabajo educativo de la Facultad 1 de la Universidad de las Ciencias Informáticas (UCI). Actualmente no se cuenta con un sistema que informatice dicho proceso y se detectan afectaciones en la gestión de la información. Este proceso no se desarrolla con la calidad ni el tiempo requerido, la información de las denuncias no se encuentra siempre disponible, se almacena de manera física, lo que provoca dificultad en el momento de necesitar la información. Se llevó a cabo una investigación utilizando los métodos teóricos y empíricos que permitieron analizar el problema propuesto. Se realizó un estudio acerca de los sistemas existentes diseñados para la gestión de comisiones disciplinarias y procesos afines. Para el desarrollo de la solución se seleccionaron las herramientas, lenguajes y metodologías de desarrollo más adecuadas, garantizando que se encuentre disponible la información en todo momento. Se planteó la propuesta de solución y se especificaron los requisitos funcionales y no funcionales, los elementos fundamentales del diseño y de la arquitectura que se deben tener en cuenta para el desarrollo del módulo, ayudado por la realización del modelado de diagramas. Se realizaron las propuestas de pruebas funcionales, carga y estrés y seguridad para cumplir con el objetivo planteado. El desarrollo de este módulo provee a la Facultad 1 de la UCI de un eficiente y organizado proceso de comisiones disciplinarias.

Palabras clave: Comisión disciplinaria, proceso de gestión, módulo de gestión, denuncias, expedientes disciplinarios.

ABSTRACT

The present investigation was carried out with the purpose of computerizing the process of disciplinary commissions in Faculty 1 of the University of Informatics Sciences (UCI). Currently, there is no system that computerizes this process and there are effects on information management. This process is not carried out with the quality or time required, the information on the complaints is not always available, it is stored physically, which causes difficulties when the information is needed. An investigation was carried out using theoretical and empirical methods that allowed analyzing the proposed problem. A study was carried out on the existing systems designed for the management of disciplinary commissions or related processes. For the development of the solution, the most appropriate development tools, languages and methodologies were selected, guaranteeing that the information is available at all times.

The solution proposal was raised and the functional and non-functional requirements were specified, the fundamental elements of the design and architecture that must be taken into account for the development of the module, helped by the realization of diagram modeling. The proposals for functional tests, load and stress and safety were made so that the proposal meets the stated objective. With the development of this module, Faculty 1 of the UCI is provided with an efficient and organized process of disciplinary commissions.

Keywords: Disciplinary commission, management process, management module, complaints, disciplinary proceedings.

Índice

INTRODUCCIÓN	8
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA SOBRE EL PROCESO DE LAS COMISIONES DISCIPLINARIAS.	14
1.1 Conceptos asociados al proceso de comisiones disciplinarias.....	14
1.2 Sistemas relacionados con la comisión disciplinaria en el mundo.	16
1.3 Soluciones informáticas para gestionar el trabajo de las comisiones disciplinarias desarrolladas en la Universidad de las Ciencias Informáticas.	17
1.4 Valoración de soluciones existentes.	19
1.5 Metodología de desarrollo de software	20
1.6 Tecnologías y herramientas seleccionadas.	22
1.6.1 Laravel 9.8.1	22
1.6.2 PHP 8.0.3	23
1.6.3 Sistema gestor de base de datos.	24
1.6.4 Bootstrap 5.0.2	25
1.6.5 HTML 5	26
1.6.6 Servidor web Apache 2.4.46	26
1.6.7 Visual Paradigm 8.0.....	27
1.6.8 Apache JMeter 5.4.1.....	27
1.6.9 Acunetix Web Vulnerability Scanner 11.0	28
Conclusiones del capítulo.....	28
CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN: MÓDULO DE COMISIONES DISCIPLINARIAS DEL SISTEMA DE GESTIÓN DE TRABAJO EDUCATIVO DE LA FACULTAD 1.....	29
2.1 Descripción del Negocio.....	29
2.1.1 Actores del negocio.	30
2.2 Modelo Conceptual.....	31
2.3 Propuesta de solución.	32
2.4 Especificación de los requisitos del sistema.	32
2.4.1 Requisitos funcionales	34
2.4.2 Requisitos no funcionales	37
2.5 Modelo de Caso de uso del sistema.	40
2.5.1 Actores del sistema.	40
2.5.2 Diagrama de Caso de uso del sistema.....	40
2.5.3 Descripción de casos de uso del sistema.	41
2.6 Descripción de la Arquitectura de Software.	48
2.6 Patrones de diseño.....	50
2.6.1 Patrones GRASP.	51

2.6.2 Patrones GOF	53
2.7 Modelo físico de datos.....	55
2.8 Diagrama de clases de diseño.	56
2.9 Diagramas de secuencia.	57
2.10 Diagrama de despliegue.....	58
Conclusiones del capítulo.....	59
CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN: MÓDULO DE COMISIONES	
DISCIPLINARIAS DEL SISTEMA DE GESTIÓN DE TRABAJO EDUCATIVO DE LA	
FACULTAD 1.....	60
3.1 Diagrama de componentes.	60
3.2 Estándares de codificación.	61
3.3 Validación de la solución.	63
3.3.1 Pruebas funcionales	64
3.3.2 Pruebas de seguridad.	68
3.3.3 Pruebas de carga y estrés.....	69
3.3.4 Pruebas de integración.	71
3.4 Validación de los resultados de la investigación aplicando la técnica ladov.	73
Conclusiones del capítulo.....	75
CONCLUSIONES FINALES	76
RECOMENDACIONES.....	77
REFERENCIAS BIBLIOGRÁFICAS	78
ANEXOS.....	82

ÍNDICE DE TABLAS

TABLE 1. ESTUDIO DE HOMÓLOGOS	19
TABLE 2. ACTORES DEL NEGOCIO	30
TABLE 3. REQUISITOS FUNCIONALES	34
TABLE 4. ACTORES DEL SISTEMA	40
TABLE 5. DESCRIPCIÓN DEL CASO DEL USO GESTIONAR DENUNCIA	41
TABLE 6. VALIDACIÓN DE LA SOLUCIÓN.....	64
TABLE 7. DESCRIPCIÓN DE LAS VARIABLES DEL CASO DE PRUEBA 1 BASADO EN EL CU- CREAR DENUNCIA.	65
TABLE 8. CASO DE PRUEBA DEL RF1_CREAR DENUNCIA.	65
TABLE 9. RESULTADO DE LAS PRUEBAS DE CARGA Y ESTRÉS.....	71
TABLE 10. ESCALA NUMÉRICA PARA EL ISG.	73
TABLE 11. RESULTADOS DE APLICAR EL TEST DE IADOV.....	74
TABLE 12. ENTREVISTA MÉTODO EMPÍRICO.	82
TABLE 13. ENTREVISTA PARA DEFINIR REQUISITOS.	83
TABLE 14. CUADRO LÓGICO DE IADOV PARA USUARIOS POTENCIALES.	84

ÍNDICE DE FIGURAS

FIGURE 1. MODELO CONCEPTUAL	31
FIGURE 2. ISO 25010.....	37
FIGURE 3. DIAGRAMA DE CASO DE USO DEL SISTEMA.....	41
FIGURE 4. FUNCIONAMIENTO DEL ESTILO ARQUITECTÓNICO MODELO-VISTA-CONTROLADOR.....	48
FIGURE 5. FUNCIONAMIENTO DEL ESTILO M-V-C EN LARAVEL	49
FIGURE 6. EJEMPLO DE PATRÓN CREADOR EN LA CLASE CONTROLADORA COMISION_DISCIPLINARIACONTROLLER.PHP.....	51
FIGURE 7. EJEMPLO DEL PATRÓN CONTROLADOR EN LA CLASE CONTROLADORA EXPEDIENTECONTROLLER.PHP.	52
FIGURE 8. EJEMPLO DEL PATRÓN EXPERTO EN LA CLASE DENUNCIA.PHP.....	52
FIGURE 9. EJEMPLO DEL PATRÓN ALTA COHESIÓN EN LA CLASE ENTIDAD EXPEDIENTE.PHP.	53
FIGURE 10. EJEMPLO DEL PATRÓN DECORADOR UTILIZANDO LA PLANTILLA ADMINLTE.	54
FIGURE 11. MODELO FÍSICO DE DATOS.	55
FIGURE 12. DIAGRAMA DE CLASES CON ESTEREOTIPOS WEB "GESTIONAR DENUNCIA".....	56
FIGURE 13. DIAGRAMA DE SECUENCIA PARA EL CASO DE USO GESTIONAR DENUNCIA ESCENARIO CREAR DENUNCIA.	57
FIGURE 14. DIAGRAMA DE DESPLIEGUE	58
FIGURE 15. DIAGRAMA DE COMPONENTES DEL CASO DE USO GESTIONAR DENUNCIA.	60
FIGURE 16. EJEMPLO EN EL CÓDIGO	62
FIGURE 17. EJEMPLO DE ETIQUETA DE APERTURA EN EL CÓDIGO.	62
FIGURE 18. EJEMPLO DE NOMBRE DE UNA CLASE EN EL CÓDIGO.....	63
FIGURE 19. EJEMPLO DE NOMBRES DE VARIABLES EN EL CÓDIGO.	63
FIGURE 20. COMPORTAMIENTO DE LAS NO CONFORMIDADES POR ITERACIONES EN LA PRUEBA DE CAJA NEGRA.....	66
FIGURE 21. RESULTADOS PRUEBA DE SEGURIDAD.....	68
FIGURE 22. RESULTADOS PRUEBAS DE SEGURIDAD.	69
FIGURE 23. GITHUB.	72
FIGURE 24. RESULTADOS DE SATISFACCIÓN (IADOV).	74

INTRODUCCIÓN

La época actual demanda la formación de ciudadanos educados en la condición humana y social, capaces de pensar el futuro de la sociedad y de intercambiar valor con ella mediante el ejercicio profesional. Las universidades tienen ante sí el inmenso reto de transformarse a sí mismas para estar en capacidad de contribuir sensiblemente en la formación integral de sus estudiantes. La educación superior está llamada a consolidar el carácter, la personalidad y la ética. La formación integral va más allá de la capacitación profesional, aunque la incluye. Contribuye a enriquecer el proceso de socialización del estudiante y su desarrollo moral. La educación que brinda la universidad es integral en la medida en que enfoque a la persona del estudiante como una totalidad y que no lo considere únicamente en su potencial cognoscitivo o en su capacidad para el quehacer técnico o profesional (Alarcón Ortiz, 2019).

El Comité Ejecutivo del Consejo de Ministros adoptó el Acuerdo No. 4001 de 24 de abril del 2001, el cual establece que el Ministerio de Educación Superior es la Institución encargada de dirigir la Educación Superior en Cuba, con el objetivo de aplicar la política educacional en el nivel de la enseñanza superior y dirigirla metodológicamente. El Reglamento disciplinario para los estudiantes de la educación superior establece las regulaciones asociadas a la disciplina estudiantil, la calificación de las faltas disciplinarias, el procedimiento a seguir por las comisiones disciplinarias, las medidas que se deben tomar, entre otras disposiciones (Resolución No. 240, 2007).

Este documento refleja que las faltas disciplinarias que se cometen por los estudiantes de la educación superior matriculados en los Centros de carácter civil, serán conocidas, juzgadas y sancionadas. Las indisciplinas deben ponerse en conocimiento de los Vicerrectores, Decanos, Vicedecanos, directores de Filiales, o Sedes Universitarias Municipales, de Unidades Docentes, jefes de Departamento o miembros del personal docente, tan pronto se produzcan o se conozcan por cualquier trabajador o estudiante del centro de educación superior. Las Comisiones Disciplinarias designadas en cada Universidad o Facultad universitaria reciben las denuncias por indisciplinas y posteriormente deberán seguir el procedimiento establecido, el cual consiste en analizar las pruebas, tomar declaraciones sobre los hechos e interrogar a testigos y otros implicados (Resolución No. 240, 2007).

La Universidad de las Ciencias Informáticas (UCI), tiene la misión de contribuir a la transformación digital de la sociedad cubana, mediante la formación integral y continua de profesionales de las ciencias informáticas comprometidos con su Patria, así como la producción y comercialización de productos y servicios informáticos, aplicando ciencia e innovación, con proyección internacional y responsabilidad social para la sostenibilidad de la nación socialista. La misma consta de 6 facultades con una red de centros de desarrollo (Universidad de las Ciencias Informáticas, 2022). El Reglamento disciplinario establece el procedimiento a seguir por las comisiones disciplinarias, pero aun así cada Facultad de la UCI tiene particularidades en cuanto a la cantidad de comisiones y la distribución del trabajo de cada una.

De acuerdo a la investigación realizada y a la utilización de métodos empíricos se ha detectado que en la actualidad en la Facultad 1 de la UCI, este proceso se realiza de forma manual lo que entorpece la gestión de la información que se maneja, propicia pérdida de tiempo y atenta contra el buen desempeño de la comisión y el cumplimiento de los términos establecidos para su trabajo.

Al inicio de cada curso escolar se nombran las comisiones y no se ha logrado contar con estabilidad en el personal, experiencia en su trabajo y dominio de los documentos rectores, por lo que es necesario dedicar tiempo de manera planificada a su capacitación. Se han identificado diferentes factores que atentan contra la calidad de este proceso: las denuncias no son notificadas a tiempo a la comisión que debe tramitarla, se atrasan las entrevistas a los testigos o implicados, el cierre del proceso disciplinario no se realiza en el tiempo establecido, los documentos que guardan la información referente no cuentan con buena calidad y organización. La gestión manual de la información de este proceso no ha facilitado que se realicen análisis asociados a la cantidad de denuncias tramitadas y concluidas en cada año académico, al tipo de indisciplina cometida o de medida aplicada, lo que ha afectado la realización de una correcto y eficiente proceso en función de elevar el trabajo educativo en la facultad.

A partir de la situación problemática antes descrita se propone el siguiente **problema de investigación**: ¿Cómo contribuir a la gestión del proceso de las comisiones disciplinarias para mejorar el trabajo educativo de la Facultad 1 de la UCI? Se define como **objeto de estudio**: el proceso de gestión del trabajo educativo de la Facultad 1, enmarcado en el **campo de acción**: la gestión de las comisiones disciplinarias para el trabajo educativo de la Facultad 1 de la UCI.

El **objetivo general** de esta investigación es desarrollar un módulo que permita la gestión del proceso de las comisiones disciplinarias para mejorar el trabajo educativo de la Facultad 1 de la UCI.

Las preguntas científicas que se definen para guiar la investigación son:

1. ¿Cuáles son los referentes teóricos fundamentales que sustentan la investigación relacionados con el proceso de comisión disciplinaria en la Facultad 1 de la UCI?
2. ¿Cuáles son las características que debe cumplir el módulo de gestión del proceso de comisión disciplinaria en la Facultad 1 de la UCI?
3. ¿Cómo validar el funcionamiento del módulo de gestión del proceso de comisión disciplinaria en la Facultad 1 de la UCI?
4. ¿Cómo evaluar la propuesta de solución del módulo de gestión del proceso de comisión disciplinaria en la Facultad 1 de la UCI?

Con el propósito de darle cumplimiento al objetivo general y al problema anteriormente planteado se trazaron las siguientes tareas de la investigación:

1. Estudio de los referentes teóricos fundamentales que sustentan la investigación relacionados con el proceso de comisión disciplinaria en la Facultad 1 de la UCI.
2. Implementación de las funcionalidades del módulo para la gestión del proceso comisión disciplinaria en la Facultad 1 de la UCI.
3. Validación de las funcionalidades del módulo de gestión del proceso de comisión disciplinaria en la Facultad 1 de la UCI.
4. Validación de la propuesta de solución del módulo de gestión del proceso de comisión disciplinaria en la Facultad 1 de la UCI.

Para el cumplimiento de las tareas antes descritas se utilizan los siguientes **métodos científicos** de investigación:

Métodos teóricos:

- **Historio-lógico:** Favoreció el estudio de los antecedentes, el desarrollo, las regularidades y tendencias actuales de gestión, planificación de las comisiones disciplinarias, así como los sistemas existentes en el mundo y en la Universidad de las Ciencias Informáticas (Rodríguez et al., 2015).
- **Analítico-sintético:** Consiste en analizar y conceptualizar lo que queremos estudiar para, a continuación, extraer lo esencial. Permitió el procesamiento de la información y arribar a las conclusiones prácticas y teóricas de la investigación, así como precisar las herramientas utilizadas para el diseño y la implementación del módulo (Rus Arias, 2022).
- **Modelación:** Se utilizó un lenguaje de modelado unificado (UML), para mostrar los diferentes artefactos que se diseñaron como resultado del proceso de ingeniería de software. Estos permitieron describir las funcionalidades de los procesos, así como la interacción de los actores con estos. El modelado de estos artefactos está en correspondencia con los estándares de la metodología AUP-UCI escenario 2 (Barreda & Medina, 2019).

Métodos empíricos:

- **Entrevista:** Para el análisis más detallado sobre los procesos asociados a la comisión disciplinaria, se entrevistan a los principales profesores que se involucran en cada etapa (Ávila et al., 2020).
- **Observación:** se utiliza para comprender cómo se lleva a cabo de forma objetiva el proceso de comisión disciplinaria en cada una de sus etapas (La observación como método empírico de investigación, 2019).

Para la presente investigación se empleará como **técnica de validación** el siguiente:

Fabre y Ortega (2019) señalan que la técnica de ladov 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 (preguntas 1, 3, 5) del cuestionario aplicado y cuya relación el sujeto desconoce. Estas tres preguntas se relacionan a través de lo que se denomina el "Cuadro Lógico de ladov". El grado de satisfacción-insatisfacción, es un estado psicológico que se manifiesta en las personas como expresión de la interacción de un conjunto de vivencias afectivas que se mueven entre los polos positivo y negativo en la medida en que en la actividad que el sujeto desarrolla, el objeto, da respuesta a sus necesidades y se corresponde con sus motivos e intereses. Se recomienda su empleo como vía para valorar resultados en aquellos casos en que los evaluadores son usuarios de lo que se propone, lo cual es de utilidad para la presente investigación, pues se considera pertinente tener en cuenta el criterio de los profesores miembros de las comisiones disciplinarias.

Con el cumplimiento de cada tarea y actividad desarrollada en el proceso de investigación se anticipan los siguientes resultados:

- Un módulo de gestión que permita llevar a cabo todos los procesos asociados a la comisión disciplinaria de la Facultad 1 de la UCI.
- Los artefactos y documentación técnica correspondientes al diseño e implementación del módulo.

Para el desarrollo del presente trabajo se proponen tres capítulos, los cuales están estructurados de la siguiente forma:

CAPÍTULO 1: Fundamentación teórica sobre el proceso de las Comisiones Disciplinarias: En este capítulo se realiza un estudio acerca de los sistemas existentes diseñados para la realización de análisis disciplinarios o procesos afines. En el capítulo también se estudiaron las diferentes metodologías y herramientas que pueden ser empleadas para implementar el Módulo Comisiones Disciplinarias.

CAPÍTULO 2: Análisis y diseño de la solución: Módulo de Comisiones Disciplinarias del Sistema de Gestión de Trabajo Educativo de la Facultad 1: Se explica cómo se desarrolla el flujo actual de los procesos, y se realiza la propuesta de solución para resolver el problema planteado. Por otra parte, se especifican los requisitos funcionales y no funcionales, los elementos fundamentales del diseño y de la arquitectura que se deben tener en cuenta para el desarrollo del módulo, ayudado por la realización del modelado de diagramas.

CAPÍTULO 3: Implementación y validación de la solución: Módulo de Comisiones Disciplinarias del Sistema de Gestión de Trabajo Educativo de la Facultad 1: Se implementa el módulo para darle solución a la problemática planteada. Se realizan las pruebas funcionales, carga y estrés y seguridad para que la propuesta cumpla con el objetivo planteado.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA SOBRE EL PROCESO DE LAS COMISIONES DISCIPLINARIAS.

El presente capítulo aborda los elementos que permitieron realizar una correcta caracterización del módulo, sustentados en un estudio de los conceptos fundamentales a los que está relacionado. Se realiza un análisis de sistemas existentes en el mundo que informatizan procesos relacionados con el módulo a desarrollar. Se ofrece el resultado de las principales herramientas, plataforma de desarrollo y metodología de desarrollo de software a utilizar para el desarrollo.

1.1 Conceptos asociados al proceso de comisiones disciplinarias.

Según el diccionario de la Real Academia Española RAE (2022), la disciplina es una doctrina, instrucción de una persona, especialmente en lo moral. Acción y efecto de disciplinar. Una falta disciplinaria es aquella que atenta contra la disciplina. La indisciplina escolar son las acciones, palabras, actitudes, gestos y reacciones que contrarían las normas disciplinarias vigentes en un centro de enseñanza, o que representan atentados contra la moral, la autoridad, el orden, el espíritu y las tradiciones de la institución (Deisy, 2015). Un reglamento es una colección ordenada de reglas o preceptos, que por la autoridad competente se da para la ejecución de una ley o para el régimen de una corporación, una dependencia o un servicio (RAE, 2022).

El Reglamento Disciplinario para los estudiantes de la educación superior, establece que las faltas disciplinarias se clasifican en Muy graves, Graves y Menos graves. Cuando un estudiante comete un falta disciplinaria se realiza una denuncia, documento en que se da noticia a la autoridad competente, de la comisión de un delito o de una falta (RAE, 2022). Las denuncias son analizadas por las comisiones disciplinarias, estas son un conjunto de personas encargadas por la ley, o por una corporación o autoridad, de ejercer unas determinadas competencias permanentes o entender en algún asunto específico (RAE, 2022). Estas comisiones están compuestas por dos profesores, de los cuales uno actuará como presidente y el otro como secretario y además estarán constituidas por un miembro de la Federación Estudiantil Universitaria (FEU). Se encargan también de tomar las declaraciones, las opiniones y las evidencias de las denuncias (Resolución No. 240, 2007).

Las comisiones disciplinarias son las encargadas de crear los expedientes a los denunciados, un expediente disciplinario es un procedimiento seguido contra el acusado de cometer una infracción disciplinaria. Una declaración es la manifestación o explicación de lo que otro u otros dudan o ignoran. Una opinión es un juicio o valoración que se forma una persona respecto de algo o de alguien (RAE, 2022). A los expedientes disciplinarios se le realiza un dictamen final con el objetivo de decidir la medida disciplinaria a imponer al estudiante denunciado. Según RAE (2022), un dictamen es una opinión y juicio que se forma o emite sobre algo. Una medida disciplinaria es una sanción, castigo por una falta a las normas, tomada por una autoridad competente (Diccionario Español abierto, libre y social, 2010).

En el Reglamento disciplinario, Resolución No.240 del 2007 se establece que las medidas a imponer pueden ser por:

Faltas Muy Graves:

- 1- Expulsión de la Educación Superior.
- 2- Separación indefinida de la Educación Superior.
- 3- Separación de 3 a 5 cursos de la Educación Superior.

Faltas Graves:

- 1- Separación hasta tres cursos de la educación superior.
- 2- Pérdida de sus derechos como becario por un semestre y hasta dos cursos.

Faltas Menos Graves:

- 1- Separación por uno o dos cursos de la educación superior.
- 2- Amonestación pública ante el colectivo estudiantil.
- 3- Pérdida de sus derechos como becario de uno a seis meses.
- 4- Ofrecer una satisfacción al estudiante, trabajador, persona o colectivo que haya ofendido.
- 5- Reparar cuando ello sea posible, en el plazo perentorio que se le fije, el daño ocasionado.

1.2 Sistemas relacionados con la comisión disciplinaria en el mundo.

En la actualidad la informatización de todo tipo de procesos es primordial para contribuir a su correcto y eficiente desarrollo. El sistema educativo incluye un sinnúmero de procesos que necesitan mejorar su funcionamiento mediante informatización, en el caso de los procesos disciplinarios es muy importante el desarrollo de sistemas que contribuyan a gestionarlos de manera eficaz. Se realizó una investigación de los sistemas que existen internacionalmente para la gestión de comisiones disciplinarias o procesos afines y se seleccionaron los siguientes sistemas debido a que fueron los que más aportaron a la investigación teniendo en cuenta sus características y funcionalidades, ya que son sistemas web y gestionan información referente a incidencias y procesos disciplinarios.

Dumbo: Portal Integrado para la Gestión de Peticiones de Cambio e Incidencias de la Universidad de Murcia (Urbina, 2013).

- Herramienta para la gestión de incidencias y atención a usuarios.
- Solución tecnológicamente avanzada.
- Sistema web apoyado sobre una arquitectura de tres capas.
- Garantiza fácil acceso.

Este sistema es privativo y ajustado a las necesidades propias del cliente. El mismo no es multiplataforma, solo opera en el sistema Windows. Gestiona varios procesos de los cuales se pueden tomar ideas a la hora de definir los requisitos como son:

- Solicitud de nuevas incidencias.
- Exploración de incidencias solicitadas.
- Atención de incidencias recibidas.
- Gestión de la cola genérica de incidencias.
- Administración del sistema.
- Notificaciones.

Orión gestión de Procesos Disciplinarios: Solución para Manejo de Procesos Disciplinarios (Orión gestión de Procesos Disciplinarios, 2022).

- Facilita la administración de procesos y control de vencimiento de términos (Quejas Disciplinarias, Investigación Disciplinaria y Proceso Disciplinario).
- Permite el registro y asignación de procesos.
- Actualiza y consulta su información básica e histórica.
- Control de gestiones realizadas por los usuarios que participan en su ejecución.
- Orión cuenta con tecnología Workflow que permite diseñar y automatizar procedimientos asegurando que los términos legales e internos se cumplan.
- Permite la administración de documentos (digitalizados y demás) asociándolos a los expedientes y visualizándolos junto con la información del sistema.
- Posee vistas especializadas de agenda y vencimiento de términos para el seguimiento de compromisos.
- El sistema es adaptable a las necesidades de su Compañía.
- Genera notificación de recordatorios al correo corporativo o personal.
- La solución cuenta con un generador dinámico de consultas y reportes para el suministro de información oportuna y veraz a organismos de control y de cooperación.
- Conecta en línea a personal propio de su organización y personal externo ya que se puede acceder a ORION desde la Intranet e Internet.

1.3 Soluciones informáticas para gestionar el trabajo de las comisiones disciplinarias desarrolladas en la Universidad de las Ciencias Informáticas.

En la Universidad de las Ciencias Informáticas se han realizado múltiples intentos de automatizar este proceso disciplinario, entre estos se encuentran:

- CODIS: Sistema para la informatización del proceso de Comisión Disciplinaria de la Facultad 3. Este sistema fue desarrollado en el 2012 y abarca todo el trabajo que realizan las comisiones disciplinarias. Entre sus principales funcionalidades se encuentran la gestión de las comisiones, los implicados, los casos, las denuncias, opiniones y expedientes. La generación de reportes es muy pobre, sólo permite visualizar los casos por cursos. Por otra parte, no realiza citaciones y notificaciones por correo (Fernández Castillo et al., 2012).

- SIPCD: Desarrollo de una aplicación informática para el proceso de gestión de las comisiones disciplinarias en la Facultad 3.

Este sistema fue creado en el 2014. El mismo posee como principales funcionalidades la gestión de usuarios, comisiones, denuncias, opiniones, declaraciones, permite reportes de estudiantes reincidentes, con condicional y por tipo de sanciones. Incorpora la gestión de plantillas que se usan en el expediente del proceso, así como la gestión de las apelaciones. Permite realizar notificaciones, no así alertas por proximidad de las fechas y tiempos establecidos (LavinScott, 2014).

- SIPCD versión 2: Aplicación informática para el proceso de gestión de las comisiones disciplinarias en la Facultad 3.

Este sistema fue creado en el 2015. Dicho sistema posee como principales funcionalidades se encuentran la gestión de usuarios, denuncias, comisiones, notificación, reporte, términos de tiempo, plantillas, incluye el proceso de apelación, revisión y suspensión condicional de la medida, no soportado completamente en la versión anterior del sistema. Incorpora la Gestión de reporte estadístico del proceso y permite la realización de búsquedas (Matos & García, 2015).

- CDis: Sistema Informático para la gestión de la información de expedientes disciplinarios de la Facultad 2.

Este sistema fue desarrollado en el 2017 y basa su trabajo en el proceso de comisiones disciplinarias. Posee como principales funcionalidades la gestión de usuarios, denuncias, comisiones, expedientes, resoluciones, reportes sobre el estado de las denuncias y de los expedientes, no así de los estudiantes procesados, reincidentes u otros de interés para la Facultad. Cuenta con una barra de progreso del estado terminación de los expedientes. No cuenta con Alertas y notificaciones (Yrzula & Aranda, 2017).

- ControlID: Sistema para la gestión y control de la información generada por las Comisiones Disciplinarias de la Facultad 1.

Este sistema fue desarrollado en el 2021, está basado en una propuesta de solución, contiene requisitos funcionales de gestión de usuario, denuncias, comisiones, expedientes, declaraciones, opiniones, dictamen, notificaciones, reportes, denuncias procesadas por año, denuncias por estado por año, comisiones por estado, alertas y búsqueda avanzada (Almaguer Millan, 2021).

1.4 Valoración de soluciones existentes.

En la tabla se muestra varios indicadores que se definieron para el análisis de las soluciones existentes teniendo en cuenta las principales funcionalidades y si el software es libre o privativo:

Table 1. Estudio de homólogos

Sistema	Gestionar expedientes	Gestionar dictamen	Generar documentos de todo el proceso	Generar reportes	Gestionar Denuncias	Gestionar declaraciones	Gestionar comisiones	Software libre
Dumbo	No	Si	Si	Si	Si	No	No	No
Orión	Si	Si	Si	Si	Si	Si	Si	No
CODIS	Si	Si	No	Si	Si	Si	Si	Si
SIPCD	Si	Si	No	Si	Si	Si	Si	Si
SIPCDv2	Si	Si	No	Si	Si	Si	Si	Si
CDIs	Si	Si	No	Si	Si	Si	Si	Si

Luego de analizados los sistemas internacionales Dumbo y Orión se llega a la conclusión de que no pueden ser utilizados en la UCI porque están implementados específicamente para atender a las necesidades de los lugares donde fueron realizados y son privativos.

Los sistemas encontrados en la UCI no se encuentran desplegados. Tienen requisitos funcionales muy similares, por ejemplo, todos permiten autenticarse, gestionar denuncias y comisiones, aunque algunos presentan funcionalidades que otros no, por ejemplo, los sistemas SIPCD, SIPCD en su versión 2 y ControlID presentan alertas y notificaciones y CODIS y CDIS no presentan estas funcionalidades. También entre los sistemas SIPCD, SIPCD en su versión 2 existen diferencias y la versión 2 está mucho más actualizada con nuevas funcionalidades. En el caso de CODIS y SIPCD, CODIS contiene una generación de reportes muy pobre y SIPCD permite reportes de estudiantes reincidentes, con condicional y por tipo de sanciones. Los sistemas CDIS y ControlID permiten exportar documentos al formato PDF y los otros no.

Después de este estudio se toman en cuenta para el desarrollo del Módulo de Comisiones Disciplinarias del Sistema de Gestión de Trabajo Educativo de la Facultad 1 las funcionalidades de los sistemas anteriormente descritos gestionar denuncias, comisiones, declaraciones, opiniones, evidencias, expedientes, dictamen, reportes y documentos en PDF.

Existe la necesidad de desarrollar para la Facultad 1 de la UCI un Módulo de Comisiones Disciplinarias que contribuya al correcto y eficaz desarrollo de su proceso, además este módulo debe ser capaz de adaptarse al Sistema de Gestión de Trabajo Educativo para su correcta integración, por lo que las soluciones hasta ahora propuestas no responden a esta necesidad ya que no son módulos y no utilizan las herramientas necesarias para cumplir con el objetivo.

1.5 Metodología de desarrollo de software

AUP-UCI

La Universidad de las Ciencias Informáticas (UCI) 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. Está formada por tres fases, (Inicio, Ejecución y Cierre) para el ciclo de vida de los proyectos de la universidad, las cuales contienen las características de las cuatro fases Inicio, Elaboración, Construcción y Transición propuestas en AUP (Variación de AUP para la UCI (AUP-UCI) - Metodologías tradicionales, 2022).

Las características de las fases de la metodología de la universidad son:

- Inicio:

Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.

- Ejecución:

En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, se obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto. Durante esta fase el software es transferido al ambiente de los usuarios finales o entregado al cliente junto con la documentación. Además, en esta transición se capacita a los usuarios finales sobre la utilización de la aplicación.

- Cierre:

En el cierre se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

La metodología de software AUP-UCI a partir de que el modelado de negocio propone tres variantes a utilizar en los proyectos, como son: CUN (Casos de uso del negocio), DPN (Descripción de proceso de negocio) o MC (Modelo conceptual) y existen tres formas de encapsular los requisitos, los cuales son: CUS (Casos de uso del sistema), HU (Historias de usuario), DRP (Descripción de requisitos por proceso) (Variación de AUP para la UCI (AUP-UCI) - Metodologías tradicionales, 2022).

Surgen cuatro escenarios para modelar el sistema en los proyectos, los cuales son:

- Escenario No. 1: Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.
- Escenario No. 2: Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.
- Escenario No. 3: Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.
- Escenario No. 4: Proyectos que no modelen negocio solo pueden modelar el sistema con HU.

Luego del análisis de la investigación realizada, se usa la variante 3 (MC) y el Escenario No. 2 para el desarrollo del sistema. Este proceso es complejo y requiere de un estudio y análisis detallado del proceso de comisiones disciplinarias de la Facultad 1 de la UCI. La identificación de las clases que intervienen en el negocio permite la comprensión de todo el proceso, todas las actividades que se realizan y su relación entre las clases. El Escenario No. 2 garantiza el análisis concreto del negocio y una disminución de las probabilidades de fracaso en el desarrollo del sistema.

1.6 Tecnologías y herramientas seleccionadas.

1.6.1 Laravel 9.8.1

Laravel es un framework de PHP de código abierto utilizado para crear sitios web y aplicaciones escritas en este lenguaje de programación. Este framework ayuda en muchos aspectos al desarrollar una aplicación, por medio de sus sistemas de paquetes y de ser un framework del tipo MVC (Modelo-Vista-Controlador) dando como resultado facilidad de desarrollo en varios aspectos, cómo instanciar clases y métodos para su uso en diferentes partes de la aplicación sin la necesidad de escribirlo y repetirlo numerosas veces. Trabaja con una arquitectura de carpetas avanzada, de modo que promueve la separación de los archivos con un orden correcto y definido, que guiará a todos los integrantes del equipo de trabajo y será un estándar a lo largo de los distintos proyectos. Por supuesto, dispone también de una arquitectura de clases también muy adecuada, que promueve la separación del código por responsabilidades. Laravel dispone de un potente sistema de comandos de consola para resumir muchas tareas tediosas y repetitivas. Artisan, la herramienta de línea de comandos de Laravel ha conseguido que la experiencia de desarrollo con Laravel sea muy atractiva para las personas que se han decidido por este framework (Torrejón, 2022) .

Zúñiga (2015) señala que Laravel contiene un amplio conjunto de características, que sirven para realizar la mayoría de las aplicaciones web. Entre ellas podemos encontrar:

- Un sistema de rutas, mediante las cuales es fácil crear y mantener todo tipo de URLs amistosas a usuarios y buscadores, rutas de API, etc.
- Un sistema de abstracción de base de datos, con un ORM potente pero sencillo de manejar, mediante el que podemos tratar los datos de la base de datos como si fueran simples objetos.
- Un sistema para creación de colas de trabajo, de modo que es posible enviar tareas para ejecución en *background* y aumentar el rendimiento de las aplicaciones.
- Un sistema de notificaciones a usuarios, mediante email, base de datos y otros canales.

- Sistema de autenticación, con todo lo necesario como recordatorios de clave, confirmación de cuentas, recordar un usuario logueado, etc.

Algunos de los servicios y herramientas para el framework Laravel más populares y útiles son los siguientes:

- Forge: es un sistema de administración de servidores, como una especie de panel con el que puedes crear dominios, instalar Laravel, desplegar aplicaciones, etc. Pone a los desarrolladores toda una serie de herramientas para que instalar aplicaciones Laravel en producción sea muy sencillo.
- Nova: es un sistema de gestión de contenido que permite crear los típicos paneles de administración de una manera muy cómoda. Es muy versátil y ahorra cientos de horas de programación que muchas veces son tediosas y repetitivas (Torrejón, 2022).

Para el desarrollo de este módulo se emplea la versión Laravel 9.8.1. La elección de este framework posibilita un desarrollo más rápido del módulo debido a la versatilidad en el desarrollo de aplicaciones web con el uso de un rendimiento mejorado y seguridad de clase alta. Además, permite construir un módulo de gestión que responda a las necesidades del cliente, brindar soporte por un período de tiempo extenso, debido a que las futuras versiones de Laravel serán compatibles a la anterior, por lo tanto, facilitarán la migración. Otro elemento que justifica la elección es que el marco de Laravel está evolucionando todos los días y el código fuente es versátil y flexible, lo que facilita su uso para los desarrolladores (Sharma, 2019).

1.6.2 PHP 8.0.3

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de programación de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Lo que distingue a PHP de algo del lado del cliente como JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP (The PHP Group, 2021).

PHP puede emplearse en todos los sistemas operativos principales, incluyendo Linux, muchas variantes de Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, macOS, RISC OS y probablemente otros más. PHP admite la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros. Con PHP, se tiene la libertad de elegir el sistema operativo y el servidor web. Además, se tiene la posibilidad de utilizar programación por procedimientos o programación orientada a objetos. Una de las características más potentes y destacables de PHP es su soporte para un amplio abanico de bases de datos. Escribir una página web con acceso a una base de datos es increíblemente simple utilizando una de las extensiones específicas de bases de datos (The PHP Group, 2021).

El uso de este lenguaje en su versión 8.0.3 es compatible con el framework Laravel en su versión 9.8.1, también con MySQL como sistema gestor de base de datos y con el servidor web Apache, por lo que el desarrollo del Módulo de Comisiones Disciplinarias del Sistema de Gestión de Trabajo Educativo de la Facultad 1 de la UCI se hace más robusto y disminuye el riesgo de errores en su implementación.

1.6.3 Sistema gestor de base de datos.

MySQL 8.0.3

Es un sistema gestor de base de datos SQL (Structured Query Language) muy rápido, multiproceso, multiusuario y sólido. MySQL 8.0.3 está diseñado para sistemas de producción de carga pesada y de misión crítica, así como para integrarse en software de implementación masiva. MySQL es el sistema de administración de bases de datos SQL de código abierto desarrollado, distribuido y respaldado por Oracle Corporation (Oracle Corporation, 2021).

Está escrito en los lenguajes C y C++ y se destaca por su gran adaptación a varios entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más empleados como PHP, Perl y Java. Este sistema es multiplataforma, es adaptable a distintos sistemas operativos. Permite definir varios usuarios que se autoricen para el manejo de las bases de datos con determinados permisos revocables, garantizando la seguridad de la plataforma. MySQL usa la memoria para búferes internos y cachés que respaldan el número de concurrencias de consultas a bases de datos, y se emplean para las respuestas rápidas del sistema de acuerdo a las condiciones del hardware del sistema (Oracle Corporation, 2021).

En el caso de bases de datos en las que el número de registros no es demasiado grande (hasta 1000 registros), se utiliza una aplicación web y los parámetros técnicos del dispositivo en el que se ejecuta la base de datos son de clase baja o media, MySQL funciona muy bien. Este hecho confirmaría la gran popularidad de esta solución en el caso de aplicaciones de pequeños consumidores. PostgreSQL logró resultados ligeramente peores para una pequeña cantidad de datos (hasta 1000 registros) en la base de datos, lo que le otorga el segundo lugar entre las dos soluciones probadas (Wodyk & Skublewska, 2020).

Se llega a la conclusión de que MySQL es el sistema de gestor de base de datos indicado para proyectos basados en aplicaciones web que requieran alta velocidad y no necesiten de consultas complejas, como es el caso del Módulo de Comisiones Disciplinarias del Sistema de Gestión de Trabajo Educativo de la Facultad 1 de la UCI.

1.6.4 Bootstrap 5.0.2

Bootstrap es un framework que ayuda a la creación de las interfaces de usuario mediante la utilización de componentes como CSS y JavaScript. Esto se puede visualizar en cualquier dispositivo porque se ajusta a las proporciones y resolución de la pantalla en la cual se observa, es más conocido como Diseño Responsivo. Mejora notablemente las proporciones de los CSS de las interfaces de usuario y ayuda al desarrollo web de las aplicaciones; teniendo una interfaz más amigable con el usuario y el incremento en la interacción con la misma desde cualquier lugar (Berrones & Alfonso, 2020; Moreto et al., 2017).

Bootstrap 5 es la versión más reciente de Bootstrap; con nuevos componentes, hojas de estilo más rápidas y más capacidad de respuesta. Es compatible con las últimas versiones estables de todos los principales navegadores y plataformas. Las principales diferencias entre Bootstrap 5 y sus versiones anteriores 4 y 3 es que ha cambiado a JavaScript estándar en lugar de jQuery (Sánchez, 2021).

1.6.5 HTML 5

Es el lenguaje usado por los navegadores para mostrar las páginas web al usuario, publicando la interface más extendida en la red. Permite agrupar textos, sonidos e imágenes y combinarlos a nuestro gusto. Además, nos permite la introducción de referencias a otras páginas por medio de los enlaces hipertexto (Auquilla & José, 2018).

HTML5 es el nuevo estándar para HTML que proporciona a los usuarios y desarrolladores web una funcionalidad mejorada. Las versiones anteriores de HTML, HTML 4.01, que llegaron en 1999, y el desarrollo web han cambiado notablemente desde entonces. Se creó para ofrecer contenido enriquecido sin la necesidad de complementos adicionales ni tecnologías patentadas. El nuevo poder de HTML5 proporciona al usuario todo, desde animación hasta gráficos, música y películas, y también se puede utilizar para crear aplicaciones web complicadas y también es multiplataforma (Ratha et al., 2018).

1.6.6 Servidor web Apache 2.4.46

Apache es un servidor web gratuito y de código abierto basado en Unix desarrollado por Apache Software Foundation. Apache es liviano, con todas las funciones y más poderoso que otros servidores web basados en Unix (Nguyen, 2017). Algunos sondeos realizados demuestran que más del 70% de los sitios web en internet están manejados por Apache, haciéndolo más extensamente usado que todos los otros servidores web juntos. Apache 2.4.46 es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos HTTP, es multiplataforma, puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos, y gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor (Blanco & Benjamin, 2019).

1.6.7 Visual Paradigm 8.0

Visual Paradigm es una herramienta UML CASE. La herramienta está diseñada para una amplia gama de usuarios, incluidos ingenieros de software, analistas de sistemas, analistas de negocios y arquitectos de sistemas, o para cualquier persona que esté interesada en construir de manera confiable sistemas de software a gran escala utilizando un enfoque orientado a objetos. Además, Visual Paradigm admite los últimos estándares de notación UML (Lenguaje Unificado de Modelado). Visual Paradigm 8.0 es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite modelar todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación (Peña et al., 2016; Visual Paradigm, 2021).

1.6.8 Apache JMeter 5.4.1

La aplicación Apache JMeter es un software de código abierto escrito en Java. diseñado para probar el rendimiento tanto en recursos estáticos como dinámicos. Se puede utilizar para simular una carga pesada en un servidor, grupo de servidores, red u objeto para probar su fuerza o para analizar el rendimiento general bajo diferentes tipos de carga (Apache Software Foundation, 2021).

Apache JMeter permite realizar pruebas de carga al sistema que se desarrolla y mide su rendimiento de los recursos web. Toda la información de las pruebas y análisis del rendimiento se muestran a través de las estadísticas del servidor o servicio de destino mediante diagramas y gráfico. Para diseñar un plan de pruebas, JMeter dispone de una interfaz gráfica a modo de diseñador, en la que el probador puede ir agregando componentes de manera visual, y ejecutar los componentes agregados, viendo el resultado. Una vez finalizado el diseño del plan de pruebas, la herramienta permite grabar este como un fichero.jmx. La propia herramienta permite ejecutar un fichero.jmx previamente generado, mediante línea de comandos o la propia interfaz GUI. Para cada petición ejecutada, JMeter recopila ciertos datos. Además, en el fichero.jmx se puede especificar el número de usuarios de cada tipo que ejecuta las peticiones contra la aplicación, es decir, el .jmx simula una o más comunidades de usuarios trabajando contra la aplicación objetivo (Pila & Ernesto, 2016).

1.6.9 Acunetix Web Vulnerability Scanner 11.0

Es una solución completa de pruebas de seguridad de aplicaciones web que se puede utilizar tanto de forma independiente como parte de entornos complejos. Ofrece evaluación y gestión de vulnerabilidades integradas, así como muchas opciones para la integración con herramientas de desarrollo de software líderes en el mercado. Al usar Acunetix, aumenta el nivel de ciberseguridad y elimina muchos riesgos de seguridad a un bajo costo de recursos. Esta herramienta permite fortalecer la seguridad del módulo de gestión que se desarrolla mediante una base de pruebas que incluye inyecciones de SQL, XSS, configuraciones incorrectas, pruebas de penetración manuales, contraseñas débiles y bases de datos expuestas. Los resultados de la base de pruebas que aplica la herramienta muestran las vulnerabilidades que deben ser solucionadas para una mayor seguridad de la aplicación (Acunetix, 2021).

Conclusiones del capítulo

Luego de lo analizado en el presente capítulo se llegan a las siguientes conclusiones:

- El análisis de los sistemas informáticos desarrollados en otros países mostró que no pueden ser utilizados en nuestro país ya que son privativos, los sistemas anteriormente implementados en la universidad no cumplen con todas las funcionalidades, herramientas y tecnologías que se requiere, lo que demuestra la necesidad de desarrollar un módulo que cumpla con las características propias del proceso.
- La selección de la metodología de desarrollo AUP-UCI permitirá lograr mayor organización, planificación y ejecución de las actividades a cumplir.
- La base tecnológica seleccionada favorecerá el desarrollo del Módulo de Comisiones Disciplinarias y su exitosa integración en el Sistema de Trabajo Educativo de la Facultad 1.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN: MÓDULO DE COMISIONES DISCIPLINARIAS DEL SISTEMA DE GESTIÓN DE TRABAJO EDUCATIVO DE LA FACULTAD 1.

En el presente capítulo se describe el negocio, su funcionamiento y características. Se abordan los elementos relacionados con el diseño de la propuesta de solución y el funcionamiento del módulo, especificando sus principales características. Se caracteriza el estilo arquitectónico seleccionado. Se definen los requisitos funcionales y no funcionales que deben estar presentes en la solución mediante la etapa de Ingeniería de requisitos. Se modela el Diagrama de Caso de Uso del Sistema y el Diagrama de Despliegue.

2.1 Descripción del Negocio.

Teniendo en cuenta lo establecido por el Reglamento disciplinario para los estudiantes del Ministerio de Educación Superior, Resolución 240 del 2007. El proceso de comisión disciplinaria comienza cuando la falta disciplinaria se denuncia por cualquier trabajador o estudiante tan pronto se produce o se conoce, mediante correo o personalmente. La denuncia es analizada por el decano, el cual determina si procede o no, cuando la denuncia no procede se archiva la misma. En caso de que proceda, entonces las comisiones designadas por el Decano de la Facultad comienzan a solicitar las declaraciones del denunciante, denunciado y testigos, las opiniones de la Federación Estudiantil Universitaria (FEU), Unión de Jóvenes Comunistas (UJC), el tutor o profesor guía y las evidencias que dan el hecho por probado, las mismas se pueden obtener mediante entrevistas individuales, declaraciones de testigos, registro de conversaciones, elaborando así un expediente por cada estudiante denunciado. Teniendo todas las declaraciones por escrito, los miembros de la comisión se reúnen para analizar y concluir el tipo de falta cometida, se elabora el dictamen con la medida propuesta por la comisión y ésta debe ser analizada por el decano (Resolución No. 240, 2007).

El dictamen contiene los hechos que se consideren probados, señalando las pruebas que, a su juicio, respaldan su criterio; las circunstancias modificativas de la responsabilidad del estudiante, planteando las atenuantes que le reconozca y las agravantes que se le atribuyan; el resultado del análisis realizado en el expediente académico y en los testimonios que expidan las organizaciones políticas, estudiantiles o sindicales si se presentan y la calificación de las faltas cometidas. La comisión podrá proponer, si la aprueba la mayoría de sus integrantes, que se le aplique la suspensión condicional, total o parcial, de la sanción que se imponga, fundamentando en cada caso su propuesta (Resolución No.240, 2007).

El decano analiza el dictamen y determina si procede o no la medida propuesta por la comisión disciplinaria. Si está de acuerdo notifica la medida disciplinaria a los infractores y se les comunica que cuentan con 10 días hábiles para entregar la solicitud de apelación si lo desean, como está establecido en el Reglamento disciplinario. Una vez pasado ese período si no hay solicitud de apelación emitida por parte de los sancionados se comunica al denunciado la medida disciplinaria impuesta y el asesor de trabajo educativo toma las medidas necesarias para que él o los infractores cumplan la sanción impuesta. En caso de emitirse una solicitud de apelación, el expediente es entregado al departamento de Asesoría jurídica, el cual cuenta con 30 días hábiles para proceder, al finalizar esa fecha comunica al decano si la apelación tuvo lugar (Resolución No. 240, 2007).

2.1.1 Actores del negocio.

Un actor del negocio es cualquier individuo, grupo, entidad, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados (Hernández González, 2005). Se han identificado los siguientes actores del negocio:

Table 2. Actores del negocio.

Nombre del actor	Descripción
Denunciante	Es aquel que realiza la denuncia.
Decano	Es aquel que determina si procede o no procede la denuncia, designa los miembros de las comisiones, aprueba las medidas disciplinarias y el dictamen final.
Asesor de Trabajo Educativo	Es aquel que asigna las denuncias a las comisiones disciplinarias.
Presidente de la comisión disciplinaria	Es aquel que realiza la recopilación de las pruebas, genera el expediente del caso, elabora un dictamen y propone las medidas disciplinarias.
Secretario de la comisión disciplinaria	Es aquel que realiza la recopilación de las pruebas, genera el expediente del caso, elabora un dictamen y propone las medidas disciplinarias.
Miembro de la comisión disciplinaria	Es aquel que realiza la recopilación de las pruebas, genera el expediente del caso, elabora un dictamen y propone las medidas disciplinarias.

2.2 Modelo Conceptual.

Weinschenk (2011) señala que el modelo conceptual de un sistema constituye una abstracción externa que describe mediante diagramas y notaciones con distinto grado de formalidad el conocimiento que debe poseer una persona acerca de un sistema, conocimiento que se encuentra almacenado en la Memoria a Largo Plazo. Así pues, desde el punto de vista del usuario el modelo conceptual:

- Representa la información que cualquier usuario debería tener o adquirir sobre el sistema.
- Está formado por un conjunto de elementos (conceptos) y relaciones entre ellos observables desde el exterior.

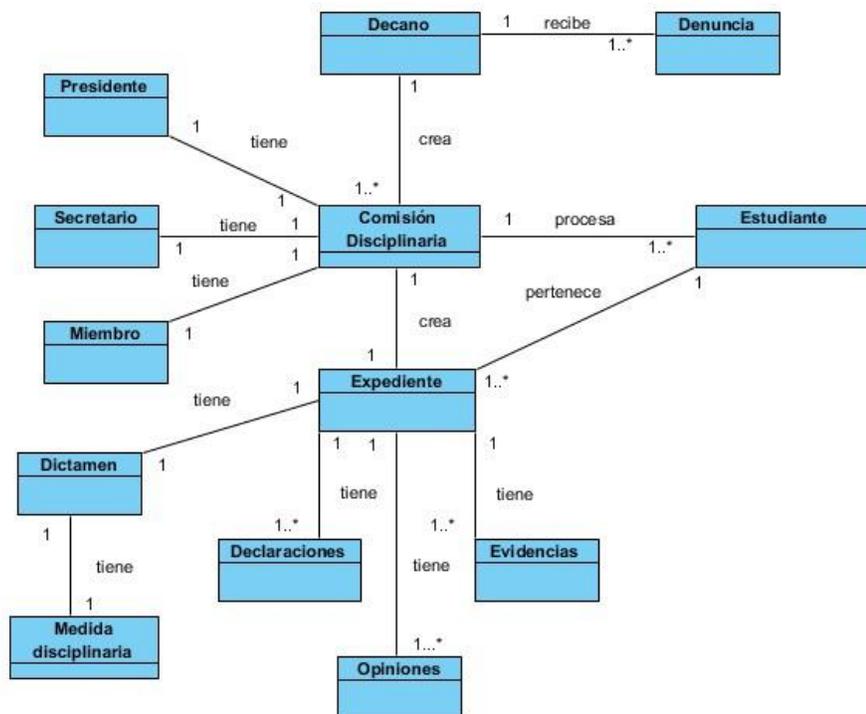


Figure 1. Modelo Conceptual.

El modelo conceptual realizado recoge y describe los conceptos más importantes dentro del contexto del sistema, así como las relaciones entre ellos. Como parte del modelo se aprecian un Decano que recibe una o varias denuncias y crea una o varias comisiones. Las comisiones están compuestas por un presidente, un secretario y un miembro. Una comisión crea uno o varios expedientes y procesa uno o varios estudiantes denunciados. Cada estudiante denunciado tiene un expediente y cada expediente tiene una o varias declaraciones, una o varias opiniones y una o varias evidencias. Cada expediente de cada denunciado tiene un Dictamen final que tiene la medida disciplinaria impuesta.

2.3 Propuesta de solución.

Se propone el desarrollo de un Módulo de Comisiones Disciplinarias del Sistema de Gestión de Trabajo Educativo de la Facultad 1 que permita gestionar denuncias, comisiones disciplinarias, denunciados, expedientes disciplinarios, declaraciones, opiniones, evidencias y dictámenes, así como generar reportes y documentos. Este módulo permitirá controlar la información que genera el proceso de comisiones disciplinarias, además de poder almacenarla, organizarla y procesarla. Se facilitará el trabajo de las personas involucradas en el desarrollo del proceso y se obtendrá un resultado eficiente de éste. Todo esto con el fin de garantizar la seguridad en la información y la gestión y control de la información del proceso de las comisiones disciplinarias en la Facultad 1.

2.4 Especificación de los requisitos del sistema.

El proceso de recogida de información, análisis y documentación sobre los requisitos software, se conoce como ingeniería de requisitos. El objetivo de este tipo de Ingeniería es el de desarrollar y mantener un documento de especificación de requisitos del sistema de forma sofisticada y descriptiva. Los requisitos software son la descripción de las características y las funcionalidades del sistema. Nos comunican las expectativas de los consumidores de productos software (Sommerville, 2002).

Para el éxito en la captura de requisitos se necesitará poner en práctica distintas técnicas de forma que se vayan refinando y completando los requisitos obtenidos (Desarrollo de software: importancia y técnicas de la captura de requisitos, 2019).

Las técnicas empleadas para la obtención de los requisitos que debe cumplir la solución se describen a continuación:

- **Entrevista:** Ésta es la técnica más común utilizada para la obtención de requisitos. Se deben utilizar técnicas de entrevista para construir relaciones sólidas entre los analistas comerciales y las partes interesadas. En esta técnica, el entrevistador dirige la pregunta a las partes interesadas para obtener información. La entrevista uno a uno es la técnica más utilizada (Las 10 técnicas de obtención de requisitos más comunes., 2022). Permite entender con mayor claridad el problema existente por el cual es necesario el desarrollo de dicha solución, siendo entrevistado varios profesores y otros cargos de la facultad encargados de manipular toda esta información referente a las comisiones disciplinarias.
- **Talleres:** Esta técnica se utiliza para generar nuevas ideas y encontrar una solución para un problema específico. Los miembros incluidos para la lluvia de ideas pueden ser expertos en el dominio, expertos en la materia. Múltiples ideas e información le brindan un depósito de conocimientos y puede elegir entre diferentes ideas. Esta sesión generalmente se lleva a cabo alrededor de la mesa de discusión. Todos los participantes deben tener la misma cantidad de tiempo para expresar sus ideas (Las 10 técnicas de obtención de requisitos más comunes., 2022). Se intercambiaron ideas entre miembros del equipo de desarrollo del Sistema de Trabajo Educativo de la Facultad 1 y los encargados de manipular esta información, de manera tal que se obtuvieran los requisitos específicos a desarrollar, para lograr una solución con mayor calidad dada las necesidades existentes.
- **Análisis de Sistema Homólogos:** Esta técnica se utiliza para recopilar información mediante la revisión, examen de los materiales disponibles que describen el entorno empresarial. Este análisis es útil para validar la implementación de las soluciones actuales y también es útil para comprender las necesidades comerciales (Las 10 técnicas de obtención de requisitos más comunes., 2022). Como fuente importante para la obtención de requisitos principales del módulo a desarrollar, se encuentra el análisis a fondo de los sistemas homólogos estudiados y antes descritos.

Una vez definidos los conceptos principales relacionados con el dominio, se muestran a continuación los requerimientos funcionales y no funcionales, de acuerdo con el objetivo planteado al inicio de este trabajo.

2.4.1 Requisitos funcionales

Para el correcto funcionamiento de la solución propuesta se espera que la aplicación permita:

Table 3. Requisitos funcionales.

Requisitos funcionales				
No.	Requisitos	Descripción	Complejidad	Prioridad
RF1.	Autenticar usuario	permite al usuario autenticarse en el sistema	Alta	Alta
RF2.	Listar denuncias	se visualizará la información relevante de las denuncias existentes	Media	Alta
RF3.	Crear denuncia	permite que los usuarios registren las denuncias.	Media	Alta
RF4.	Mostrar denuncia	permite que los usuarios visualicen las denuncias.	Media	Alta
RF5.	Buscar denuncia	permite que los usuarios busquen las denuncias.	Media	Alta
RF6.	Modificar denuncia	permite que los usuarios modifiquen las denuncias.	Media	Alta
RF7.	Eliminar denuncia	permite que los usuarios eliminen las denuncias.	Baja	Alta
RF8.	Listar comisiones disciplinarias	se visualizará la información relevante de las comisiones existentes	Media	Alta
RF9.	Crear comisión disciplinaria	permite que los usuarios creen la comisión disciplinaria.	Media	Alta
RF10.	Modificar comisión disciplinaria	permite que los usuarios modifiquen la comisión disciplinaria.	Media	Alta
RF11.	Eliminar comisión disciplinaria	permite que los usuarios eliminen la comisión disciplinaria.	Baja	Alta
RF12.	Buscar comisión disciplinaria	permite que los usuarios busquen las comisiones disciplinarias	Media	Alta
RF13.	Listar expedientes	se visualizará la información relevante de los expedientes existentes	Media	Media
RF14.	Crear expediente disciplinario	permite que los usuarios creen expedientes disciplinarios.	Media	Media

RF15.	Modificar expediente disciplinario	permite que los usuarios modifiquen los expedientes disciplinarios.	Media	Media
RF16.	Mostrar expediente disciplinario	permite que los usuarios visualicen los expedientes disciplinarios.	Media	Media
RF17.	Eliminar expediente disciplinario	permite que los usuarios eliminen los expedientes disciplinarios.	Baja	Media
RF18.	Buscar expediente disciplinario	permite que los usuarios busquen expedientes disciplinarios	Media	Media
RF19.	Listar declaraciones	se visualizará la información relevante de las declaraciones existentes	Media	Media
RF20.	Registrar declaración	permite que los usuarios registren las declaraciones del denunciado.	Media	Media
RF21.	Modificar declaración	permite que los usuarios modifiquen las declaraciones del denunciado.	Media	Media
RF22.	Eliminar declaración	permite que los usuarios eliminen las declaraciones del denunciado.	Baja	Media
RF23.	Mostrar declaración	permite que los usuarios visualicen las declaraciones del denunciado.	Media	Media
RF24.	Buscar declaración	permite que los usuarios busquen declaraciones	Media	Media
RF25.	Listar opiniones	se visualizará la información relevante de las opiniones existentes	Media	Media
RF26.	Registrar opinión	permite que los usuarios registren las opiniones.	Media	Media
RF27.	Modificar opinión	permite que los usuarios modifiquen las opiniones.	Media	Media
RF28.	Eliminar opinión	permite que los usuarios eliminen las opiniones.	Baja	Media
RF29.	Mostrar opinión	permite que los usuarios visualicen las opiniones.	Media	Media
RF30.	Buscar opinión	permite que los usuarios busquen opiniones	Media	Media
RF31.	Listar evidencias	se visualizará la información relevante de las evidencias existentes	Media	Media

RF32.	Registrar evidencia	permite que los usuarios registren las evidencias.	Media	Media
RF33.	Eliminar evidencia	permite que los usuarios eliminen las evidencias.	Media	Media
RF34.	Mostrar evidencia	permite que los usuarios visualicen las evidencias.	Media	Media
RF35.	Buscar evidencia	permite que los usuarios busquen las evidencias.	Media	Media
RF36.	Listar dictámenes	se visualizará la información relevante de los dictámenes existentes	Media	Media
RF37.	Registrar dictamen	permite que los usuarios registren el dictamen.	Alta	Alta
RF38.	Modificar dictamen	permite que los usuarios modifiquen el dictamen.	Media	Alta
RF39.	Eliminar dictamen	permite que los usuarios eliminen el dictamen.	Baja	Alta
RF40.	Mostrar dictamen	permite que los usuarios busquen dictámenes.	Media	Alta
RF41.	Buscar dictamen	permite que los usuarios busquen las evidencias.	Media	Alta
RF42.	Listar denunciados	se visualizará la información relevante de los denunciados existentes	Media	Alta
RF43.	Crear denunciado	permite que los usuarios registren los denunciados.	Media	Alta
RF44.	Modificar denunciado	permite que los usuarios modifiquen los denunciados	Media	Alta
RF45.	Eliminar denunciado	permite que los usuarios eliminen los denunciados.	Media	Alta
RF46.	Buscar denunciado	permite que los usuarios busquen los denunciados.	Media	Alta
RF47.	Exportar a PDF	permite que los usuarios exporten en formato PDF los documentos.	Media	Media
RF48.	Mostrar reporte	permite que los usuarios puedan visualizar diferentes reportes.	Media	Media

2.4.2 Requisitos no funcionales.

Los requerimientos no funcionales representan características generales y restricciones de la aplicación o módulo que se esté desarrollando. El modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto. En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software determinado. La calidad del producto software se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor. Son precisamente estos requisitos (funcionalidad, rendimiento, seguridad, mantenibilidad, etc.) los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto en características y subcaracterísticas (ISO 25010, 2022). El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad que se muestran en la siguiente figura:



Figure 2. ISO 25010

Los requisitos no funcionales definidos para el módulo son los siguientes:

- **Usabilidad**

Subcaracterística Operabilidad:

RnF 1: El módulo debe presentar una interfaz intuitiva para el usuario. Las interfaces deben tener mensajes, títulos y elementos de ayuda para que el usuario a operar con facilidad con facilidad.

RnF 2: El módulo debe gestionar y requerir información de usuarios para su uso.

Subcaracterística Estética:

RnF 3: El módulo debe poseer un diseño *Responsive* a fin de garantizar la adecuada visualización en múltiples dimensiones de pantallas de computadoras y dispositivos móviles. Podrá ser visualizado en dispositivos desde las resoluciones 768x1024 y 1366x768.

RNF4: La interfaz debe ser de color claro, preferiblemente fondo blanco con un alto contraste entre éste y las opciones del sistema. Para los botones, enlaces, tablas, etc. se deben utilizar clases que brinda Bootstrap 5 para garantizar el alto contraste, para los botones de inserción se utilizará un color verde oscuro, para modificaciones azul claro, para eliminar rojo y para mostrar verde oscuro.

- **Eficiencia del desempeño:**

Subcaracterística comportamiento:

RnF 5: El tiempo de demora de una petición al servidor debe ser menor de 8 a 10 segundos aproximadamente.

Subcaracterística capacidad:

- **Hardware:**

RnF 6: El servidor de base de datos debe poseer una capacidad mínima de 20 GB.

RnF 7: El servidor de aplicaciones web debe poseer una capacidad mínima de 80 GB.

RnF 8: Los servidores web y de base de datos deben poseer como mínimo 2 GB de memoria RAM.

RnF 9: Se requiere una tarjeta de red con velocidad de 100 Mbps.

RnF 10: Se requiere 2 GB de memoria RAM como mínimo.

RnF 11: Se requiere un procesador de 2 GHZ como mínimo.

Subcaracterística utilización de recursos:

- **Software:**

RnF 12: Un navegador Chrome versión 83.0 o superior; o Mozilla Firefox versión 84.0 o superior.

RnF 13: Sistema operativo GNU Linux Ubuntu Server 12.04 o superior o Windows 10.

RnF 14: Servidor Web Apache con PHP 8.0.3

RnF 15: MySQL 8.0 como gestor de bases de datos.

▪ **Portabilidad:**

Subcaracterística adaptabilidad:

RnF 16: La propuesta de solución se debe ejecutar en los navegadores existentes en la Universidad actualmente.

▪ **Seguridad:**

Subcaracterística confidencialidad:

RnF 17: La información manejada por el módulo está protegida con el objetivo de evitar accesos no autorizados.

RnF 18: Ante los posibles errores que puedan ocurrir en el módulo, no se debe mostrar detalles específicos de información que pueda comprometer su seguridad.

Subcaracterística integridad:

RnF 19: Se garantizará la integridad de la información mediante mecanismos de control de acceso, utilizando usuario, contraseña y niveles de accesos para cada usuario de acuerdo a su rol, asegurando que cada uno pueda tener disponible solamente las opciones que se encuentran en correspondencia con su actividad.

▪ **Mantenibilidad:**

Subcaracterística capacidad de ser modificado:

RnF 20: Se debe permitir realizar modificaciones posteriores para mejorar el módulo o en caso de que existan nuevas exigencias por parte del cliente.

RnF 21: El módulo debe estar bien documentado de manera tal que el tiempo de mantenimiento sea lo más pequeño posible.

▪ **Confiabilidad:**

RnF 22: El módulo debe mostrar la información necesaria con el fin de orientar al usuario.

Subcaracterística capacidad de recuperación:

RnF 23: El sistema deberá tardar un máximo de 10 minutos para la recuperación de un fallo de caída total, en el 95% de las ocasiones.

2.5 Modelo de Caso de uso del sistema.

Una vez recopilados los requisitos se crean un conjunto de escenarios que identifiquen una línea de utilización para el módulo que va a ser implementado, estos escenarios son llamados casos de uso y facilitan la descripción de cómo el módulo se usará. En general, un caso de uso es, simplemente, un texto escrito que describe el papel de un actor que interactúa con el acontecer del sistema.

2.5.1 Actores del sistema.

Table 4. Actores del sistema.

Nombre del actor	Descripción
Asesor de Trabajo Educativo	Usuario que tiene el privilegio de gestionar las denuncias y los denunciados.
Decano	Usuario que tiene el privilegio de gestionar las denuncias, los denunciados y las comisiones. Además de gestionar dictamen del expediente.
Comisión Disciplinaria	Usuario que tiene el privilegio de gestionar los expedientes, las declaraciones y las opiniones. Además de gestionar el dictamen.
Usuario de sistema	Usuario que tiene el privilegio de realizar búsquedas y reportes.

2.5.2 Diagrama de Caso de uso del sistema.

Los Diagramas de Casos de Uso son una técnica para capturar requisitos o información de cómo un sistema o negocio trabaja, y están compuesto por los casos de uso, los actores que se pueden definir como algo con comportamiento, como una persona (identificada por un rol), sistema informatizado u organización y las relaciones existentes entre ellos (Pressman, 2010).

En la siguiente figura se muestra el diagrama de Caso de Uso del Sistema:



Figure 3. Diagrama de caso de uso del sistema.

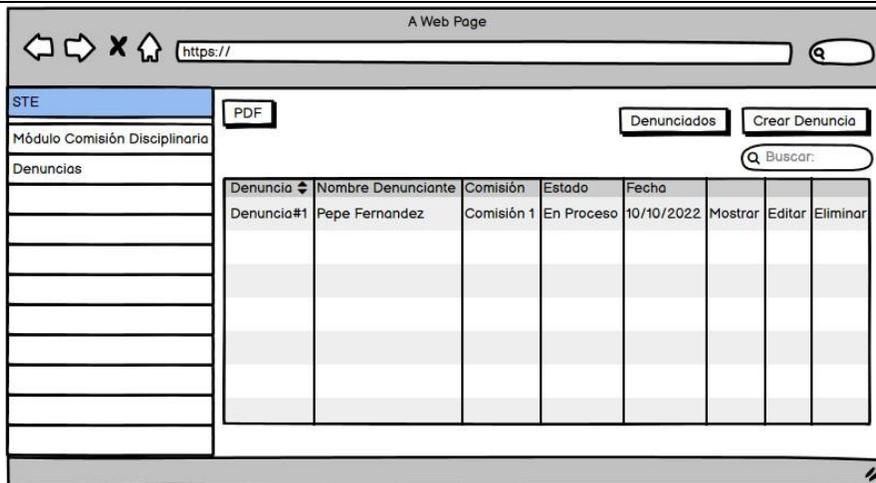
2.5.3 Descripción de casos de uso del sistema.

Para realizar la especificación de requerimientos del sistema se hace uso de las descripciones textuales de los casos de uso, técnica que propone el escenario 2 de la metodología AUP-UCI para encapsular los requerimientos funcionales del sistema.

Table 5. Descripción del caso del uso Gestionar denuncia

Objetivo	El sistema debe ser capaz de crear, eliminar, modificar, mostrar y buscar denuncias
Actores	Administrador
Resumen	El caso de uso se inicia cuando el actor decide crear, eliminar, modificar, buscar y mostrar.
Complejidad	Alta.
Prioridad	Alta.
Precondiciones	Usuario ya autenticado como administrador.

Postcondiciones		El Administrador realizó las operaciones de crear, eliminar, modificar, buscar y mostrar denuncias.
Flujo de eventos Flujo básico < Gestionar Denuncia >		
Actor		Sistema
1.	Selecciona la opción "Comisión disciplinaria" y luego "Denuncias" de la página principal.	
2.		Muestra en pantalla una lista con las opciones de Crear, Modificar, Eliminar y Mostrar datos de las denuncias
3.	Desea Crear, Modificar, Eliminar y Mostrar datos de las denuncias.	
4.		El sistema muestra en pantalla las opciones para la gestión de las denuncias: a) Para insertar datos de nueva denuncia presionar el botón "Crear Denuncia" y se abre una nueva pestaña con el formulario para llenar los datos de la denuncia. b) Para modificar datos de nueva denuncia presionar el botón "Modificar Denuncia" y se abre una nueva pestaña con el formulario para modificar los datos de la denuncia. c) Para ver detalles de los datos de las denuncias presionar el botón "Ver Datos" y se abre una nueva pestaña con todos los datos de la denuncia.
Flujos alternos		
2a. Listado de Denuncias vacío.		
Actor		Sistema
1.		Muestra en pantalla un mensaje "No hay información".
Prototipo de interfaz de usuario (Listar denuncia)		



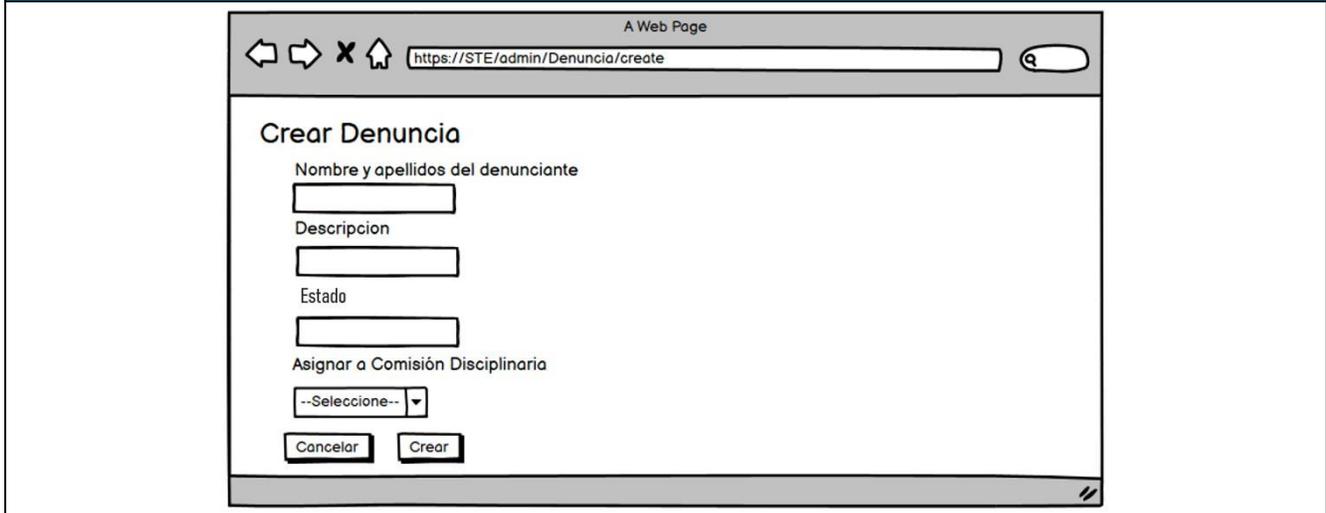
Flujo básico "Crear denuncia"

Actor		Sistema
1.	Selecciona la opción "Comisión disciplinaria" y luego "Denuncias" de la página principal.	
2.		Se visualiza la página de las Denuncias.
3.	Selecciona de la página Denuncias opción "Crear Denuncia".	
4.		Muestra una nueva pestaña con un formulario con los campos: (Nombre y Apellidos del denunciante, Descripción de la denuncia, Asignar a Comisión Disciplinaria).
5.	Introduce los datos (Nombre y Apellidos del denunciante, Descripción de la denuncia, Asignar a Comisión Disciplinaria) para crear la denuncia y presionar el botón "Crear".	
6.		Se valida los datos (Nombre y Apellidos del denunciante, Descripción de la denuncia, Asignar a Comisión Disciplinaria) en caso de estar correctos se guarda la información introducida y se muestra un mensaje "Denuncia Registrada".

Flujo alternativo “Crear Denuncia”

6.1		Si los datos (Nombre y Apellidos del denunciante, Descripción de la denuncia, Asignar a Comisión Disciplinaria) introducidos son incorrectos se muestra un mensaje de “El formato del campo es inválido”
-----	--	--

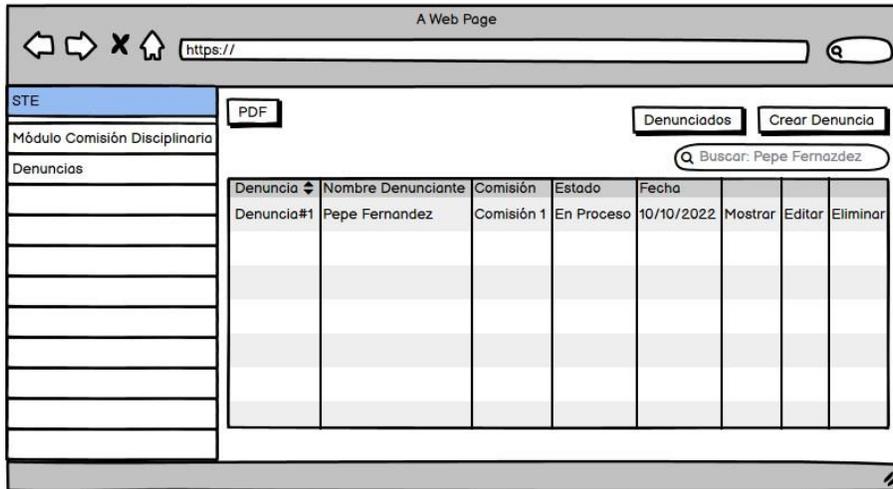
Prototipo de interfaz de usuario (Crear denuncia)



Flujo básico “Buscar denuncia”

Actor		Sistema
1.	Selecciona la opción “Comisión disciplinaria” y luego “Denuncias” de la página principal.	
2.		Se visualiza la página de las Denuncias.
3.	Se llenan los campos en la barra de búsqueda	
		Se visualiza un listado de la denuncia encontrada y las opciones mostrar, modificar y eliminar.

Prototipo de interfaz de usuario (Buscar denuncia)



Flujo básico “Mostrar denuncia”

Actor	Sistema
1.	Selecciona la opción “Mostrar” de la denuncia deseada
2.	Se muestra una nueva pestaña con los datos de la denuncia

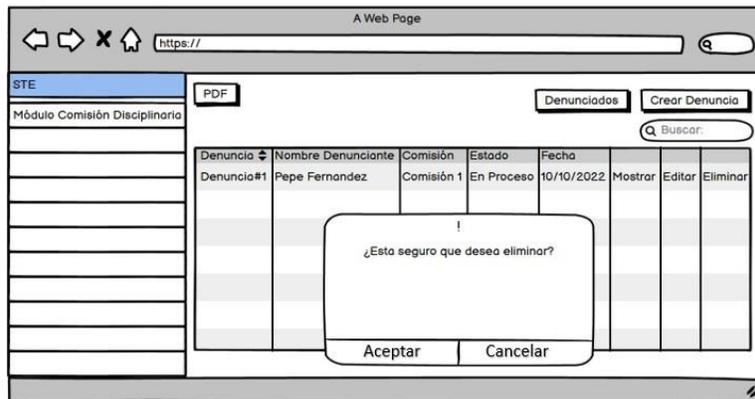
Prototipo de interfaz de usuario (Mostrar denuncia)

Denuncia: 1
Nombre y apellidos del denunciante:
Laura Pérez Fernández
Descripción:
Denuncio a Manuel Sánchez del grupo 1204 por poseer bebidas alcohólicas en el Docente 3.
Comisión: 1
Presidente: Jorge Prieto Sanchez
Secretario: Maisabel Verdano Fuentes
Miembro: Belkis Ricardo Bonilla
Estado:
En Proceso
Fecha de creación:
2022-11-20 22:31:01
Denunciados:
Manuel Sánchez Rodríguez

Flujo básico “Eliminar denuncia”		
Actor		Sistema
1.	Selecciona la opción “Eliminar” de la denuncia deseada.	
2.		Se muestra un mensaje “¿Está seguro que dese eliminar?”
3.	Se presiona el botón “Aceptar”	
4.		Se muestra un mensaje “Se ha eliminado satisfactoriamente”

Flujo alternativo “Eliminar denuncia”		
3.1	Presiona el botón “Cancelar”	
		Se retorna a la página con el listado de las denuncias.

Prototipo de interfaz de usuario (Eliminar denuncia)



Flujo básico “Modificar denuncia”		
Actor		Sistema
1.	Selecciona la opción “Modificar” de la denuncia deseada.	
2.		Muestra una nueva pestaña con un formulario con los campos: (Nombre y Apellidos del denunciante, Descripción de la denuncia Asignar a Comisión Disciplinaria, Fecha de creación).
3.	Introduce los datos (Nombre y Apellidos del denunciante, Descripción de la denuncia, Asignar a Comisión Disciplinaria, Fecha de creación) y presiona el botón “Modificar”.	

4.		Se valida los datos (Nombre y Apellidos del denunciante, Descripción de la denuncia, Asignar a Comisión Disciplinaria, Fecha de creación) en caso de estar correctos se guarda la información introducida y se muestra un mensaje “Su denuncia se ha modificada satisfactoriamente”.
Flujo alternativo “Modificar denuncia”		
4.1		Si los datos (Nombre y Apellidos del denunciante, Descripción de la denuncia, Asignar a Comisión Disciplinaria, Fecha de creación) introducidos son incorrectos se muestra un mensaje de “Datos incorrectos”
4.2	Presiona el botón “Cancelar”	
4.3		Se retorna a la página con el listado de las denuncias

Prototipo de interfaz de usuario (Modificar denuncia)

A Web Page
<https://STE/admin/Denuncia/edit>

Modificar Denuncia

Nombre y apellidos del denunciante

Descripcion

Estado

Asignar a Comisión Disciplinaria

2.6 Descripción de la Arquitectura de Software.

Un estilo de arquitectura es un conjunto de decisiones de diseño arquitectural que son aplicables en un contexto de desarrollo específico, restringen las decisiones de diseño de un sistema a ese contexto y plantean como objetivo ciertas cualidades para el sistema resultante (Cervantes, 2022).

Para el desarrollo del módulo se selecciona como estilo arquitectónico la arquitectura Modelo-Vista-Controlador (MVC). Es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo. El framework Laravel es compatible con patrones de la arquitectura Modelo-Vista-Controlador (Patrón de diseño MVC. ¿Qué es y cómo puedo utilizarlo?, 2020).

Está estructurada por:

- El Modelo que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- La Vista, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.
- El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

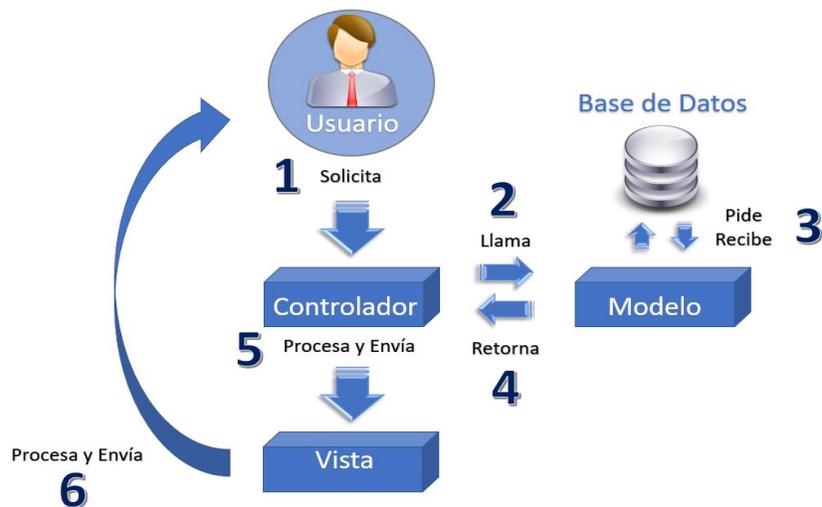


Figure 4. Funcionamiento del estilo arquitectónico Modelo-Vista-Controlador.

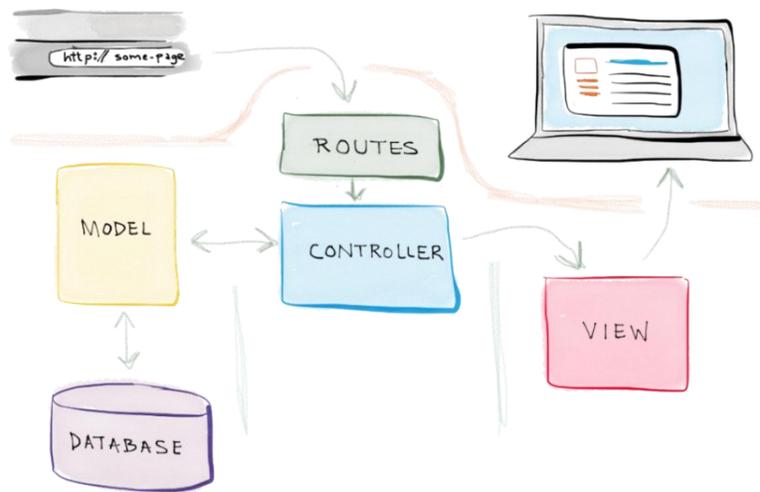


Figure 5. Funcionamiento del estilo M-V-C en Laravel

Descripción del funcionamiento del uso de la arquitectura Modelo-Vista-Controlador en el módulo:

Modelos

Es la capa donde se trabaja con los datos, por tanto, contendrá mecanismos para acceder a la información y también para actualizar su estado. Los datos los tendremos habitualmente en una base de datos, por lo que en los modelos tendremos todas las funciones que accederán a las tablas y harán los correspondientes *selects*, *updates*, *inserts*, entre otros. Los cuales se ven reflejados por ejemplo en el Controlador ExpedienteController.php el cual pide acceso a toda la información referente al Expediente del denunciado a través del modelo Expediente.php.

Vistas

Las vistas, contienen el código de nuestra aplicación que va a producir la visualización de las interfaces de usuario, o sea, el código que permitirá renderizar los estados de nuestra aplicación en HTML. En las vistas están presente códigos HTML y PHP que nos permite mostrar la salida. La vista generalmente trabaja con los datos, sin embargo, no se realiza un acceso directo a éstos. En ellas se muestran toda esa información y estructuradas de siguiente forma por ejemplo de las vistas expediente se distribuye de la siguiente forma: create.blade.php, edit.blade.php, destroy.blade.php, index.blade.php en los cuales cada una se encarga de mostrar una vista diferente con respecto a esa información.

Controladores

Contiene el código necesario para responder a las acciones que se solicitan en la aplicación, es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de la aplicación. Sin embargo, su responsabilidad no es manipular directamente datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo. Reflejándose por ejemplo en el Controlador ExpedienteController.php en el cual están presente los métodos store (), update () que se encargan de manipular toda esta información.

Rutas

Las rutas en Laravel son el componente más importante y poderoso del Framework, se encarga de manejar el flujo de solicitudes HTTP, desde y hacia el cliente; las peticiones realizadas por el navegador son en lo general *get*, *post*, *put*, *delete*, *patch* de HTTP hacia una URL concreta.

2.6 Patrones de diseño.

Dentro de los patrones de diseño se encuentran los Patrones Generales de Software para Asignación de Responsabilidades (GRASP por sus siglas en ingles) y los patrones Banda de los Cuatro (GOF por sus siglas en inglés) (Lago, 2022).

Los patrones de diseño GOF son soluciones habituales a problemas comunes en el diseño de software. Cada patrón es como un plano que se puede personalizar para resolver un problema de diseño particular de tu código (Soto, 2021).

Los patrones GRASP son guías o principios que sirven para asignar correctamente las responsabilidades a las clases (Casas De la Torre, 2017).

2.6.1 Patrones GRASP.

Creador: El patrón creador nos ayuda a identificar quién debe ser el responsable de la creación o instanciación de nuevos objetos o clases. Al escogarlo como creador, da soporte al bajo acoplamiento y favorece la reutilización. Se refleja en las clases controladoras, donde se encuentran las acciones definidas para las operaciones lógicas del negocio en cuestión y se ejecutan cada una de ella (Casas De la Torre, 2017).

En el caso del módulo desarrollado se evidencia el patrón creador en la clase controladora `Comision_DisciplinariaController.php`, se encarga de crear las instancias de los objetos que se manejan favoreciendo la reutilización y el bajo acoplamiento.

```
<?php

namespace App\Http\Controllers\Modulo_ComisionDisciplinaria;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\Modulo_ComisionDisciplinaria\Comision_Disciplinaria;

class Comision_DisciplinariaController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $comision_disciplinaria = Comision_Disciplinaria::all();
        return view('Modulo_ComisionDisciplinaria.Comision_disciplinaria.index')->with('comision_disciplinaria', $comision_disciplinaria);
    }
}
```

Figure 6. Ejemplo de patrón creador en la clase controladora `Comision_DisciplinariaController.php`.

Controlador: Es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control. Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento (Nuñez, 2022).

En el caso del módulo desarrollado se evidencia el patrón en la clase controladora ExpedienteController.php que contiene los métodos necesarios para controlar las declaraciones, evidencias, opiniones y dictamen con respecto a éste.

```
<?php
namespace App\Http\Controllers\Modulo_ComisionDisciplinaria;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\Modulo_ComisionDisciplinaria\Expediente;
use App\Models\Modulo_ComisionDisciplinaria\Denunciado;
use App\Models\Modulo_ComisionDisciplinaria\Declaraciones;
use App\Models\Modulo_ComisionDisciplinaria\Opiniones;
use App\Models\Modulo_ComisionDisciplinaria\Dictamen;
use Illuminate\Support\Facades\DB;
use PDF;

class ExpedienteController extends Controller
{
    /**
```

Figure 7. Ejemplo del patrón controlador en la clase controladora ExpedienteController.php.

Experto: El Patrón Experto consiste en asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad (Vallejo, 2017).

Este patrón se ve reflejado en el módulo, mediante la declaración de los atributos y funciones destinadas al trabajo de las clases. Este indica que la responsabilidad de la creación de las tablas de la base de datos, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. Se pone en práctica en las clases controladoras, modelos y repositorios, que contienen toda la información necesaria para llevar a cabo las responsabilidades que le son asignadas. En el módulo desarrollado se refleja el patrón experto en la clase entidad Denuncia.php, experta en información relacionada con las denuncias.

```
<?php
namespace App\Models\Modulo_ComisionDisciplinaria;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Maatwebsite\Excel\Concerns\ToCollection;

class Denuncia extends Model
{
    use HasFactory;
    protected $table = 'denuncia';
    protected $fillable = ['id', 'Nombre_denunciante', 'descripcion', 'Nombre_denunciado', 'id_comision', 'created_at'];
```

Figure 8. Ejemplo del patrón experto en la clase Denuncia.php.

Alta cohesión: Se evidencia en cada clase del diseño propuesto que realiza una labor única dentro del sistema y que colabore con las otras para llevar a cabo una tarea (Casas De la Torre, 2017).

En el módulo desarrollado se representa en la clase entidad Expediente.php, colabora con la clase denunciado para cumplir su objetivo.

```
<?php
namespace App\Models\Modulo_ComisionDisciplinaria;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Expediente extends Model
{
    use HasFactory;
    protected $table = 'expediente';
    protected $fillable = ['id', 'id_denunciado', 'created_at'];
}
```

Figure 9. Ejemplo del patrón alta cohesión en la clase entidad Expediente.php.

Bajo acoplamiento: Se pone en práctica, puesto que las clases poseen pocas relaciones entre sí, para que, en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión en las demás. Esta característica permitió potenciar la reutilización y disminuyó la dependencia entre las clases (Cardenas Campaña & Barcenas T., 2017).

En el módulo se ve presente en la clase entidad Denuncia.php y ComisionDisciplinaria.php, al mostrar la información de la denuncia que le corresponde a cada comisión, se tiene que recorrer la lista de las denuncias existentes, por lo que se aumenta el acoplamiento y proporciona una baja dependencia entre las dos clases. Lo mismo se evidencia con Expediente.php y Dictamen.php, al mostrar la información del dictamen que le pertenece a cada expediente, se tiene que recorrer la lista de los dictámenes existentes, por lo que se aumenta el acoplamiento entre las dos clases.

2.6.2 Patrones GOF.

El catálogo de patrones más famoso es el contenido en el libro “Design Patterns: Elements of Reusable Object-Oriented Software”, también conocido como: El libro GOF (Gang-Of-Four Book). Según este documento estos patrones se clasifican por su propósito en creacionales, estructurales y de composición, mientras que respecto a su ámbito se clasifican en clases y objetos (Refactoring, 2014).

Según Gracia (2013):

Patrones de creación: Tratan de la inicialización y configuración de clases y objetos.

Patrones estructurales: Tratan de desacoplar interfaz e implementación de clases y objetos.

Patrones de comportamiento: Tratan de las interacciones dinámicas entre sociedades de clases y objetos.

Dentro de los patrones creacionales se encuentra el patrón **Fábrica abstracta**, el cual se encuentra representado en el módulo desarrollado, proporciona una interfaz para la creación de objetos, pero delega la responsabilidad de instanciarlo a sus subclases. Se utiliza en las clases controladoras para la creación de formularios, mediante el método `createForm()`, pasándole como parámetros el formulario a crear y el objeto de la entidad correspondiente (Refactoring, 2014).

Entre los patrones estructurales utilizados en el módulo desarrollado se encuentran Fachada y Decorador:

Fachada: Es una interfaz que actúa como mediadora entre dos capas en la aplicación. Este patrón se empleó en las clases controladoras como intermediarias entre el modelo y las restantes capas (Soto, 2021).

Decorador: Aplicado a la generación de vistas, la solución que ofrece dicho patrón es la de añadir funcionalidad adicional a las plantillas. Por ejemplo, añadir el menú y el pie de página a las plantillas que lo requieran, se trata de decorar las plantillas con elementos adicionales (Soto, 2021) .

Se utilizó la plantilla `adminlte`. Se hereda en todos los archivos `index.blade.php`, `create.blade.php`, `show.blade.php`, `edit.blade.php`.

```
@extends('adminlte::page')
```

Figure 10. Ejemplo del patrón decorador utilizando la plantilla `adminlte`.

En el módulo desarrollado se ve reflejado el patrón de comportamiento **Observador**, este patrón define una dependencia “uno-a-muchos” entre objetos, para que, cuando uno de ellos cambie su estado, todos los que dependan de él sean avisados y puedan actualizarse convenientemente (Parra, 2022).

2.7 Modelo físico de datos.

Un modelo de datos físicos define todos los componentes y servicios lógicos de la base de datos que se requieren para construir una base de datos. Un modelo de datos físicos consta de la estructura de la tabla, los nombres y valores de las columnas, las claves externas y primarias y las relaciones entre las tablas (KeepCoding, 2022).

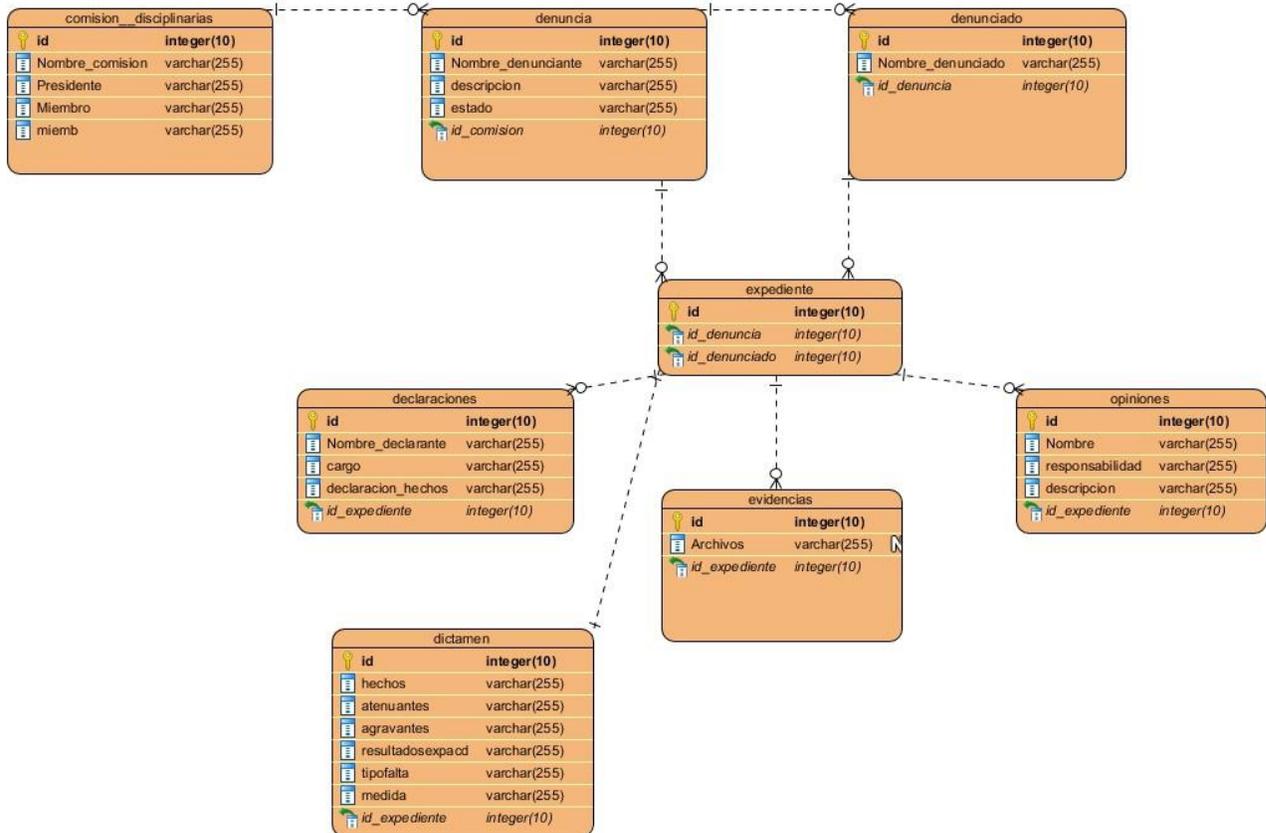


Figure 11. Modelo físico de datos.

2.8 Diagrama de clases de diseño.

Un diagrama de clases del diseño con estereotipos web tiene el mismo objetivo o propósito que un diagrama de clases tradicional, con la particularidad de que se emplea para el modelado de aplicaciones web. En este diagrama se representa la estructura y el comportamiento de cada uno de los objetos del sistema y sus relaciones con los demás objetos, pero no muestra información temporal (Cómo se crean diagramas de clases con notación UML, 2022).

El Módulo de Comisiones Disciplinarias presenta 8 clases encargadas de todo su funcionamiento. Entre estas se encuentra la clase Comisión disciplinaria, encargada del manejo de toda la información de las comisiones correspondiente a la facultad. Se tiene la clase Expediente, encargada de gestionar la información correspondiente a todos los denunciados de la facultad. Además, la clase Denuncia, que gestiona toda la información referente a las denuncias recibidas en la facultad.

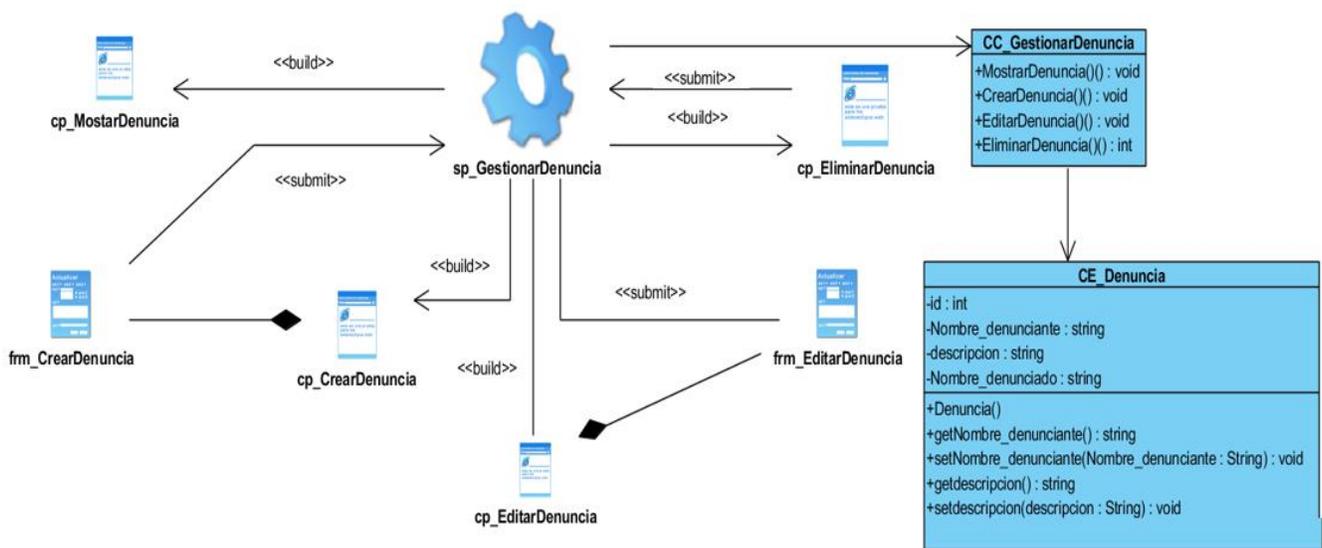


Figure 12. Diagrama de clases con estereotipos web "Gestionar denuncia".

2.9 Diagramas de secuencia.

En contraste con los diagramas de clase y con los diagramas de implementación, que muestran la estructura estática de un componente de software, un diagrama de secuencia se usa para mostrar las comunicaciones dinámicas entre objetos durante la ejecución de una tarea. Este tipo de diagrama muestra el orden temporal en el que los mensajes se envían entre los objetos para lograr dicha tarea (Pressman, 2010).

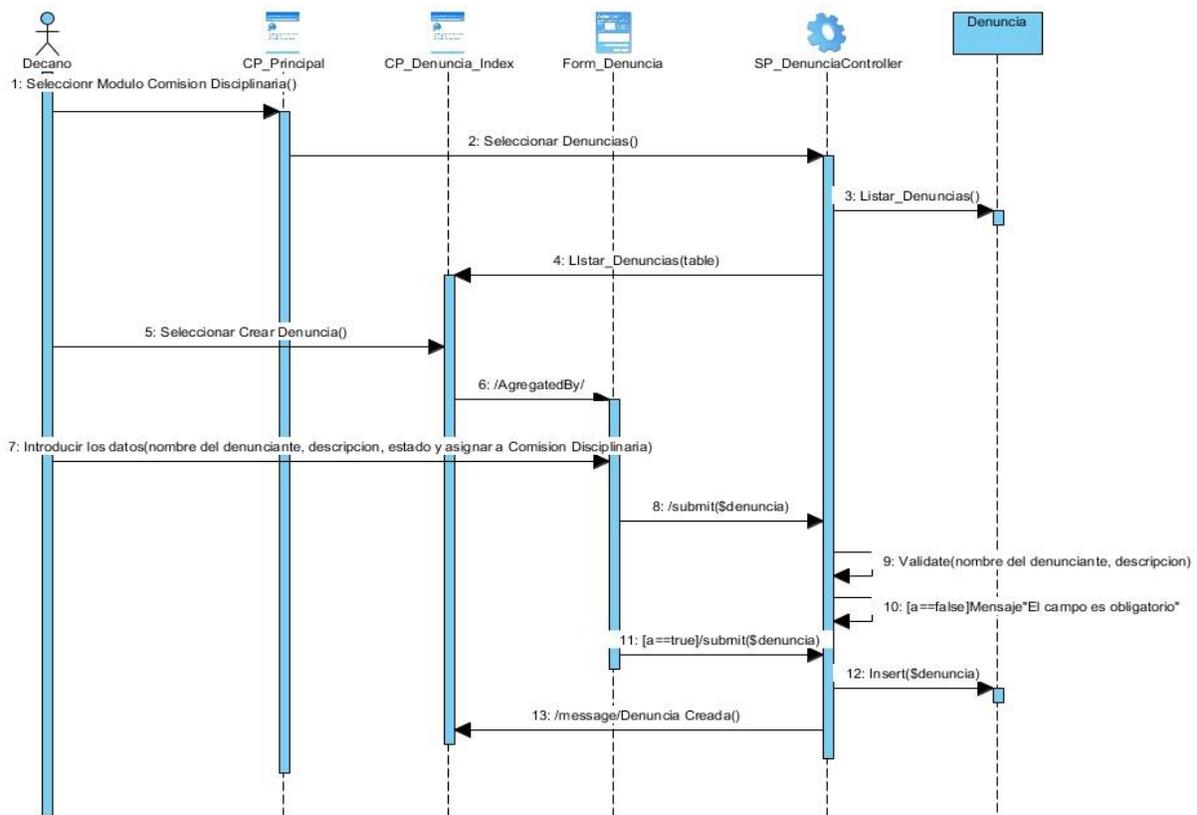


Figure 13. Diagrama de secuencia para el caso de uso Gestionar Denuncia escenario Crear denuncia.

2.10 Diagrama de despliegue.

Los diagramas de despliegue se utilizan como parte del diseño de la arquitectura y se representan en forma de descriptor. De este modo, las funciones principales de un sistema (que con frecuencia se representan como subsistemas) se representan en el contexto del ambiente de computación que las contendrá (Pressman, 2010).



Figure 14. Diagrama de despliegue.

Diagrama de Despliegue: PC cliente mediante el protocolo Https usando puerto 443 se conecta al Servidor web y este mediante protocolo IP con puerto 3306 se conecta al servidor de base de datos.

Dispositivo del cliente: Contiene la estación de trabajo cliente que necesita un navegador Chrome versión 83.0 o superior; o Mozilla Firefox versión 84.0 o superior que conecte con la aplicación web (Apache) situado en el servidor (computadora personal, Tablet, teléfono celular), el cual utiliza para conectarse el protocolo de comunicación HTTPS (Protocolo Seguro de Transferencia de Hipertexto), tiene presente un Sistema operativo GNU Linux Ubuntu Server 12.04 o superior o Windows 10 Home, requieren de una tarjeta de red con velocidad de 100 Mbps, 2 GB de memoria RAM como mínimo y un procesador de 2 GHZ como mínimo.

Servidor Web: Aquí se encuentra el código fuente de la aplicación, brinda a los usuarios las interfaces necesarias durante todos los procesos del negocio. Esta estación se comunica con el servidor de base de datos MySQL donde se almacenan los datos, realizando la comunicación a través del protocolo IP:8080. El servidor de aplicaciones web debe poseer una capacidad mínima de 80 GB como mínimo 1 GB de memoria RAM, requieren de una tarjeta de red con velocidad de 100 Mbps, 2 GB de memoria RAM como mínimo, un procesador de 2 GHZ como mínimo y el Servidor Web Apache con PHP 8.0.

Servidor de BD: Este nodo es el encargado del almacenamiento de los datos. Permite el acceso a la información que tanto el usuario como la aplicación pueden manipular, es el resultado almacenado de las iteraciones del software con una capacidad mínima de 20 GB, requieren de una tarjeta de red con velocidad de 100 Mbps, 2 GB de memoria RAM como mínimo, un procesador de 2 GHZ como mínimo y MySQL 8.0 como gestor de bases de datos.

Conclusiones del capítulo

- Los requisitos funcionales y no funcionales identificados a partir del proceso de captación de requisitos, permitieron desarrollar las distintas funcionalidades que debe cumplir el módulo para solucionar las necesidades detectadas.
- Los diagramas y modelos desarrollados permiten un mayor entendimiento del módulo a implementar mostrando las relaciones existentes entre las distintas clases y las funcionalidades presentes en las mismas.
- El modelo de diseño permite identificar las principales clases del módulo, las relaciones existentes entre ellas y sus métodos asociados, lo que posibilitó un aumento de la reutilización en la implementación.

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN: MÓDULO DE COMISIONES DISCIPLINARIAS DEL SISTEMA DE GESTIÓN DE TRABAJO EDUCATIVO DE LA FACULTAD 1.

En este capítulo se describe el proceso de implementación del Módulo de Comisiones Disciplinarias del Sistema de Gestión de Trabajo Educativo de la Facultad 1. Se generan los artefactos generados por la metodología utilizada en la investigación para el desarrollo y se exponen las pruebas realizadas como demostración del buen funcionamiento del módulo. Se realiza una revisión de los requisitos definidos contra las funcionalidades. Todo ello tiene como objetivo evaluar la calidad del software desarrollado y garantizar que cumpla con las funcionalidades necesarias.

3.1 Diagrama de componentes.

Representa cómo un sistema de software es dividido en componentes, y las dependencias entre estos. Este tipo de diagrama contiene interfaces y sus relaciones, pudiendo contener también paquetes que se utilizan para agrupar elementos del modelo (Diagrama de componentes UML: explicación, creación y ejemplo, 2022).

A continuación, se muestran un diagrama de componentes elaborado y estructurado de acuerdo con el Modelo-Vista-Controlador utilizado por el framework de desarrollo de PHP Laravel.

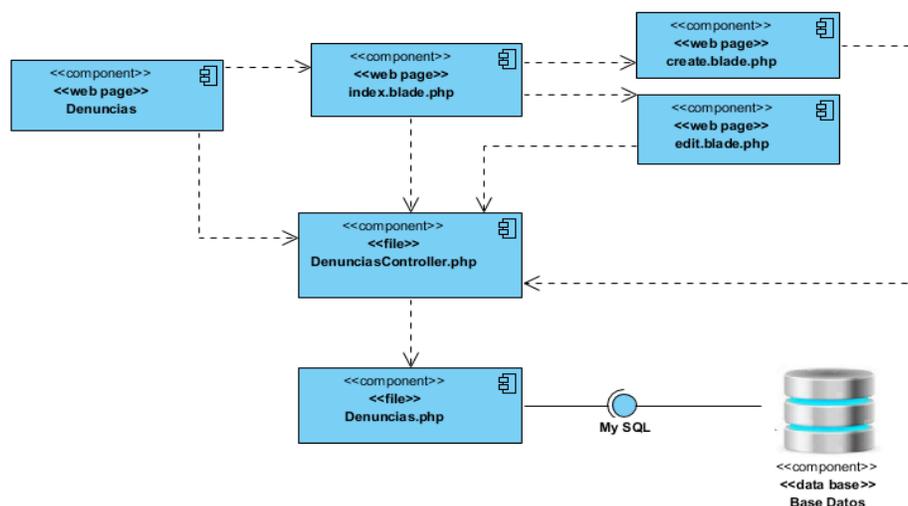


Figure 15. Diagrama de componentes del caso de uso Gestionar Denuncia.

Componentes:

Denuncia.php: Clase entidad que almacena las denuncias en el sistema.

index.blade.php: Se encarga de mostrar el listado de denuncias almacenadas en la base de datos y las opciones necesarias para gestionar las denuncias.

create.blade.php: Se encarga de registrar una nueva denuncia en la base de datos.

edit.blade.php: Se encarga de editar una denuncia en la base de datos.

DenunciaController: Se encarga de realizar todos los métodos relacionados a la gestión de las denuncias en el módulo.

3.2 Estándares de codificación.

Los estándares de codificación son una serie de convenciones que deben seguir los desarrolladores manteniendo buenas prácticas definidas por la ingeniería de software, para obtener un código fácil de comprender y de alta calidad. Posibilita que un equipo de programadores mantenga un código legible sobre el que se efectuarán luego revisiones; con el objetivo de regular la calidad de la implementación estableciendo un estándar de desarrollo común por el cual se rige la programación del sistema. El objetivo de los estándares de codificación de software es inculcar prácticas de programación probadas que conduzcan a un código seguro, confiable, comprobable y mantenible. Por lo general, esto significa evitar las prácticas de codificación inseguras conocidas o el código que puede causar un comportamiento impredecible (Estándares de codificación de software y pautas de programación, 2020).

- Cabecera del archivo: Los archivos .php inician con una cabecera específica que indique información relevante del mismo, tales como: nombre del autor, paquete al que pertenece, acceso al archivo, etc. Cada programador decide si debe agregar más datos.
- Cada archivo especificará el paquete que lo contiene y el uso de otras clases.

En la siguiente imagen se muestra que en el código del módulo desarrollado cada archivo especifica el paquete que lo contiene:

```
<?php
namespace App\Http\Controllers\Modulo_ComisionDisciplinaria;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\Modulo_ComisionDisciplinaria\Expediente;
use App\Models\Modulo_ComisionDisciplinaria\Denunciado;
use App\Models\Modulo_ComisionDisciplinaria\Declaraciones;
use App\Models\Modulo_ComisionDisciplinaria\Opiniones;
use App\Models\Modulo_ComisionDisciplinaria\Dictamen;
use Illuminate\Support\Facades\DB;
use PDF;
```

Figure 16. Ejemplo en el código

donde cada archivo especifica el paquete que lo contiene y el uso de otras clases.

- Etiquetas de apertura y cierre de PHP: Cuando se utiliza el lenguaje PHP, siempre se deben utilizar las etiquetas: <?PHP y?>, y en ningún caso la versión corta <? Y?>

```
<?php
namespace App\Http\Controllers\Modulo_ComisionDisciplinaria;
```

Figure 17. Ejemplo de etiqueta de apertura en el código.

Con respecto a las estructuras de control: Debe haber un espacio entre el comando que define la estructura (if, for, while, etc.) y el paréntesis de apertura. Se recomienda usar siempre las llaves {} aún en los casos en que no sea obligatorio su uso. La llave de apertura se situará en la misma línea que la definición de la estructura, separada por un espacio. Las estructuras else y elseif se escribirán en la línea siguiente al cierre de la sentencia anterior.

- Clases: Las clases serán colocadas en un archivo.php aparte, donde solo se colocará el código de la clase. El nombre del archivo será el mismo de la clase y siempre empezará en mayúscula.

```
class DeclaracionesController extends Controller
{
```

Figure 18. Ejemplo de nombre de una clase en el código.

- Uso de comillas: Se pueden usar tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres. Las comillas dobles son necesarias si se desean incluir variables dentro de las cadenas de texto. Por ejemplo, "<h1>\${titulo}</h1>". También se recomienda el uso de comillas dobles cuando el texto puede incluir alguna comilla simple.
- Nombres de variables: Deben estar en minúscula y ser descriptivos y concisos.

```
$expediente = Expediente::find($id);
$denunciado=Denunciado::all();
$declaraciones=Declaraciones::all()->
```

Figure 19. Ejemplo de nombres de variables en el código.

3.3 Validación de la solución.

La prueba de software es un elemento de un tema más amplio que usualmente se conoce como verificación y validación (V&V). La verificación se refiere al conjunto de tareas que garantizan que el software implementa correctamente una función específica. La validación es un conjunto diferente de tareas que aseguran que el software que se construye sigue los requerimientos del cliente (Pressman, 2010).

Se propone una estrategia de pruebas, para validar el módulo desarrollado, que contempla la realización de pruebas funcionales (ver Tabla 6), en función de garantizar y validar su calidad.

Table 6. Validación de la solución.

Tipo de prueba	Método (técnica) de prueba	Validación
Funcional	Casos de prueba (Caja Negra)	Valida las funcionalidades diseñadas para el módulo
Seguridad	Software Acunetix	Valida la confidencialidad, integridad y disponibilidad de los datos en el módulo.
Carga y estrés	Software Apache Jmeter	Valida el comportamiento del módulo con distintos niveles de usuarios concurrentes y el consumo excesivo de sus recursos
Integración	Integración de módulos	Consiste en la comprobación de que los elementos del software que interactúan entre sí, funcionan de manera correcta.

3.3.1 Pruebas funcionales

Los requerimientos de software se revisan usando técnicas de diseño de casos de prueba de “caja negra”. La prueba de caja negra se lleva a cabo en la interfaz del software. Una prueba de caja negra examina algunos aspectos fundamentales de un sistema con poca preocupación por la estructura lógica interna del software (Pressman, 2010).

Con el objetivo de evaluar la calidad del producto desarrollado y garantizar que el módulo diseñado e implementado cumpla con las funcionalidades, se elaboraron casos de prueba basados en requisitos para realizar pruebas de caja negra. A continuación, se muestra uno de los casos de prueba mencionados. En las celdas de la tabla del caso de prueba (ver tabla 7) se pueden encontrar los valores V, para datos válidos, I, para datos inválidos y N/A para datos a los que no es necesario proporcionarles un valor. Para poder realizar este tipo de pruebas de caja negra se debe ejecutar cada uno de los casos de pruebas diseñados en tantas iteraciones como sea posible hasta que no se detecten no conformidades, al finalizar cada iteración se debe corregir las no conformidades detectadas hasta obtener un módulo funcional.

Table 7. Descripción de las variables del caso de prueba 1 basado en el CU- Crear denuncia.

Variable	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre y Apellidos del denunciante	Campo de texto	NO	Se deben escribir palabras comenzando por mayúscula y sin caracteres extraños.
2	Descripción de la denuncia	Campo de texto	NO	Se debe escribir palabras, sin caracteres extraños
3	Asignar a comisión disciplinaria	Campo de selección	Si	Se debe seleccionar el valor que se requiere

A continuación, se muestra los distintos escenarios del caso de prueba “Crear denuncia”.

Table 8. Caso de prueba del RF1_Crear denuncia.

Escenario	Descripción	1	2	3	Respuesta del Sistema	Flujo Central
EC 1.1 Crear una denuncia al sistema de forma correcta	Interfaz con el formulario para llenar los datos de la denuncia, si todos son correctos, se crea la denuncia.	V	V	V	Agrega la denuncia y muestra un mensaje de notificación.	1. Selecciona de la página principal la opción “Denuncias”. 2. Llenar los campos correspondientes en el formulario y seleccionar la opción “Crear”.
		Isabel González Rodríguez	Música alta por la madrugada en la beca.	Comisión 1		
EC 1.2 Crear denuncia con campos vacíos	Interfaz con el formulario para llenar los datos de la denuncia, si existe algún campo vacío, se muestra un mensaje pidiendo llenar el campo.	I	V	N/A	Comprueba si los campos están vacíos, si lo están, muestra un mensaje que solicita llenarlos.	
EC 1.3 Crear denuncia con caracteres extraños	Interfaz con el formulario para llenar los datos de la denuncia, si existe algún campo con caracteres especiales se muestra un mensaje introducir datos válidos.	I	I	V	Comprueba si los campos tienen caracteres, si los tienen, muestra un mensaje que debajo de cada campo señalando que el campo tiene un error y por lo tanto debe ser llenado correctamente.	
		Aleja/*#\$ Pérez% &/ Sánchez	/\$5Música alta por la madrugada en la beca /8)	Comisión 1		

A continuación, se muestran los resultados obtenidos en 3 iteraciones de pruebas de caja negra realizadas al módulo, la figura 20 muestra la información del total de no conformidades encontradas y las que se fueron resolviendo por cada iteración. Con un total de 48 requisitos se detectaron 8 no conformidades en la primera iteración resolviéndose satisfactoriamente 4, en la segunda iteración quedaron 4 no conformidades resolviéndose satisfactoriamente 3 y en la tercera iteración quedó 1 no conformidad resolviéndose satisfactoriamente.

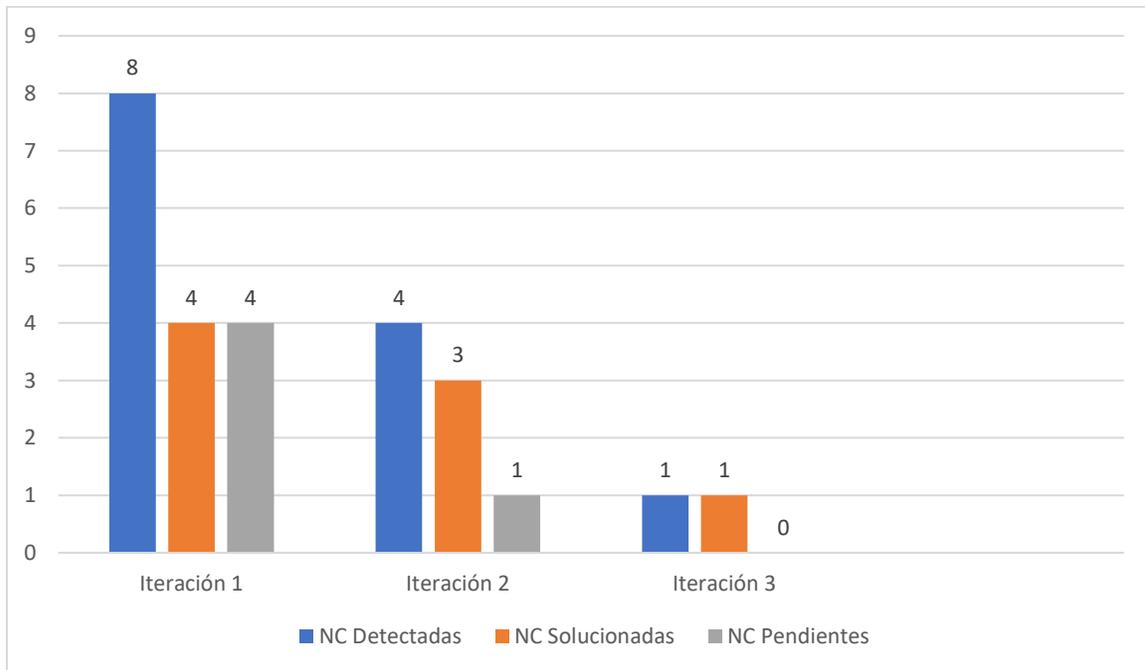


Figure 20. Comportamiento de las no conformidades por iteraciones en la prueba de caja negra.

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

- Errores de estructuración de los contenidos mostrados en las vistas.
- Errores ortográficos.
- Errores de funcionamiento de algunas rutas.
- Se mostraban algunos mensajes en inglés.

Descripción de las no conformidades:

De las 8 no conformidades que se encontraron, en una primera iteración se solucionaron 4, entre estas se encuentran:

- Las vistas de los formularios de las denuncias y las comisiones presentaban los botones cancelar y guardar mal ubicados, se encontraban por encima de los últimos campos del formulario, se solucionó cambiando el orden de los botones en las vistas index.php de denuncias y comisiones, se ubicaron los botones al final del formulario y se solucionó la no conformidad.
- El módulo permitía registrar los datos de los denunciados sin que se le asignara una denuncia, el mismo fue resuelto al validar el campo asignar denuncia como campo requerido.
- Al crear una declaración el campo cargo se podía quedar vacío, el mismo fue solucionado al validar el campo cargo como campo requerido.

En una segunda iteración de 4 no conformidades pendientes se solucionaron 3, entre estas:

- No se generaba el PDF con los datos de las comisiones, se solucionó revisando el método downloadPDF implementado en el Controlador Comision_DisciplinariaController.php donde se detectó un error en el nombre de la variable del return que no coincidía con la variable \$pdfcomisiones declarada en el inicio, se modificó el nombre y se solucionó la no conformidad.
- Los mensajes de validación campos requeridos en crear expedientes y crear denuncias se mostraban en inglés, se solucionó creando en los métodos store de los controladores correspondientes a Expedientes y Denuncias cada uno los mensajes en español utilizando la variable \$messages.

En una última iteración se solucionó la no conformidad pendiente de la anterior iteración:

- Se intentaba acceder a los expedientes, pero el botón no accedía a la página correspondiente, se solucionó modificando la URL que se encontraba mal configurada en la redirección del controlador responsable ExpedienteController.php.

3.3.2 Pruebas de seguridad.

Las pruebas de seguridad se realizan con la intención de explotar las vulnerabilidades en la webapp o dentro de su entorno (Pressman, 2010).

Para la realización de las pruebas de seguridad se utilizó la herramienta Acunetix Web Vulnerability Scanner, que divide la criticidad de los errores en 4 niveles (Alto, Medio, Bajo e Informativo). Durante la primera iteración, se detectaron 18 no conformidades: 15 de nivel medio, 1 de nivel bajo y 2 de nivel informativo, los cuales fueron comprobados manualmente y solucionados, logrando así que en la segunda iteración de la aplicación no se obtuviera ninguna deficiencia en la seguridad del módulo. A continuación, en la siguiente figura se pueden observar todos los resultados antes descritos:

Alerts distribution

Total alerts found	18
High	0
Medium	15
Low	1
Informational	2

Figure 21. Resultados prueba de seguridad.

Luego de analizar los resultados se puede concluir que la mayoría de los problemas son debido a mensajes de errores y de redirección propios del Framework Laravel utilizado en la implementación, en la segunda iteración no se obtuvo ninguna deficiencia en la seguridad del sistema, permitiendo esto poder utilizar la herramienta desarrollada como propuesta de solución de manera segura. A continuación, en la siguiente figura se pueden observar todos los resultados antes descritos:

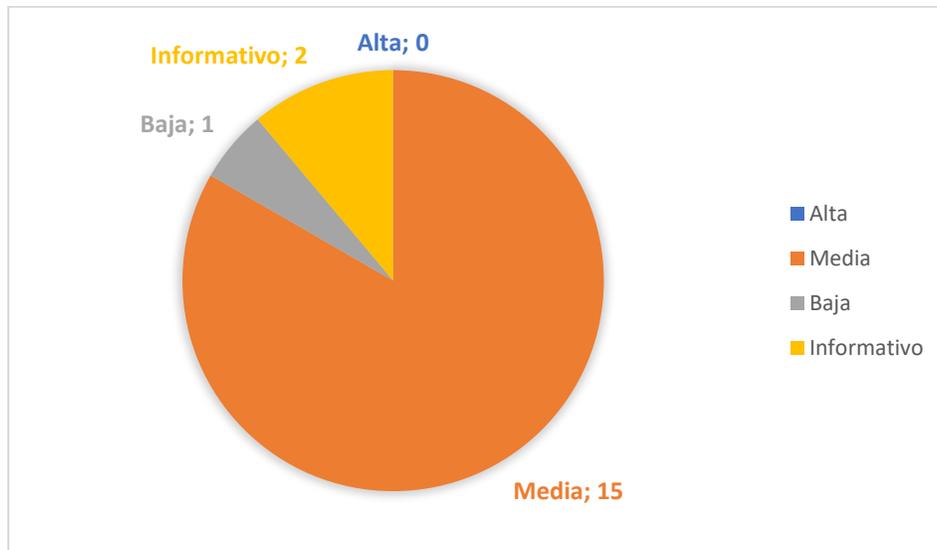


Figure 22. Resultados pruebas de seguridad.

3.3.3 Pruebas de carga y estrés.

La intención de la prueba de carga es determinar cómo responderán las webapps y su entorno del lado servidor a varias condiciones de carga. Conforme avanzan las pruebas, las permutas de las siguientes variables definen un conjunto de condiciones de prueba:

N, número de usuarios concurrentes.

T, número de transacciones en línea por unidad de tiempo.

D, carga de datos procesados por el servidor en cada transacción (Pressman, 2010).

Para la realización de las pruebas de carga y estrés, se utilizó la aplicación Apache JMeter, que admite una serie de parámetros, arrojando un número importante de resultados y distintas formas de visualizar los mismos. Teniendo en cuenta que se espera que el módulo tenga una mayor concurrencia de usuarios, se estableció una muestra de 50 y 100 usuarios concurrentes y las siguientes condiciones:

Hardware de prueba (PC servidor):

- Sistema operativo GNU Linux Ubuntu Server 12.04 o superior o Windows 10.
- Microprocesador: 2 GHZ
- Memoria RAM: 2.00 GB
- Disco Duro: 1000 GB

Hardware de prueba (PC cliente):

- Sistema Operativo: Windows 10 Pro
- Microprocesador: Intel® Celeron® CPU N3060 @ 1.60GHz
- Memoria RAM: 4.00 GB
- Disco Duro: 1000 GB
- Tipo de Sistema: Sistema operativo de 64 bits, procesador x64

Software instalado en ambas PC:

- Tipo de servidor web: Apache con PHP 8.0.3.
- Plataforma: Sistema operativo GNU Linux Ubuntu Server 12.04 o superior o Windows 10.
- Servidor de BD: MySQL 8.0.

Luego de definido el hardware se configuran los parámetros del Apache. Parámetros que componen las pruebas de rendimiento:

Muestra: Cantidad de peticiones realizadas para cada URL.

Media: Tiempo promedio en milisegundos en el que se obtienen los resultados.

Mediana: Tiempo en milisegundos en el que se obtuvo el resultado que ocupa la posición central.

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

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

Línea 90 %: Máximo tiempo utilizado por el 90 % de la muestra, al resto de la misma le llevo más tiempo.

% Error: Por ciento de error de las páginas que no se llegaron a cargar de manera satisfactoria.

Rendimiento (Rend): El rendimiento se mide en cantidad de solicitudes por segundo.

Kb/s: Velocidad de carga de las páginas.

Como se muestra en la siguiente tabla, se simularon las peticiones realizadas al módulo por un total de 50, 100 usuarios simultáneamente en cada caso. Se obtuvieron los siguientes resultados:

Table 9. Resultado de las pruebas de Carga y Estrés.

Usuarios	Muestras	Media	Mediana	Mín.	Máx.	Línea 90%	%Error	Rend.	Kb/s
50	220	1430	1228	60	4120	1690	0.00%	6.9	20.8
100	440	1613	1030	84	4602	2461	0.00%	40.8	180.7

Análisis de los resultados de las pruebas de rendimiento:

El informe de Apache JMeter arrojó que el Módulo de Comisiones Disciplinarias es capaz de responder 220 peticiones de 50 usuarios en un tiempo promedio de 1430 milisegundos con un error de 0.0%. y 440 peticiones de 100 usuarios en un tiempo promedio de 1613 milisegundos con un error de 0.0%. A partir de este resultado, puede afirmarse que el Módulo responde de manera satisfactoria.

3.3.4 Pruebas de integración.

La prueba de integración es uno de los tipos de prueba funcional más común y se realiza de forma automatizada. Se realizan para probar componentes individuales con el objetivo de verificar cómo los módulos, que trabajan de forma individual, funcionan cuando estén integrados (Trycore, 2019).

GitHub es una forja para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.

Procedimiento para la integración:

- Se emplean una serie de comandos en la terminal de Visual Studio Code para clonar el proyecto ubicado en GitHub.
- Se trabaja el módulo en una nueva rama separada de trabajo en el Visual Studio Code.

- Autenticarse en Git app de escritorio.
- Autenticarse en GitHub.
- Dar permiso en Visual Studio Code para la modificación de una rama.
- En Visual Studio Code se presiona la opción commit.

En una primera iteración se encontraron 2 no conformidades y fueron solucionadas, al realizar una segunda iteración no se encontraron no conformidades, llegándose a la conclusión que existe una correcta integración entre los componentes internos del sistema. En el presente proceso de ejecución de las pruebas de integración se pudo comprobar de manera conjunta cada uno de los componentes del Sistema de Trabajo Educativo de la Facultad 1, lo que permitió realizar varias interacciones del sistema en total integración entre los módulos y comprobando el correcto funcionamiento de los mismos.

En una primera iteración las no conformidades estaban relacionadas con errores de migración de tablas, relacionadas con otros módulos, existían tablas de uso común que fueron creadas independientes. La tabla de base de datos que necesitaba utilizar el Módulo de Comisiones Disciplinarias para su correcto funcionamiento es la tabla Estudiantes del Módulo de Perfil de Usuario, esta tabla estaba repetida porque estos dos módulos la tenían creada, se solucionó borrando la tabla del Módulo de Comisiones Disciplinarias y utilizando la tabla correcta perteneciente al Módulo de Perfil de Usuario que brinda todos los estudiantes de la Facultad 1.

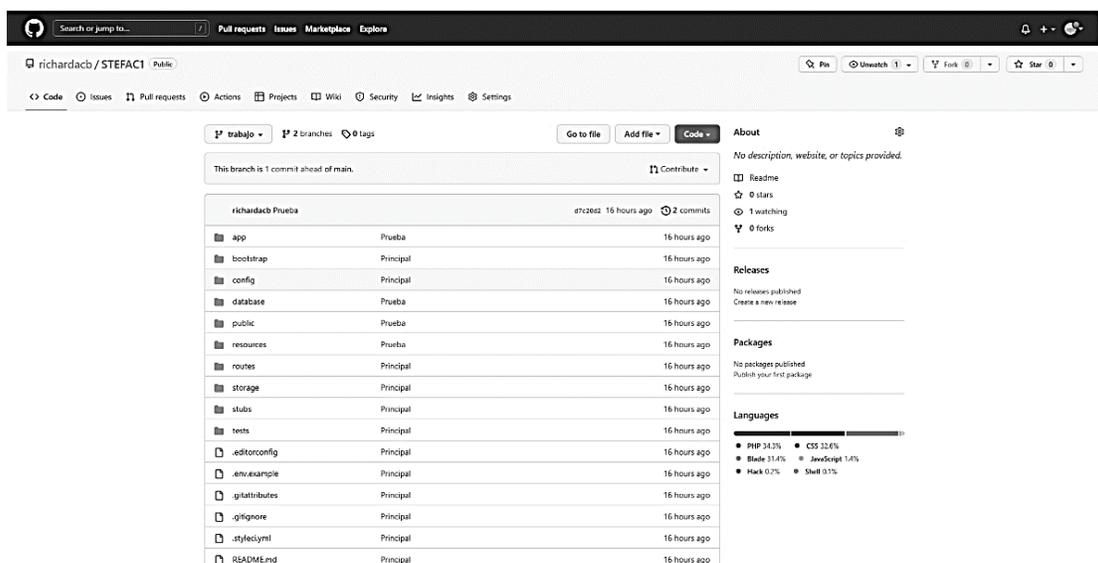


Figure 23. GitHub.

3.4 Validación de los resultados de la investigación aplicando la técnica ladov.

Para la aplicación del Test de ladov se seleccionaron los 5 profesores de la Facultad 1. Los criterios que se utilizaron en esta investigación para el estudio de la satisfacción se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario de siete preguntas y cuya relación el encuestado desconoce. Las tres primeras preguntas se relacionan a través de lo que se denomina “Cuadro Lógico de ladov” se pueden ver en los Anexos. Las respuestas interrelacionadas de las preguntas cerradas (preguntas 1, 2 y 3) permiten determinar el grado de satisfacción de cada individuo de acuerdo con la escala siguiente:

1. Clara satisfacción
2. Más satisfecho que insatisfecho
3. No definido
4. Más insatisfecho que satisfecho
5. Clara insatisfacción
6. Contradictorio

El índice de satisfacción grupal (ISG) se obtiene utilizando la fórmula:

$$ISG = \frac{A (+1) + B (0.5) + C (0) + D (-0.5) + E (-1)}{N}$$

En esta fórmula A, B, C, D, E, representan el número de sujetos con su índice individual y N representa el número total de profesores encuestados. El índice de satisfacción grupal puede oscilar entre (+1) y (-1), esto permite reconocer las siguientes categorías grupales:

Table 10. Escala numérica para el ISG.

Escala entre	Nivel de satisfacción
-1 y -0.5	Insatisfacción
-0.49 y 0.49	No definidos y contradictorios
0.5 y 1	Satisfacción

Aplicando el cuadro lógico de ladov para cada uno de los encuestados se obtuvieron los resultados siguientes:

Table 11. Resultados de aplicar el test de ladov.

Resultado	Cantidad	%
Máximo de satisfacción	3	60%
Más satisfecho que insatisfecho	2	40%
No definida	0	0%
Más insatisfecho que satisfecho	0	0%
Clara insatisfacción	0	0%

Por tal motivo:

$$3 (+1) + 2 (+0.5) + 0 (0) + D (-0.5) + E (-1)$$

$$ISG = \frac{\quad}{5} = 0.8$$

El índice de satisfacción grupal alcanzado (0.8) reflejando la clara satisfacción por la propuesta y reconocimiento de su utilidad en el mejoramiento de lo que plantea el marco teórico de esta investigación.

En la figura 24 se representan los porcentos de satisfacción obtenidos luego de calculado el ISG, no encontrándose los tres últimos colores por estar en 0%.



Figure 24. Resultados de satisfacción (ladov).

Conclusiones del capítulo

En este capítulo se han abordado los elementos de la implementación del módulo, así como las pruebas sugeridas a realizar; lo cual permite arribar a las siguientes conclusiones:

- La elaboración del diagrama de componentes, permite una mejor comprensión de la estructura de los componentes del módulo implementado.
- El estándar de codificación empleado facilitó la comprensión del código fuente, permitiendo desarrollar el módulo en un menor tiempo.
- El proceso de validación de la propuesta de solución, a través de la estrategia de pruebas especificada, arroja como resultado que el módulo implementado responderá a los requerimientos definidos.
- La realización de las pruebas de funcionalidad, seguridad y carga y estrés, evidenciaron la calidad y el correcto funcionamiento del módulo.

CONCLUSIONES FINALES

Luego del estudio realizado se puede concluir que:

1. A partir del estudio realizado de los referentes teóricos relacionados con el proceso comisiones disciplinarias, se determinó que existen una serie de sistemas que se utilizan para contribuir a la gestión de la información referente al proceso de comisiones disciplinarias o afines a este, y se seleccionaron las funcionalidades para la propuesta de solución de acuerdo a las necesidades existentes.
2. El uso de la metodología AUP-UCI, orientada a equipos de desarrollo pequeños, permitió la obtención de un producto que responde a los objetivos de esta investigación.
3. Las tecnologías y herramientas seleccionadas resultaron ser las adecuadas para el desarrollo del módulo, facilitando el proceso de desarrollo y proporcionando mayor mantenibilidad y flexibilidad.
4. Con la definición de la arquitectura, los patrones de diseño y los estándares de codificación se logró el empleo de buenas prácticas en el desarrollo del módulo.
5. Con el desarrollo del módulo diseñado se cumplió con el objetivo planteado en la investigación, surgiendo así una herramienta para la gestión y control de la información referente al proceso de comisiones disciplinarias en la Facultad 1 de la UCI.
6. La utilización de la estrategia de pruebas garantizó la identificación temprana de las deficiencias en el módulo desarrollado y su posterior corrección.

RECOMENDACIONES

Se realizan las siguientes recomendaciones:

1. Agregar nuevas funcionalidades acordes a nuevos requisitos que pudiesen surgir por alguna causa o para aumentar el rendimiento del módulo.
2. Realizar un estudio más profundo de este módulo en vista al perfeccionamiento en versiones futuras.

REFERENCIAS BIBLIOGRÁFICAS

- Alarcón Ortiz, R. A. (2019). *Revista Estudios del Desarrollo Social: Cuba y América Latina*. 7(3).
- Almaguer Millan, N. (2021). *Sistema para la gestión y control de la información generada por las Comisiones Disciplinarias de la Facultad 1 (ControlD)*.
- Ávila, H. F., González, M. M., & Licea, S. M. (2020). LA ENTREVISTA Y LA ENCUESTA: ¿MÉTODOS O TÉCNICAS DE INDAGACIÓN EMPÍRICA? *Didasc@lia: didáctica y educación* ISSN 2224-2643, 11(3), Art. 3.
- Barreda, D. H., & Medina, H. S. (2019). La modelación como método del conocimiento científico en las ciencias sociales. El caso del modelo cubano de televisión local. *Revista Latinoamericana de Metodología de las Ciencias Sociales*, 9(1). <https://www.relmecs.fahce.unlp.edu.ar/article/download/Relmecse053/10738?inline=1>
- Cardenas Campaña, J. A., & Barcenas T., R. J. (2017). *Bajo acoplamiento · Atributos de calidad y patrones de diseño*. <https://fjimenezg.gitbooks.io/atributos-de-calidad-y-patrones-de-diseno/content/bajo-acoplamiento.html>
- Casas De la Torre, F. A. (2017). *Patrones GRASP*. https://es.slideshare.net/Indiana_1969/patrones-grasp-76283581
- Cervantes, H. (2022). *Arquitectura de Software*. SG Buzz. <https://sg.com.mx/revista/27/arquitectura-software>
- Cómo se crean diagramas de clases con notación UML*. (2022). <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagramas-de-clases-con-uml/>
- Deisy. (2015, junio 24). La indisciplina. *LA INDISCIPLINA ESCOLAR*. <http://indiciplinaescolar.blogspot.com/>
- Desarrollo de software: Importancia y técnicas de la captura de requisitos*. (2019). <https://www.atsistemas.com/es/blog/desarrollo-de-software-importancia-y-tnicas-de-la-captura-de-requisitos>
- Diagrama de componentes UML: explicación, creación y ejemplo*. (2022). <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-componentes/>
- Diccionario Español abierto, libre y social*. (2010). <https://www.significadode.org/palabras.htm>
- Estándares de codificación de software y pautas de programación*. (2020). <https://es.parasoft.com/blog/announce-of-prevention-software-safety-security-through-coding-standards/>

- Fernández Castillo, R., Naranjo García, A., & Nápoles Gámez, F. (2012). *CODIS Sistema para la informatización del proceso de comisión disciplinaria de la Facultad 3*. https://repositorio.uci.cu/jspui/handle/ident/TD_05348_12
- Gracia, L. (2013, enero 2). Un poco de Patrones de Diseño GoF (Gang of Four). *Un poco de Java*. <https://unpocodejava.com/2013/01/02/un-poco-de-patrones-de-diseno-gof-gang-of-four/>
- Hernández González, A. (2005). *IDENTIFICACIÓN DE PROCESOS DE NEGOCIO*. *Ingeniería Industrial*. XXVI(1), 54-59.
- ISO 25010. (2022). <https://iso25000.com/index.php/normas-iso-25000/iso-25010?start=3>
- KeepCoding, R. (2022, septiembre 27). *¿Qué es un modelo de datos físicos? | KeepCoding Tech School*. <https://keepcoding.io/blog/modelo-de-datos-fisicos/>
- La observación como método empírico de investigación*. (2019). <http://ri.uaemex.mx/handle/20.500.11799/108147>
- Lago, N. (2022). *Los patrones de diseño en el desarrollo de software | Saasradar*. <https://saasradar.net/patrones-diseno-de-software/>
- Las 10 técnicas de obtención de requisitos más comunes*. (2022). <https://spa.myservername.com/top-10-most-common-requirements-elicitation-techniques>
- LavinScott, E. L. (2014). *Desarrollo de una aplicación informática para el proceso de gestión de las comisiones disciplinarias en la Facultad 3*.
- Matos, Y., & Garcia, D. (2015). *SIPCD versión 2: Aplicación informática para el proceso de gestión de las comisiones disciplinarias en la Facultad 3*. https://repositorio.uci.cu/jspui/bitstream/123456789/7504/1/TD_08029_15.pdf
- Núñez, R. (2022). *Patrón controlador ~ Buenas Prácticas de Programación*. <http://buenaspracticassupc.blogspot.com/2016/05/patron-controlador.html>
- Orión gestión de Procesos Disciplinarios*. (2022). Guía TIC. <https://guiatic.com>
- Parra, D. (2022, noviembre 3). ► Patrones de diseño—Observer. *The power ups*. <https://thepowerups-learning.com/patrones-de-diseno-observer/>
- Patrón de diseño MVC. ¿Qué es y cómo puedo utilizarlo?* (2020). <https://www.easyappcode.com/patron-de-diseno-mvc-que-es-y-como-puedo-utilizarlo>

- PHP - *Guía de estilo de codificación*. (s. f.). Recuperado 28 de octubre de 2022, de <http://coppeldev.github.io/php/standards/psr-2.html>
- Pressman, R. S. (2010). *Ingeniería de software. Un enfoque práctico* (7.ª ed.). MCGRAW-HILL. <https://www.casadellibro.com/libro-ingenieria-del-software-7-ed/9786071503145/1775061>
- Pruebas funcionales / No funcionales: ¿Qué son y para qué sirven? (2019, marzo 26). *Tester House*. <https://testerhouse.com/teoria-testing/pruebas-funcionales/>
- RAE. (2001). *Disciplina | Diccionario de la lengua española (2001)*. «Diccionario esencial de la lengua española». <https://www.rae.es/drae2001/disciplina>
- RAE. (2022). *Diccionario de la lengua española | Edición del Tricentenario*. «Diccionario de la lengua española» - Edición del Tricentenario. <https://dle.rae.es/>
- Refactoring, G. (2014). *Patrones de diseño / Design patterns*. <https://refactoring.guru/es/design-patterns>
- Resolución No. 240*. (2007).
- Rodríguez, V., Hernández, C. C., Juárez Ramírez, A. L., & Martínez Montoya, R. (2015). *Método teórico histórico-lógico*. <https://prezi.com/l1trei4ha4w6/metodo-teorico-historico-logico/>
- Rus Arias, E. (2022). *Método sintético—Qué es, definición y concepto | 2022 | Economipedia*. <https://economipedia.com/definiciones/metodo-sintetico.html>
- Sánchez, V. (2021, marzo 9). *¿Cuáles son las novedades de Bootstrap 5? - Caronte Web Studio*. Blog SEO, Diseño Web & Gráfico | Caronte Web Studio Vitoria-Gasteiz. <https://carontestudio.com/blog/novedades-de-bootstrap-5/>
- Sharma, P. (2019). Las 10 principales ventajas de los servicios de desarrollo de Laravel para empresas. *Cynoteck*. <https://cynoteck.com/es/blog-post/top-10-advantages-of-laravel-development-services-for-enterprises/>
- Sommerville, I. (2002). *Requerimientos del software* (6.ª ed.).
- Soto, N. (2021, julio 2). ¿Qué son los patrones de diseño en programación? *Craft - Code | La Academia de las Buenas Prácticas*. <https://craft-code.com/que-son-los-patrones-de-diseno/>
- The PHP Group. (2021). *PHP: ¿Qué es PHP? - Manual*. <https://www.php.net/manual/es/intro-what-is.php>
- Torrejón, M. (2022, junio 17). Laravel, el framework PHP para crear las mejores webs. *El blog de Omatech*. <https://www.omatech.com/blog/2022/06/17/laravel-el-framework-php-para-crear-las-mejores-webs/>

- Trycore, E. (2019, octubre 18). Tipos de pruebas funcionales de software testing • Trycore. *Trycore*. <https://trycore.co/transformacion-digital/tipos-de-pruebas-funcionales/>
- Universidad de las Ciencias Informáticas. (2022). Universidad de las Ciencias Informáticas. <http://www.uci.cu>
- Urbina, L. (2013). *Digitum: Repositorio Institucional de la Universidad de Murcia: Pregón a cargo de la madrina de las Fiestas de Letras 2013, D.ª Francisca Moya del Baño, y entrega del Premio San Isidoro 2013 al servicio Dumbo Junior de ÁTICA. Hemiciclo de Letras*. <https://digitum.um.es/digitum/handle/10201/83555>
- Vallejo, R. (2017). *Experto en información · Atributos de calidad y patrones de diseño*. <https://fjimenezg.gitbooks.io/atributos-de-calidad-y-patrones-de-diseno/content/experto-en-informacion.html>
- Variación de AUP para la UCI (AUP-UCI)—Metodologías tradicionales. (2022). <https://1library.co/article/variacion-aup-uci-aup-uci-metodologias-tradicionales.zkwgn934>
- Vela Valdez, J. (2007). *Resolucion-No.240-07*.
- Yrzula, L., & Aranda, Y. (2017). □ *CDis: Sistema Informático para la gestión de la información de expedientes disciplinarios de la Facultad 2*. https://repositorio.uci.cu/jspui/bitstream/123456789/8090/1/TD_08852_17.pdf
- Zúñiga, F. G. de. (2015, diciembre 11). Laravel: ¿Qué es? | Arsys. *Blog de arsys.es*. <https://www.arsys.es/blog/que-es-laravel>

Table 13. Entrevista para definir requisitos.

Nombre y Apellidos:		
Marque con una X.		
Cargo:	<input type="checkbox"/> Decano	<input type="checkbox"/> Profesor
<input type="checkbox"/> Asesor de trabajo		
¿Qué características cree usted que debe tener el Módulo de Comisiones Disciplinarias del Sistema de gestión de trabajo Educativo de la facultad 1?		
¿Cómo le gustaría que funcione el Módulo de Comisiones Disciplinarias del Sistema de Gestión de Trabajo Educativo de la Facultad 1 de manera que se vaya a sentir cómodo trabajando con éste?		

Encuesta de satisfacción de ladov:

Estimado profesor al contestar esta encuesta podrá dar a conocer su nivel de satisfacción con relación al Módulo de Comisiones Disciplinarias del Sistema de Gestión de Trabajo Educativo de la Facultad 1 Se espera su sinceridad y se le agradece su colaboración en esta investigación:

1. ¿Le satisface Módulo de Comisiones Disciplinarias desarrollado para contribuir a la gestión de la información generada por estas comisiones a partir del diseño de interfaz de usuario y la arquitectura de información?

Me gusta mucho No me gusta mucho

Me da lo mismo Me disgusta más de lo que me gusta

No me gusta nada No sé qué decir.

2. ¿Considera usted que la Facultad 1 de la UCI precise de un Módulo de Comisiones Disciplinarias que permita una mejor gestión de la información que estas comisiones generan?

Si NO No sé

3. ¿Utilizaría usted un Módulo de Comisiones Disciplinarias para acceder o generar información y conocimientos que tributen al funcionamiento de estas?

Si NO No sé

4. ¿Qué te gustó más del Módulo de Comisiones Disciplinarias del Sistema de Gestión de Trabajo Educativo de la Facultad 1?

5. ¿Qué no te gustó del Módulo de Comisiones Disciplinarias del Sistema de Gestión de Trabajo Educativo de la Facultad 1?

6. ¿Cómo calificarías el Módulo de Comisiones Disciplinarias del Sistema de Gestión de Trabajo Educativo de la Facultad 1?

_____ Suficiente _____ Insuficiente _____ Más que suficiente

7. ¿Cómo consideras la funcionalidad y apariencia del Módulo de Comisiones Disciplinarias del Sistema de Gestión de Trabajo Educativo de la Facultad 1?

Table 14. Cuadro Lógico de ladov para usuarios potenciales.

	¿Considera usted que la Facultad 1 de la UCI precise de un Módulo de Comisiones Disciplinarias que permita una mejor gestión de la información que gestas generan?								
	No	No sé			Sí				
	¿Utilizaría usted un Módulo de Comisiones Disciplinarias para acceder o generar información y conocimientos que tributen al funcionamiento de estas?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
¿Le satisface Módulo de Comisiones Disciplinarias desarrollado para contribuir a la gestión de la información generada por estas comisiones a partir del diseño de interfaz de usuario y la arquitectura de información?									
Me gusta mucho	1	2	6	2	2	6	6	6	6
No me gusta mucho	2	2	3	2	2	3	6	3	6
Me da lo mismo	3	3	3	5	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	5	3	4	3	3	4
No me gusta nada	6	6	6	6	6	4	3	3	4
No sé qué decir	2	3	6	3	3	3	6	6	4

STEFAC1

PERFIL PERSONAL

adminj pq

PANEL DE ADMINISTRACIÓN

- Horario Docente
- Perfil de Usuario
- GECE
- Estrategia Educativa
- Actividades
- Comisión Disciplinaria**
- Comisiones Disciplinarias
- Denuncias
- Expedientes

Listado de Comisiones

Mostrar 5 Entradas

Buscar:

Crear Comisión Disciplinaria

Comisión	Presidente	Secretario	Miembro	Fecha de Creación	Mostrar	Editar	Eliminar
Comisión 1	María Pérez Sánchez	Leslie Prieto Ricardo	Maisabel Perez Vendána	2022-11-21 03:57:30			

Mostrando 1 a 1 de 1 Entradas

Anterior 1 Siguiente

STEFAC1

PERFIL PERSONAL

adminj pq

PANEL DE ADMINISTRACIÓN

- Horario Docente
- Perfil de Usuario
- GECE
- Estrategia Educativa
- Actividades
- Comisión Disciplinaria**
- Comisiones Disciplinarias
- Denuncias**
- Expedientes

Listado de Denuncias

Mostrar 5 Entradas

Buscar:

Denunciados Crear Denuncia

Denuncia	Nombre Denunciante	Comisión	Estado	Fecha de Creación	Mostrar	Editar	Eliminar
Denuncia # 1	Pepe Fernández	Comisión 1	En proceso	2022-11-20 22:31:01			
Denuncia # 2	Luis Rodríguez		Nueva	2022-11-21 04:07:32			
Denuncia # 3	Leslie de las Mercedes Prieto Ricardo		Nueva	2022-11-23 23:04:26			

Mostrando 1 a 3 de 3 Entradas

Anterior 1 Siguiente

STEFAC1

PERFIL PERSONAL
adminj pq

PANEL DE ADMINISTRACIÓN

- Horario Docente
- Perfil de Usuario
- GECE
- Estrategia Educativa
- Actividades
- Comisión Disciplinaria
- Evaluación Integrada
- Gestión de Optativas

Listado de Dictámenes

Mostrar 5 Entradas

Buscar:

Crear Dictámen

Dictámen

Mostrar Editar Eliminar

1-21 15

Mostrando 1 a 1 de 1 Entradas

Anterior 1 Siguiente

¿Estás seguro?
¿Está seguro que desea eliminar?

Aceptar Cancelar

STEFAC1

PERFIL PERSONAL
adminj pq

PANEL DE ADMINISTRACIÓN

- Horario Docente
- Perfil de Usuario
- GECE
- Estrategia Educativa
- Actividades
- Comisión Disciplinaria
- Comisiones Disciplinarias
- Denuncias
- Expedientes

Listado de Expedientes

Declaraciones Opiniones Evidencias Dictámen

Crear Expediente

Mostrar 5 Entradas

Buscar:

Expediente	Nombre y Apellidos del denunciado:	Denuncia en proceso:	Fecha de Creación	Mostrar	Editar	Eliminar
Expediente # 1	est1-1 Alarcón Jorge Denuncia #: 1	Denuncia #: 1	2022-11-20 22:32:44			

Mostrando 1 a 1 de 1 Entradas

Anterior 1 Siguiente

