

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 4



**SISTEMA INFORMÁTICO PARA EL ANÁLISIS DE LAS INDISCIPLINAS DE LOS
ESTUDIANTES DE LA FACULTAD 4**

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

Autor: Gabriel Betancourt Carrasco

Tutor: Dra. C. Dunia María Colomé Cedeño

La Habana, 2015

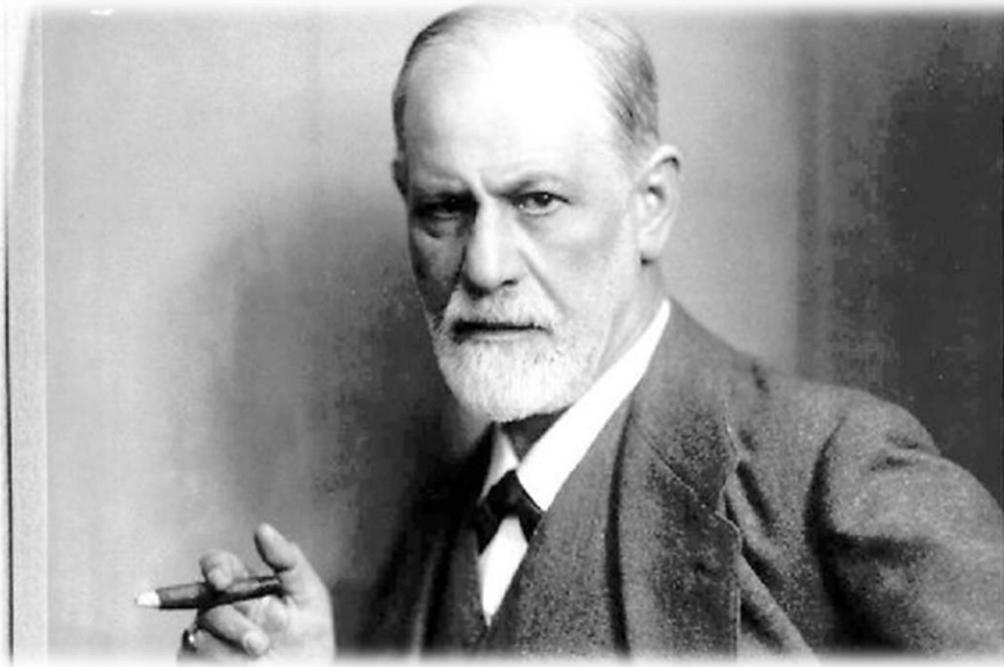
Declaración de autoría

Declaro que soy el único autor del trabajo "Sistema informático para el análisis de las indisciplinas de los estudiantes de la Facultad 4" y autorizo a la Universidad de las Ciencias Informáticas a emplearlo en su beneficio.

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

Autor: Gabriel Betancourt Carrasco

Tutora: Dra. C. Dunia María Colomé Cedeño



Un buen día, echando la vista atrás, se dará usted cuenta de que estos años de lucha han sido los más hermosos de su vida.

Sigmund Freud

1856-1939. Médico y neurólogo austriaco, fundador del psicoanálisis.

DEDICATORIA

A mi mamá, mi papá y mis hermanos por ser lo más importante y preciado que tengo en la vida.

A mi abuela Odalina, por ser mi guía, mi luz, mi todo.

A la memoria de mi abuelo Fidel, por haberme querido tanto en vida, y haberme enseñado cosas de mí que no sabía que sabía.

AGRADECIMIENTOS

A mi mamá, por ser una persona tan pura y llena de cosas lindas. Por estar ahí constantemente y no cansarse de amar a sus hijos.

A mi papá, por apoyarme tanto y haber estado ahí cuando más lo he necesitado. Este logro es tan mío como tuyo.

A mi abuela Odalina, por apoyarme y quererme tanto y ser mi guía e inspiración todos estos años.

A mi abuelo Fidel, por haberme dejado en vida suficiente combustible para lograr todo lo que él quiso que lograra.

A mis hermanos, que me den el gusto de un día ser yo el que pueda verlos graduarse.

A toda mi familia, mis primos, mi padrastro Timino y mis tíos, Julieta, Giraldo y tía Tati, la mejor profesora de matemática que tuve en toda mi trayectoria como estudiante.

A mi novia por ser tan paciente, cariñosa y linda conmigo.

A mi profesora Ada Samper por haberme guiado en mis primeros pasos.

A María Elena, por haberme ayudado tanto durante la carrera, por ser tan incondicional y transparente conmigo desde que la conocí.

Agradecer a mis amigos y compañeros de grupo y universidad por estos fantásticos años que me permitieron compartir con ellos, Lázaro, Wilson, Mayi, Liliana, Jose, Poll, el Sifre.....la lista sería larga y no me alcanza la página.

A mi tutora Dunia por guiarme y apoyarme en esta bonita tarea. Muchas Gracias.

A todos los profes que colaboraron en mi formación como ingeniero.

A la UCI por haberme cambiado la vida, aquí dejo una parte de mí.

RESUMEN

La Resolución No. 240/07 del Ministerio de Educación Superior es el documento legal que regula el análisis de las indisciplinas en estudiantes universitarios. En la Facultad 4 de la Universidad de las Ciencias Informáticas (UCI), no existe una solución informática que facilite el desarrollo de los análisis de las indisciplinas de los estudiantes, lo que trae como consecuencia que la información asociada a los casos no se encuentre siempre disponible, no se cumplan en ocasiones con los tiempos estipulados en dicho reglamento, así como la imposibilidad de acceder de manera fácil a la información de los casos cerrados. En la presente investigación se presenta el análisis, diseño e implementación de un sistema informático que permite gestionar el proceso de análisis de las indisciplinas de los estudiantes de la Facultad 4 de la UCI. Se expone un estudio de las principales herramientas y tecnologías seleccionadas para el desarrollo del sistema informático, así como de la metodología de desarrollo de software a utilizar. También son especificadas las pruebas realizadas al sistema con el objetivo de asegurar que cumpla con los objetivos propuestos.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	7
1.1 Análisis de soluciones existentes.....	7
1.2 Resolución No. 240/07 del MES	8
1.3 Sistemas de gestión de contenidos.....	9
1.3.1 Principales sistemas de gestión de contenidos.....	11
1.3.2 Valoración de los CMS analizados	15
1.3.3 Arquitectura de Drupal	17
1.4 Lenguajes de programación empleados en Drupal.	19
1.4.1 PHP	19
1.4.2 HTML.....	21
1.4.3 CSS.....	21
1.5 Análisis de gestores de bases de datos.....	23
1.5.1 MySQL	23
1.5.2 PostgreSQL.....	24
1.5.3 Comparación y selección del sistema gestor de base de datos	26
1.6 Metodologías de desarrollo de software	28
1.6.1 RUP.....	28
1.6.2 XP.....	29
1.7 Análisis de servidores web	31
1.7.1 Internet Information Services.....	31
1.7.2 Apache	33
Conclusiones	35
CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA INFORMÁTICO	36
2.1 Propuesta de solución	36
2.2 Personas relacionadas con el sistema.....	36
2.3 Exploración	37
2.3.1 Aspectos funcionales y no funcionales.....	37
2.3.2 Historias de usuario	40
2.4 Planificación.....	41
2.4.1 Estimación de esfuerzo por historia de usuario.	41
2.4.2 Plan de iteraciones	43
2.4.3 Plan de duración de las iteraciones.....	43
2.4.4 Plan de entregas.....	44

2.5 Diseño	45
2.5.1 Tarjetas CRC	46
2.5.2 Patrón arquitectónico	47
Conclusiones	49
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA INFORMÁTICO	50
3.2 Temas en Drupal	50
3.3 Implementación	52
3.3.1 Estándares de codificación	52
3.3.1 Módulos utilizados y desarrollados en la solución	53
3.3.2 Patrones de diseño	57
3.3.2 Tablas adicionales creadas en la base de datos	58
3.4 Diagrama de despliegue.....	59
3.5 Tareas de ingeniería.....	60
3.5 Pruebas.....	61
3.5.1 Pruebas unitarias	61
3.5.2 Módulo SimpleTest	62
3.5.2 Módulo Coder.....	63
3.5.3 Pruebas de aceptación	65
Conclusiones	66
CONCLUSIONES GENERALES	67
RECOMENDACIONES	68
BIBLIOGRAFÍA.....	69
ANEXOS	74
Anexo 1. Historias de Usuarios	74
Anexo 2. Tareas de Ingeniería	79
Anexo 3. Tarjetas CRC.....	85
Anexo 4. Pruebas de aceptación.....	87

ÍNDICE DE FIGURAS

Figura 1 Modelo del proceso Análisis de indisciplinas.....	9
Figura 2 Funcionamiento de PHP. (18).....	20
Figura 3 Estructura de CSS. (23).....	22
Figura 4 Sistema PostgreSQL. (27).....	24
Figura 5 Estructura del ciclo de vida del proceso de desarrollo unificado.....	28
Figura 6 Interfaz autenticar usuario.....	52
Figura 7 Tabla correspondientes al módulo Operaciones_caso.	58
Figura 8 Tabla correspondientes al módulo Agravante	59
Figura 9 Diagrama de despliegue	59
Figura 10 Menú de acceso a la interfaz de prueba.....	62
Figura 11 Interfaz del módulo SimpleTest	62
Figura 12 Avance de las pruebas.....	63
Figura 13 Resultados de las pruebas	63
Figura 14 Interfaz del módulo Coder.....	63
Figura 15 Resultado de las pruebas al módulo Operaciones_caso	64
Figura 16 Resultado de las pruebas al módulo Agravante	64

ÍNDICE DE TABLAS

Tabla 1 Comparativa entre Drupal, Joomla y WordPress.....	15
Tabla 2 Características de MySQL y PostgreSQL (29).....	26
Tabla 3 Principales características de IIS (35).....	32
Tabla 4 Personas relacionadas con el sistema.	36
Tabla 5 Requisitos funcionales.....	38
Tabla 6 Historia de usuario No. 1. Crear comisión.....	41
Tabla 7 Historia de usuario No.13. Realizar denuncia	41
Tabla 8 Estimación de cada una de las HU.	42
Tabla 9 Plan de duración de las iteraciones.....	43
Tabla 10 Plan de entregas.....	45
Tabla 11 Tarjeta CRC para gestionar comisiones.	46
Tabla 12 Tarjeta CRC para gestionar casos.	47
Tabla 13 Tarea de Ingeniería No.1.....	60
Tabla 14 Tarea de Ingeniería No.2.....	60
Tabla 15 Prueba de aceptación.No.1	65
Tabla 16 Prueba de aceptación No. 2	66

INTRODUCCIÓN

Actualmente crece vertiginosamente el volumen de la información que se manipula en diversos campos de la sociedad como la economía, las finanzas, la investigación científica, la cultura y el deporte. *“La aparición de Internet ha supuesto un cambio muy importante en estos sectores, facilitando la posibilidad de acceder, organizar y comunicar información relevante con mayor eficiencia”.* (1)

En el ámbito tecnológico, las Tecnologías de la Información y las Comunicaciones (TIC) han presentado avances reconocidos tanto en los soportes físicos como en las posibilidades que ofrecen las redes ya sean fijas o móviles. Estos avances se evidencian en el aumento de la velocidad y la capacidad de procesamiento y almacenamiento de la información, así como en la integración de aplicaciones y dispositivos que permiten el intercambio de información.

En la Universidad de las Ciencias Informáticas (UCI) se estudia la carrera Ingeniería en Ciencias Informáticas la cual tiene entre sus misiones *“formar profesionales integrales, comprometidos con la patria y con el desarrollo del modelo socialista cubano, cuya función esté asociada al desarrollo de la informatización de la sociedad cubana desde tres aristas importantes: el desarrollo de la industria de software nacional, las transformaciones de procesos en las entidades para asumir su informatización y el soporte necesario para su mantenimiento. Estas necesidades están en concordancia con el nivel alcanzado en la informatización de la sociedad, los objetivos que se propone el país, las tendencias internacionales y los problemas profesionales actuales y futuros. El ingeniero en Ciencias Informáticas tiene como objeto de la profesión el proceso de informatización de la sociedad; entendiéndose como tal, la introducción de forma gradual, masiva y planificada, de las TIC en todas las esferas de la sociedad, con el objetivo de incrementar la eficiencia y eficacia en todos los procesos y en aras lograr el aumento en la calidad de vida de los ciudadanos”.* (2)

La consagración al estudio por parte de los estudiantes, así como la exigencia constante de profesores y trabajadores sobre la buena conducta de los alumnos para minimizar las indisciplinas que se puedan cometer, son vitales para obtener el profesional que la sociedad necesita. Resultado de esta exigencia es la aplicación de medidas disciplinarias, respaldadas por la resolución No. 240/07 del Ministerio de

Educación Superior (MES) que rige el reglamento disciplinario de los estudiantes universitarios, una muestra de ello se presenta a continuación. En la Facultad 4 de la UCI, en los últimos dos cursos académicos fueron sancionados 52 estudiantes, las principales indisciplinas registradas están asociadas a la violación de las normas establecidas en el código de ética de seguridad informática, indisciplinas en la beca y ausencias injustificadas a las actividades docentes.

La resolución No. 240/07 plantea la creación de las comisiones disciplinarias por cursos académicos, las cuales tienen como principal objetivo analizar cada indisciplina cometida y proponer la medida disciplinaria que más se ajusta. En el análisis integral del estudiante que comete la falta, se tiene en cuenta la opinión de un miembro de la Federación Estudiantil Universitaria (FEU), de un militante de la Unión de Jóvenes Comunistas (UJC) y del profesor guía de la brigada a la cual pertenece el estudiante. Otras personas pueden emitir sus criterios si la comisión lo considerase necesario. Los elementos que constituyen atenuantes y agravantes también son tenidos en cuenta durante el análisis.

En la Facultad 4 se cumple con lo estipulado en la mencionada resolución, pero durante el proceso de análisis de las indisciplinas denunciadas existen las siguientes limitaciones:

- ✓ Acceder al historial de las medidas disciplinarias aplicadas a los estudiantes implica realizar una búsqueda en los expedientes físicos, almacenados en la secretaría docente de la Facultad.
- ✓ No existe un registro de las denuncias realizadas, concluyan o no con la aplicación de una medida disciplinaria.
- ✓ No se almacenan, de forma digital, los documentos que se obtienen como resultado de los análisis realizados y que conforman el expediente de cada denuncia procesada, lo cual ha provocado la pérdida de información de interés.
- ✓ Resulta difícil conocer las principales indisciplinas cometidas, identificadas por ejemplo, por años académicos y por sexo, así como el comportamiento histórico que permita realizar acciones que tributen a minimizar la cantidad de indisciplinas cometidas.
- ✓ El acceso a toda la información que se va generando durante el análisis, no es de fácil acceso por parte de los miembros de la comisión.

- ✓ El Decano de la Facultad y el profesor principal del año al que pertenece el estudiante analizado, no tienen acceso en todo momento al progreso de la comisión.

Por todo lo expresado anteriormente se establece como **problema a resolver**: ¿Cómo contribuir al desarrollo del proceso de análisis de las indisciplinas de los estudiantes de la Facultad 4?

El **objeto de estudio** de la investigación es el proceso de análisis de las indisciplinas cometidas por los estudiantes universitarios, establecido en la resolución No. 240/07 del MES.

El **campo de acción** está enmarcado en las tecnologías que favorecen el desarrollo del proceso de análisis de las indisciplinas cometidas por los estudiantes universitarios.

El **objetivo general** de la investigación es desarrollar un sistema informático que contribuya al desarrollo del proceso de análisis de las indisciplinas de los estudiantes en la Facultad 4.

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

1. Fundamentar los referentes teóricos, relacionando los aspectos principales que sustentan la investigación, mediante los cuales se consulta, extrae y recopila la información relevante sobre el problema a investigar, en lo fundamental lo relacionado con el análisis de las indisciplinas cometidas por estudiantes universitarios.
2. Realizar el análisis y el diseño del sistema informático para el análisis de las indisciplinas de los estudiantes de la Facultad 4.
3. Implementar el sistema informático para el análisis de las indisciplinas de los estudiantes de la Facultad 4.
4. Evaluar técnicamente el sistema informático para el análisis de las indisciplinas de los estudiantes de la Facultad 4.

Con el propósito de alcanzar los objetivos propuestos se formularon las siguientes **preguntas científicas**:

1. ¿Cuáles son los referentes teóricos y tecnológicos para la producción de un sistema informático para el análisis de disciplinas cometidas por estudiantes universitarios?
2. ¿Qué características esenciales debe poseer un sistema informático para el análisis de las disciplinas cometidas por estudiantes universitarios?
3. ¿Responde el sistema propuesto a las necesidades de la Facultad 4, relacionadas con el análisis de las disciplinas cometidas por los estudiantes universitarios?

Con el objetivo de guiar y perfilar el trabajo hacia el alcance de los objetivos trazados, se definieron las siguientes **tareas a cumplir**:

1. Elaboración del marco teórico de la investigación identificando las diferentes tendencias, ventajas y funcionalidades de los sistemas de gestión.
2. Identificación de las tecnologías informáticas a utilizar para desarrollar el sistema informático para el análisis de las disciplinas de los estudiantes de la Facultad 4.
3. Levantamiento de las funcionalidades a implementar en el sistema informático para el análisis de las disciplinas de los estudiantes de la Facultad 4.
4. Realización de las actividades y artefactos que tributen al análisis y el diseño de la propuesta de solución de acuerdo a la metodología de desarrollo utilizada.
5. Diseño de los prototipos de interfaz de usuario.
6. Implementación del sistema informático.
7. Realización de las pruebas de software al sistema informático para el análisis de las disciplinas de los estudiantes de la Facultad 4.

Entre los métodos científicos y las técnicas utilizadas se destacan los siguientes:

- ✓ **Análisis-síntesis**, permitió el estudio de fuentes bibliográficas referente al tema objeto de investigación, identificando elementos importantes y necesarios para dar solución al problema planteado.
- ✓ **Inductivo-deductivo**, permitió el estudio de diferentes sistemas informáticos que soportan el análisis de disciplinas cometidas por estudiantes universitarios, como son el sistema *CODIS Sistema para la informatización*

del proceso de Comisión Disciplinaria de la Facultad 3 y el Módulo de Consejos Disciplinarios (SIGEPOL), los cuales fueron desarrollados en la UCI, con el objetivo de determinar las soluciones factibles a incorporar en el nuevo sistema.

- ✓ **Modelación**, facilitó la representación explícita de la solución propuesta a través de la modelación del proceso de análisis de las indisciplinas cometidas por estudiantes universitarios.
- ✓ **Análisis documental**, utilizado en la consulta de literatura especializada en temáticas relacionadas con el objeto de estudio. Se realizó el análisis de los documentos impresos, modelos, documentos normativos para lograr una mejor comprensión de los procesos a implementar, como es el caso de la resolución No 240/07 del MES.
- ✓ **Histórico-lógico**, permitió estudiar la ocurrencia de las indisciplinas cometidas por estudiantes de la Facultad 4 en los dos últimos cursos académicos, así como el comportamiento de las comisiones disciplinarias.

La presente investigación se estructura de tres capítulos que describen el proceso de desarrollo de la solución informática. También presenta las conclusiones, las recomendaciones, la bibliografía consultada y los anexos. A continuación se presenta un resumen de los temas abordados en cada uno de los capítulos.

Capítulo 1 Fundamentación teórica. Presenta una descripción de los elementos teóricos asociados al dominio del problema, se analizan los principales conceptos relacionados con el objeto de estudio, así como las principales herramientas y sistemas a emplear, la metodología de desarrollo y otros elementos necesarios para el desarrollo de la solución informática.

Capítulo 2 Análisis y diseño. Se realiza el diseño de la solución informática donde se analizan los elementos principales que intervienen en la aplicación. Se realiza el levantamiento de los requisitos funcionales y no funcionales a tener en cuenta para la implementación del sistema informático, así como la definición de los artefactos correspondientes a la metodología utilizada.

Capítulo 3 Implementación y pruebas. Se muestra detalladamente la evolución del sistema, a través de las fases de la metodología seleccionada. Se implementan las funcionalidades identificadas logrando que el sistema cumpla con los objetivos

definidos. Se describen las pruebas realizadas al sistema informático con el objetivo de asegurar su calidad.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se aborda el proceso de análisis de las indisciplinas cometidas por los estudiantes universitarios, reglamentado por el MES. Además se presenta la manera en la que se implementa este proceso en la Facultad, enfatizando en las principales dificultades que tiene el desarrollo de los análisis de las indisciplinas. También se expone el resultado del estudio realizado sobre diferentes sistemas de gestión de contenidos (CMS¹, por sus siglas en inglés), resaltando sus ventajas y desventajas. Además se analizan diferentes gestores de bases de datos y servidores web con el propósito de identificar los más adecuados para ser utilizados y dar solución a la problemática planteada. En este capítulo se presenta un estudio de diferentes metodologías de desarrollo de software, identificando la que mejor se corresponde con las características del desarrollo que se llevará a cabo.

1.1 Análisis de soluciones existentes.

Se realizó un estudio de algunos de los sistemas de gestión de información jurídica desarrollados en la UCI, para analizar y estudiar procesos similares con el objetivo de enriquecer la presente investigación y lograr un producto novedoso y de calidad. A continuación se describen dichos trabajos:

Análisis y diseño del Sistema Integral de Documentación e Información Judicial para el Tribunal Supremo Popular, es un Trabajo de Diploma realizado en el año 2011, el cual aborda temáticas relacionadas con la presente investigación como son gestionar información concerniente a procesos judiciales. Dicha investigación se propuso como objetivo informatizar procesos que se realizaban de forma manual en el Tribunal Supremo Popular, representados en: lentitud en la prestación de los servicios, duplicación de información, enorme tiempo dedicado a la organización de la documentación y búsqueda de documentos, así como dificultad para la realización de procesos estadísticos, entre otros. Este trabajo permitió almacenar y realizar rápidas búsquedas de información para una mejor prestación de servicios de la mencionada entidad, facilitando su eficacia y eficiencia.

¹Content Management System

Sistema de Gestión Policial (SIGEPOL) “Módulo de Consejos Disciplinarios”, es un Trabajo de Diploma realizado en la UCI en el año 2011. *SIGEPOL* es un Sistema de Gestión Policial desarrollado en la UCI para la República Bolivariana de Venezuela. Dicho trabajo permitió al *SIGEPOL* conformar los consejos disciplinarios de forma automatizada y que exista un registro centralizado de los casos evaluados en los consejos de los cuerpos de policías, dando solución a varias problemáticas planteadas entre la que destacan la inexistencia de un registro centralizado de la información resultante de los procesos de investigaciones de los casos evaluados en Consejos Disciplinarios y el control manual de la información de los funcionarios miembros del Consejo Disciplinario.

Módulos Incidencias e Indisciplinas y medidas disciplinarias del Sistema Informativo de la Dirección de Establecimientos Penitenciarios, es un módulo desarrollado en la UCI para el Sistema de Establecimientos Penitenciarios (*SIDEP*). Este trabajo detalló las fases de diseño, implementación y prueba del ciclo de desarrollo de dicho módulo para su incorporación al *SIDEP*, el cual permite la gestión de incidencias, indisciplinas y medidas disciplinarias del sistema penitenciario.

Atendiendo al estudio realizado se determinó que estos sistemas no pueden utilizarse para gestionar el proceso de análisis de las indisciplinas de los estudiantes de la Facultad 4, ya que el reglamento disciplinario para los estudiantes de la educación superior es independiente de otro código penal establecido, por lo que la implementación de los procesos no puede ser la misma. También destacar que dichas investigaciones fueron realizadas hace años y con herramientas a día de hoy obsoletas, por lo que podrían ocurrir incompatibilidades con la tecnología actual.

El presente trabajo se propone convertir en una alternativa válida para la gestión del proceso de análisis de las indisciplinas, haciendo uso de herramientas y tecnologías actuales.

1.2 Resolución No. 240/07 del MES

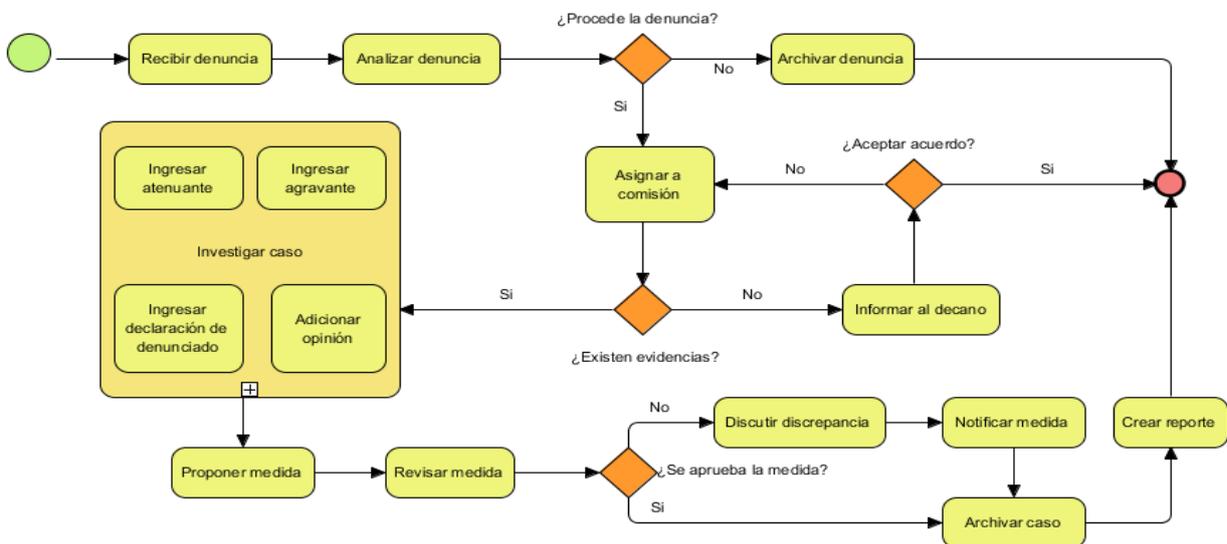
La resolución No. 240/07 del MES entrada en vigor desde el 3 de enero del año 2008 es el documento legal que estipula lo concerniente al reglamento disciplinario para los estudiantes de la educación superior en Cuba. Como resuelve dicho documento “*las faltas disciplinarias que se cometen por los estudiantes de la educación superior*

matriculados en los centros de carácter civil, serán conocidas, juzgadas y sancionadas según las disposiciones del presente Reglamento” (3), este documento proporciona las herramientas legales para llevar a cabo los procesos disciplinarios para estudiantes en el MES.

Este reglamento contiene 14 capítulos en los que se abordan las temáticas relacionadas con las indisciplinas y el proceso que deben llevar a cabo las comisiones disciplinarias. Además son tratadas las áreas de aplicación de las medidas disciplinarias, la calificación de estas atendiendo a su gravedad, el procedimiento a seguir para cada denuncia realizada, los derechos y deberes de los presuntos infractores, así como las apelaciones y las medidas disciplinarias imponibles. Presenta también una serie de disposiciones especiales y transitorias donde se describen situaciones específicas para proceder en casos excepcionales.

En la figura 1 se presentan las principales actividades, determinadas en el reglamento mencionado, para desarrollar el proceso de análisis de las indisciplinas cometidas por estudiantes universitarios.

Figura 1 Modelo del proceso Análisis de indisciplinas.



1.3 Sistemas de gestión de contenidos

Internet ha evolucionado con los años, de simples páginas HTML hasta las bellas páginas dinámicas y aplicaciones web de hoy en día. Uno de los responsables de esta

evolución han sido los CMS, que permite a los usuarios producir y gestionar el contenido de un sitio web.

Un sistema de gestión de contenidos (CMS) es un software informático o una colección de herramientas que es usada para gestionar información. Con la ayuda de un buen CMS, se pueden ingresar datos y almacenarlos en la base de datos. A veces, puede ser editado por ciertos usuarios autorizados y finalmente, puesto a disposición del público.

Un CMS profesionalmente diseñado, puede gestionar el formateo, catalogación, almacenamiento y recuperación de la información. En el proceso, no hay necesidad de que los usuarios tengan demasiados conocimientos técnicos. Hoy en día, los CMS son usados por librerías, diarios, tiendas online, actividades académicas, etc. (4)

La evolución de internet hacia portales con más contenido y la alta participación de las personas a través de blogs y redes sociales, ha convertido a los gestores de contenidos en una herramienta esencial en la red, tanto para empresas e instituciones como para las personas.

Los primeros sistemas de administración de contenidos fueron desarrollados por organizaciones que publicaban una gran cantidad de contenido en internet, y necesitaban de continuas actualizaciones; como revistas en línea, periódicos y publicaciones.

Estructuralmente, un CMS se divide en tres capas fundamentales: (6)

1. **La capa de la base de datos**, la base de datos tiene almacenado todo el contenido que se ha escrito en la web, así como muchos de los parámetros de configuración, categorías, organización, usuarios y contraseñas. Los sistemas de bases de datos más habituales de los CMS suelen ser MySQL o PostgreSQL.
2. **La capa de programación**, está contenida en los ficheros de la web. Estos ficheros al ejecutarse solicitan la información que el usuario ha pedido desde el navegador de internet y la extrae para mostrarla al usuario, ordenada y

estructurada, insertándola en los lugares que le corresponde dentro del diseño de la página web. El lenguaje de programación más habitual suele ser PHP.

- 3. La capa de diseño**, reside también en algunos ficheros. Define el diseño de la web, es decir “la maqueta o dibujo” sobre la que se insertará el contenido que la programación se encarga de extraer de la base de datos. El lenguaje de programación y maquetación de la web es el HTML y CSS (complementado en ocasiones con JavaScript y AJAX²).

Los CMS cuentan con múltiples características y funcionalidades, orientadas a diferentes tipos de proyectos, entre las más generales y comunes destacan:(7)

- ✓ Creación, edición, publicación y gestión de diferentes tipos de contenidos.
- ✓ Gestión de los usuarios y su acceso a los contenidos del sitio mediante roles y permisos.
- ✓ Despliegue sencillo y rápido.
- ✓ Creación de páginas para los motores de búsqueda.
- ✓ Inclusión de palabras clave en la URL de cada página.
- ✓ Mantenimiento y pruebas del sitio.
- ✓ Asignación de roles y responsabilidades a las diferentes categorías o tipos de contenido.
- ✓ Seguimiento y gestión múltiple de versiones de un solo tipo de contenido.
- ✓ Metadatos de arquitectura, clasificación, navegación, diseño de búsqueda y diseño.
- ✓ Mejora de la seguridad de contenidos.
- ✓ Cambio del diseño en cualquier momento sin afectar el contenido almacenado en el sitio.

1.3.1 Principales sistemas de gestión de contenidos

Existen diferentes tipos de CMS, según lenguajes de programación empleados, tipo de licencia, así como por su uso y funcionalidad. Existen CMS que se ajustan a diferentes propósitos de desarrollo, por ejemplo, para la creación de un blog o una página personal, *WordPress* o *Blogger* serían ideales; para diseñar plataformas de educación

² AJAX, acrónimo de Asynchronous JavaScript And XML, es una técnica de desarrollo web para crear aplicaciones interactivas.

en línea o llamadas e-learning³, son elegibles *Moodle*, *Dokeos* o *ATutor*. Si la intención es crear un sitio de comercio electrónico o una tienda virtual la mejor opción sería *osCommerce* o *Magneto*, así como otros que son útiles para desarrollar redes sociales, como es el caso de *elgg* o *Ning*. Si lo que se desea es crear un portal web con variadas funcionalidades, dígase algunas de las anteriormente mencionadas o todas ellas en su conjunto, la mejor opción sería inclinarse por algunos de los CMS genéricos disponibles como *WordPress*, *Joomla* o *Drupal*. (6)

Joomla, WordPress y Drupal se encuentran entre los CMS más reconocidos y utilizados (8)(9), entre otras razones por su carácter genérico, por lo anterior el análisis en esta investigación se centrará en estos tres sistemas gestores de contenido para seleccionar el que más se adapte a las necesidades de desarrollo de la propuesta de solución. A continuación se presenta el análisis de estos sistemas.

Joomla

Es un CMS que permite desarrollar sitios web dinámicos e interactivos. Su panel de administración permite crear, modificar o eliminar contenido de un sitio web de manera rápida y sencilla. Es un software desarrollado en PHP y de código abierto.

Joomla permite gestionar con mucha facilidad la web, crear un nuevo apartado, modificar los actuales, añadir nuevas imágenes, crear nuevas opciones de menú. No requiere amplios conocimientos técnicos. (10)

Este CMS puede ser empleado para crear diversos tipos de aplicaciones web, entre las que se puede mencionar:(11)

- ✓ Sitios corporativos y portales.
- ✓ Revistas en línea, periódicos y publicaciones.
- ✓ Comercio electrónico y reservaciones.
- ✓ Sitios web para pequeños negocios.
- ✓ Sitios web personales y blogs.

³ Se denomina aprendizaje electrónico (en inglés e-learning) a la educación a distancia virtualizada a través de canales electrónicos.

Joomla está desarrollado con una arquitectura Modelo-Vista-Controlador (MVC). Esta arquitectura permite interactuar directamente con las vistas, posibilitando un gran nivel de personalización en el desarrollo de las plantillas.

WordPress

Es un CMS enfocado en la creación de diversos tipos de sitios web, aunque generalmente se utiliza para la creación de blogs. Está desarrollado en lenguaje PHP para entornos que ejecutan MySQL y Apache. Es un software libre que tiene actualmente gran popularidad en la comunidad CMS de desarrollo por su facilidad de uso y por las características con las que cuenta como gestor de contenido.

Entre las características que más destacan en este gestor de contenido se pueden mencionar: (12)

- ✓ Puede actuar como gestor de contenidos, como blog o como ambos simultáneamente, lo que le permite disponer de un sitio web empresarial con su blog corporativo, todo ello gestionado de forma sencilla por una única herramienta.
- ✓ Sus páginas y artículos se generan dinámicamente a medida que se realizan las publicaciones, por lo que la actualización es fácil y rápida.
- ✓ Los artículos se organizan en categorías, lo que facilita el trabajo para los usuarios.
- ✓ Permite crear diferentes usuarios con distintos derechos de administración y edición de contenidos.
- ✓ Se instala localmente en el propio servidor, lo que proporciona mayor control sobre su configuración.

El uso de WordPress brinda gran seguridad al desarrollador, debido a que su evolución ha estado dirigida hacia convertirse en un gestor de contenidos más flexible, posibilitando que la curva de aprendizaje sea menor. Cuenta actualmente con una comunidad de desarrolladores que trabaja en nuevas versiones para garantizar una mayor seguridad y rendimiento de la plataforma. Se integra fácilmente al comercio

electrónico mediante el uso de *plugins*⁴ gratuitos. Además, posibilita la creación de diferentes roles asociados a la gestión de contenidos.

Drupal

Drupal es un CMS de código abierto de fácil configuración, que permite la publicación de artículos, imágenes, archivos, etc. Es un sistema dinámico que permite almacenar el contenido en una base de datos y editarlo en un entorno web. Está desarrollado en lenguaje PHP y es mantenido por una gran comunidad de usuarios. Su usabilidad, consistencia, calidad del código y de las páginas generadas, lo destacan entre los diferentes tipos de CMS.

Drupal ofrece la posibilidad de ampliar las características del sistema con nuevas funcionalidades, en el momento que las requiera debido a su sistema modular, cuestión que facilita ampliamente el trabajo de los desarrolladores. Varias de esas funcionalidades avanzadas y necesarias en los sistemas web que se pueden obtener con el uso de Drupal, son las siguientes: (13)

- ✓ **Galería de imágenes**, posibilita subir imágenes a través del navegador web. El gestor de contenidos se encarga de ajustarlas a las medidas necesarias para su visualización en la web.
- ✓ **Portada y visualización de datos dinámica**, es posible tener diferentes portadas según tipos de usuarios, hora del día, día de la semana, etc. A la vez estas portadas se pueden generar de manera dinámica, escogiendo cuáles deben ser los contenidos a mostrar y cuándo.
- ✓ **Subir archivos (PDF, ODT, DOC)**, el gestor de contenidos permite adjuntar archivos para ser descargados, a los diferentes contenidos de la web. Se pueden establecer cuotas y permisos de subida de archivos a los diferentes usuarios de la web.
- ✓ **Formularios personalizados**, posibilita construir formularios personalizados. Los datos pueden ser enviados por correo electrónico y/o almacenados en la base de

⁴ Un plugin es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande.

datos del gestor de contenidos. También pueden ser exportados a archivos para ser procesados en hojas de cálculo.

- ✓ **Formularios de contacto**, permite emplear formularios de contactos para poder enviar correos electrónicos con archivos adjuntos directamente a través de un formulario en la web.
- ✓ **Herramientas para redes sociales y comunidades**, permite enviar mensajes privados entre los usuarios, chat, lista de contactos, indicadores de números de usuarios en línea y gestión de perfiles de usuarios integrados en el sistema. También posibilita la difusión de contenidos de la web en redes sociales como Facebook y Twitter.
- ✓ **Integración con aplicaciones de Google**, posibilita la integración con aplicaciones de Google como *GoogleMaps*, *Google Analytics* y *Google AdSense*.
- ✓ **Agenda de eventos**, permite emplear un sistema para la publicación de eventos con fecha de inicio y fin con la posibilidad de enviar notificaciones por correo electrónico.

1.3.2 Valoración de los CMS analizados

Es recomendable antes de escoger que tecnología usar para desarrollar el producto, detenerse a analizar qué tipo de sistema informático queremos lograr, con estos conceptos bien definidos existe mayor probabilidad de que la selección sea la más adecuada para el desarrollo del proyecto.

Atendiendo a los análisis hechos anteriormente de los CMS Drupal, Joomla y WordPress se realizará una comparativa de las cualidades y características de estos, con el objetivo de identificar cual es el más adecuado para desarrollar la propuesta de solución. En la siguiente tabla se describen características fundamentales de los CMS y el índice de calidad de cada CMS asociado a esa característica: (14)

Tabla 1 Comparativa entre Drupal, Joomla y WordPress

Característica	Drupal	Joomla	WordPress
Usabilidad	Media	Baja	Alta

Flexibilidad	Muy alta	Baja	Alta
Tráfico soportado	Alto	Medio	Bajo
Calidad de la documentación	Alta. Bien programado y documentado.	Poca documentación interna y en foros.	Alta. Muy bien documentado internamente.
Complejidad	Alta	Baja	Baja
Seguridad	Alta	Media	Media-baja
Posibilidades de desarrollo	Muy alta	Media	Alta
Comunidad	Muy alta	Media-alta	Alta

Las valoraciones que se aprecian en la tabla son fruto del análisis de las características más necesitadas para la implementación del sistema informático. Seguridad, posibilidad de desarrollo, flexibilidad y calidad de la documentación asociada, son aspectos muy importantes a tener en cuenta para la selección del CMS a utilizar, así como el grueso de la comunidad de desarrolladores, ya que esta cuestión se traduce en más opciones de ampliación y desarrollo mediante los módulos contribuidos por los usuarios que la componen. Atendiendo a esos aspectos comparativos, Drupal se presenta como el CMS ideal para desarrollar el producto.

El sistema de módulos es diferente para cada plataforma. Los módulos en Drupal tienen más granularidad, esto significa que para realizar una funcionalidad se puede necesitar de varios módulos, uno que gestione las imágenes, uno que permita crear diferentes campos o mostrar diferentes vistas, etc., esto genera más trabajo pero el resultado es muy manejable. En WordPress o Joomla es distinto, con un solo módulo se pueden configurar muchas acciones, pero esta es una cuestión contraproducente al adaptarse a casos de necesidades específicas. (15)

En Drupal todo se ha diseñado de manera muy abstracta e inteligente, la arquitectura es diferenciada y está estructurada en distintas capas: con lo que el CMS está muy

compartimentado, frente al resto de CMS representa una mayor flexibilidad y más potencia para crecer y llevar a término funcionalidades complejas. (16)

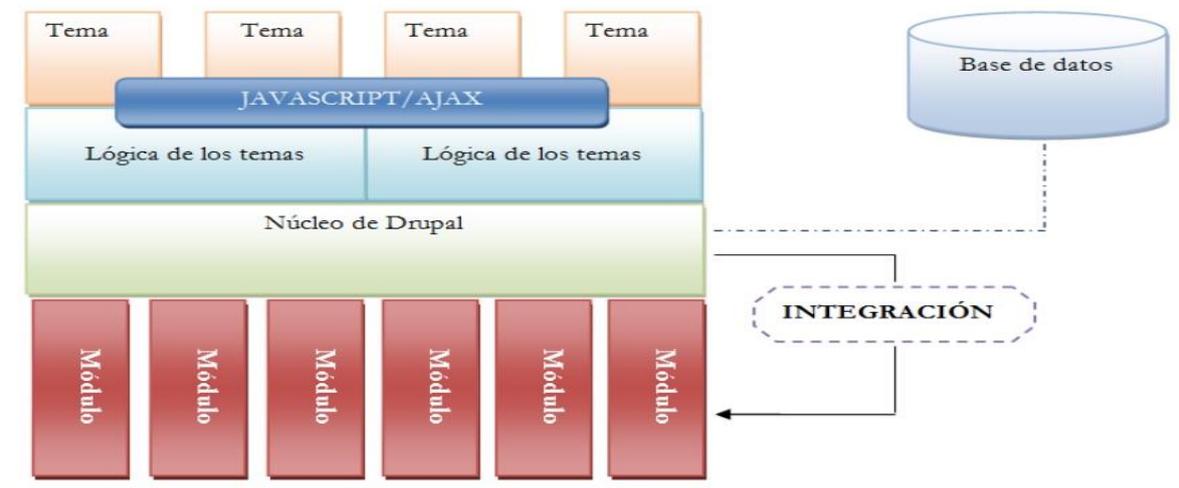
A raíz de todo lo expuesto anteriormente, se seleccionó Drupal como el CMS a utilizar para implementar el sistema informático.

1.3.3 Arquitectura de Drupal

Drupal es un CMS modular, el núcleo del sistema se complementa con algunos módulos obligatorios y otros opcionales, que vienen dentro de su distribución. Además de estos módulos, se pueden usar otros módulos adicionales desarrollados por otras personas. Su lógica está programada en PHP, siguiendo un modelo de programación estructurada y que hace uso de un sistema de base de datos relacional, como MySQL o PostgreSQL.

En la siguiente ilustración se muestra de forma esquemática los elementos que forman un sistema Drupal.

Figura 2 Arquitectura de Drupal (6)



El código que constituye el núcleo de Drupal está formado por un conjunto de librerías que permiten gestionar los procesos de arranque del sistema. Estas librerías ofrecen además un conjunto de servicios que permiten integrar las funcionalidades adicionales de los módulos, servicios como conexión y administración de la base de datos, tratamiento de imágenes, internacionalización, soporte para la codificación y un potente entorno de integración de utilidades. (6)

La arquitectura de Drupal está conformada por varios elementos fundamentales, a continuación una breve descripción de estos:(45)

El núcleo, aporta a Drupal la base necesaria para su funcionamiento y para la incorporación del resto de componentes de la arquitectura. Es posible acceder al núcleo y hacer uso directo de sus funciones a través de la API de programación de Drupal.

Los módulos, aportan funcionalidades adicionales al núcleo de Drupal. En el desarrollo de los módulos contribuyen tanto los desarrolladores de Drupal, como el resto de miembros de la comunidad, existiendo en la actualidad más de 2.500 módulos disponibles.

Área de administración, toda la administración del sitio se lleva a cabo a través del menú de Administración (Management). El menú de administración viene incorporado en la interfaz del sitio, a través de una barra de herramientas situada en la parte superior de la página. El menú de Administración se divide en grupos de tareas, teniendo inicialmente las siguientes opciones principales: Panel de control, Contenido, Estructura, Apariencia, Personas, Módulos, Configuración, Informes y Ayuda.

Nodos y tipos de contenido, los tipos de contenido en Drupal derivan de un tipo de contenido básico denominado nodo (node en inglés). El tipo de contenido principal es la Página básica, que se utiliza para contenidos estáticos del sitio.

Entidades y campos, las entidades son elementos a los que se le puede añadir campos de información de diferentes tipos (texto, imagen, fecha, etc.). La finalidad de las entidades es homogenizar la gestión y presentación de campos adicionales, algunas de ellas son los usuarios, los nodos, los términos de taxonomía y los comentarios.

Menús, los menús facilitan la organización de los nodos publicados. Drupal integra un potente gestor de menús que permite disponer de múltiples y variados menús en un mismo sitio web. Los menús se pueden colocar en distintas áreas o regiones de un tema y se adaptan al diseño gráfico del sitio, establecido a través del tema seleccionado.

Bloques, los bloques (blocks en inglés) son contenidos principalmente dinámicos que se pueden habilitar en distintas zonas (denominadas regiones) del tema del sitio. Por ejemplo, un bloque puede mostrar los últimos usuarios registrados, los últimos comentarios publicados en el sitio o un calendario de eventos. Los bloques también pueden habilitarse y deshabilitarse de forma dinámica, por ejemplo, para determinados contenidos o roles.

Temas, el tema (theme en inglés) define un diseño específico para el sitio web. Existe un repositorio oficial de temas libres de Drupal, que pueden ser descargados y modificados para adaptarlos al diseño de nuestro sitio. Mediante el uso de temas, Drupal separa los contenidos del diseño, de forma que es posible cambiar el aspecto del sitio cambiando o modificando el tema. La forma más fácil de personalizar los temas es modificando las hojas de estilo CSS que contienen.

Usuarios, roles y permisos, el control de acceso de los usuarios a las distintas funcionalidades del sitio se realiza a través de los roles y permisos. Un rol es un conjunto de permisos, y cada usuario puede tener asignados diferentes roles. De esta forma se controla si el usuario puede o no realizar una determinada acción.

Taxonomía, la taxonomía permite la clasificación de los contenidos del sitio. El módulo Taxonomy de Drupal está constituido por dos elementos fundamentales: los vocabularios (o categorías) y los términos (o etiquetas). Cada vocabulario puede agrupar a uno o más términos. Drupal permite a través de los vocabularios categorizar los nodos y por medio de los términos describir aspectos particulares de éstos.

1.4 Lenguajes de programación empleados en Drupal.

Al seleccionar el CMS Drupal, el autor de esta investigación expone en los siguientes epígrafes las principales características de los lenguajes de programación que son empleados en este sistema gestor de contenidos.

1.4.1 PHP

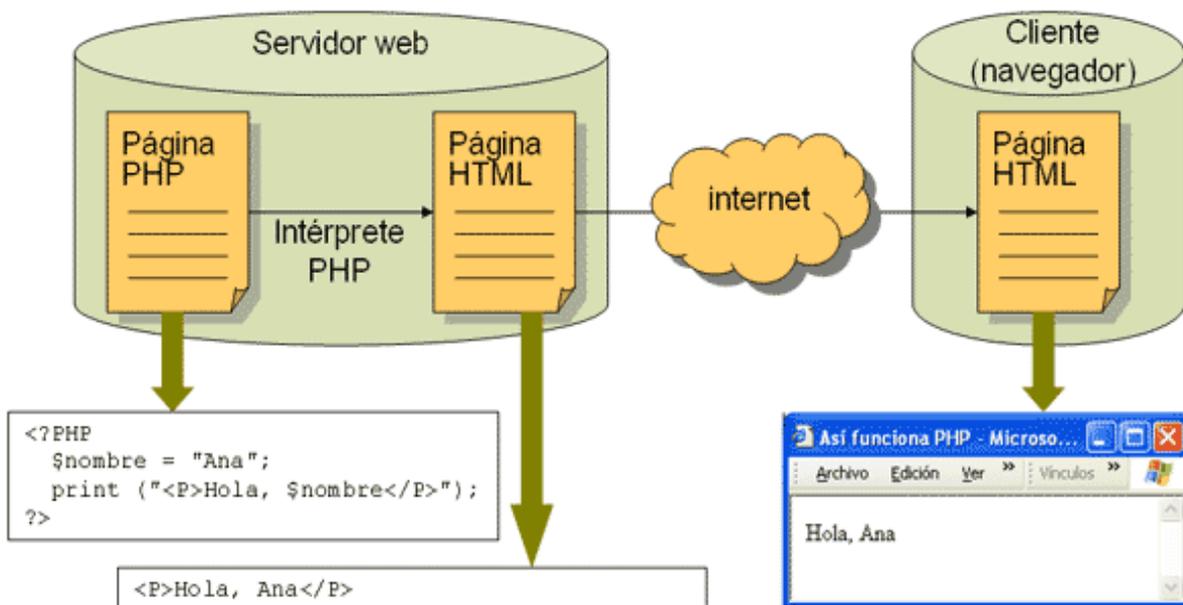
PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Un gran número de páginas y portales web están creadas con PHP. Que sea código abierto significa que es de uso libre y gratuito para

todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo se podrá combinar código PHP con código HTML, siguiendo determinadas reglas.

PHP se utiliza para generar páginas web dinámicas. Recordar que se llama página estática a aquella cuyos contenidos permanecen siempre igual, mientras que las páginas dinámicas son aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc. (17)

El lenguaje PHP se interpreta en servidores, que son potentes ordenadores con un software y hardware especial. Cuando se escribe una dirección tipo *http://www.ejemplo.uci.cu/index.php* en un navegador web como Internet Explorer, Firefox o Chrome, se envían los datos de la solicitud al servidor que los procesa, reúne los datos (por eso se dice que es un proceso dinámico) y el servidor lo que devuelve es una página HTML como si fuera estática.

Figura 2 Funcionamiento de PHP. (18)



Lo mejor de usar PHP es que es simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales y más avanzados.

En resumen, PHP es un lenguaje potente, muy usado y de gran interés para los desarrolladores web por su gran cantidad de funcionalidades y características.

1.4.2 HTML

HTML es un lenguaje de etiquetado que se utiliza para el desarrollo de páginas de Internet. Se trata de un formato abierto que surgió a partir de las etiquetas SGML (*Standard Generalized Markup Language*), concepto traducido generalmente como “Estándar de Lenguaje de Marcado Generalizado” y que se entiende como un sistema que permite ordenar y etiquetar diversos documentos dentro de una lista. Este lenguaje es el que se utiliza para especificar los nombres de las etiquetas que se utilizarán al ordenar, no existen reglas para dicha organización, razón por la cual se plantea que es un sistema de formato abierto. (19)

Este lenguaje está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla. HTML dispone de etiquetas para imágenes, hipervínculos que permiten la dirección a otras páginas, saltos de línea, listas, tablas, etc. Además de permitir la creación de las páginas web, posibilita darles estructura y contenido.

El marcado estructural es el que estipula la finalidad del texto, aunque no define cómo se verá el elemento. Por su parte, el marcado relacionado con la presentación es el que se encarga de señalar cómo se verá el texto, más allá de su función.

1.4.3 CSS

Hojas de Estilo en Cascada (CSS⁵, por sus siglas en inglés), es un mecanismo simple que describe la forma en la que se mostrará un documento en la pantalla, se imprimirá, o incluso la manera en la que será pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos

⁵ *Cascading Style Sheets, del inglés*

ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. (20)

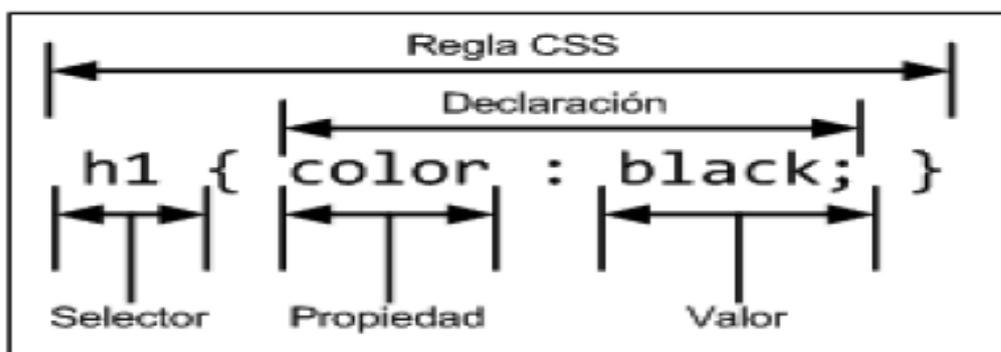
Se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los estilos definen la forma de mostrar dichos elementos. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento. (21)

La filosofía de CSS se basa en intentar separar lo que es la estructura del documento HTML de su presentación. La página web es lo que hay en el contenido general, textos, imágenes, etc. y el CSS se entiende como un cristal de color que hace que el contenido se vea de una forma u otra.

CSS tiene una sintaxis muy sencilla, que usa algunas palabras clave tomadas del inglés para especificar los nombres de varias propiedades de estilo. Una hoja de estilo se compone de una lista de *reglas*. Cada regla o conjunto de reglas consiste en uno o más *selectores* y un *bloque de declaración* (o bloque de estilo), con los estilos a aplicar para los elementos del documento, que cumplan con el selector que les precede. (22)

En la siguiente figura se describe la estructura de CSS.

Figura 3 Estructura de CSS. (23)



Las *reglas* son declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración

está compuesta por una propiedad y el valor que se le asigne. El *selector* funciona como enlace entre el documento y el estilo, especificando los elementos que se van a ver afectados por esa declaración. La declaración es la parte de la regla que establece cuál será el efecto. (21)

Las Hojas de Estilo en Cascada son un estándar muy amplio, con unas especificaciones y posibilidades muy grandes. Al conocer y dominar bien los procedimientos de CSS existe una garantía de que el sitio web que se esté desarrollando cuente con un diseño atractivo y muy elegante atendiendo a su cantidad de facilidades y características.

1.5 Análisis de gestores de bases de datos

A continuación se presenta el análisis de los sistemas gestores de bases de datos MySQL y PostgreSQL.

1.5.1 MySQL

MySQL es un sistema de administración de bases de datos para bases de datos relacionales. Existen distintos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información. Fue escrito en los lenguajes C y C++, se destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java, y su integración en distintos sistemas operativos. (24)

El sistema de base de datos MySQL es hoy en día uno de los más importantes en lo que a diseño y programación de base de datos de tipo relacional se refiere. Cuenta con millones de aplicaciones y aparece en el mundo informático como una de las más utilizadas por usuarios del medio. El programa MySQL se usa como servidor a través del cual pueden conectarse múltiples usuarios y utilizarlo al mismo tiempo.

Entre las características más interesantes de MySQL se encuentra la de recurrir a bases de datos multiusuarios a través de la web y en diferentes lenguajes de programación, que se adaptan a diferentes necesidades y requerimientos. Es conocida por desarrollar alta velocidad en la búsqueda de datos e información, a diferencia de sistemas anteriores. (25)

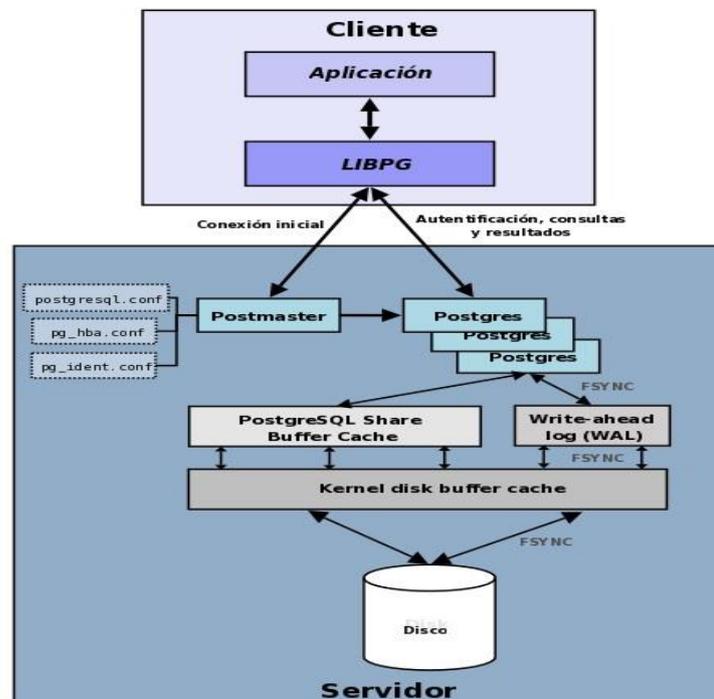
Este sistema gestor de base de datos tiene grandes potencialidades, entre las que destacan: (26)

- ✓ Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- ✓ Posee un buen control de acceso de usuarios y seguridad en los datos.
- ✓ Buena integración con el lenguaje PHP.
- ✓ Gran portabilidad entre distintos sistemas y plataformas.

1.5.2 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo la licencia BSD⁶ y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado. En la figura 4 se describen de manera general los componentes más importantes en un sistema PostgreSQL.

Figura 4 Sistema PostgreSQL. (27)



⁶ BSD son las iniciales de Berkeley Software Distribution, se utiliza para identificar un sistema operativo derivado del sistema Unix, su nombre proviene de los aportes realizados a ese sistema por la Universidad de California en Berkeley.

A continuación se realiza una descripción de cada uno de los componentes definidos en la figura anterior:(27)

- ✓ **Aplicación cliente:** esta es la aplicación cliente que utiliza PostgreSQL como administrador de bases de datos. La conexión puede ocurrir vía TCP/IP o sockets locales.
- ✓ **Demonio postmaster:** este es el proceso principal de PostgreSQL. Es el encargado de escuchar por un puerto/socket por conexiones entrantes de clientes. También es el encargado de crear los procesos hijos que se encargaran de autenticar estas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes.
- ✓ **Ficheros de configuración:** Los tres ficheros principales de configuración utilizados por PostgreSQL son: postgresql.conf, pg_hba.conf y pg_ident.conf.
- ✓ **Procesos hijos postgres:** procesos hijos que se encargan de autenticar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes.
- ✓ **PostgreSQL share buffer cache:** memoria compartida usada por PostgreSQL para almacenar datos en caché.
- ✓ **Write-Ahead Log (WAL):** componente del sistema encargado de asegurar la integridad de los datos.
- ✓ **Kernel disk buffer cache:** caché de disco del sistema operativo.
- ✓ **Disco:** disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione.

Es un gestor de bases de datos altamente personalizable, en PostgreSQL es posible escribir procedimientos almacenados en más de una docena de lenguajes como son Java, Perl, Python, Ruby, pgSQL, etc. Los disparadores (*triggers*) y procedimientos almacenados pueden ser escritos en C y se cargan en la base de datos como una biblioteca, lo cual permite una gran flexibilidad y ampliación de sus capacidades. Del mismo modo, PostgreSQL incluye un framework que permite a los desarrolladores definir y crear sus propios tipos de datos personalizados.

El código fuente de PostgreSQL está disponible bajo una licencia de código abierto: la licencia de PostgreSQL. Esta licencia le da la libertad para usar, modificar y distribuir

PostgreSQL en cualquier forma que desee, ya sea de código abierto o cerrado. Como tal, PostgreSQL no es sólo un sistema de base de datos de gran alcance capaz de usarse en las empresas, es todo una plataforma de desarrollo sobre la cual es posible desarrollar diversos tipos de software. (28)

1.5.3 Comparación y selección del sistema gestor de base de datos

Después de hacer un análisis de los posibles gestores de base de datos a utilizar, se presenta una comparación de estos sistemas, con el objetivo de definir con más profundidad las ventajas y desventajas, características y principales funcionalidades de estos potentes gestores de bases de datos.

Tabla 2 Características de MySQL y PostgreSQL (29)

Aspectos	MySQL	PostgreSQL
Características principales	Licencia GPL y también posee una licencia comercial para aquellas empresas que deseen incluirlo en sus aplicaciones privativas.	Licencia BSD.
	Soporte para transacciones y agrupación de transacciones.	Realiza varias operaciones al mismo tiempo sobre la misma tabla sin necesidad de bloquearla.
	Gestión de diferentes usuarios y los permisos asociados a estos.	Tiene mejor soporte para <u>triggers</u> y procedimientos en el servidor.
	Muy buena integración con PHP.	Implementa el uso de subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz.
Herramientas de Administración	phpMyAdmin	PgAdmin3
	MySQL Administrator	PgAccess
	PCCS MySQL Database AdminTool	PhpPgAdmin

Ventajas	Mayor rendimiento. Mayor velocidad tanto al conectar con el servidor como al servir <u>selects</u> y demás.	Por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs y la cantidad de RAM.
	Mejores utilidades de administración (backup, recuperación de errores, etc.).	Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial).
	Buen control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.	Soporta un subconjunto de SQL92, mayor que el que soporta MySQL.
Desventajas	No soporta transacciones, "roll-backs" ni subselects.	Alto consumo de recursos.
	No considera las claves ajenas.	Menos funciones en PHP.
	Ignora la integridad referencial, dejándola en manos del programador de la aplicación.	Es algo más lenta que MySQL.

Con lo expuesto anteriormente se puede concluir que ambos son potentes gestores de bases de datos y cuentan con muchas características y funcionalidades que garantizan calidad y estabilidad en cualquier sistema.

Cada uno de estos gestores posee características que los convierten en una gran opción de uso dependiendo de las principales necesidades del desarrollador. Muchas de las funcionalidades de MySQL tienen carácter privativo lo que dificulta su uso. Por su condición de software libre, así como por su potencia, robustez y rendimiento, PostgreSQL fue seleccionado para el desarrollo de la solución propuesta.

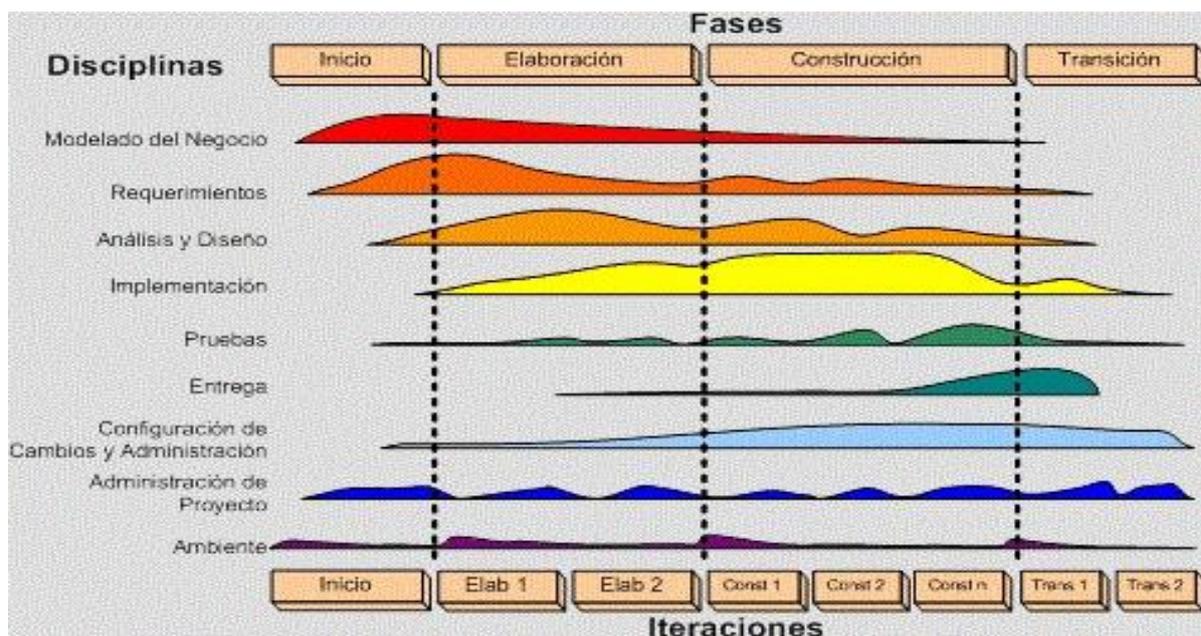
1.6 Metodologías de desarrollo de software

1.6.1 RUP

El proceso unificado conocido como RUP, es una metodología de desarrollo de software que trabaja a gran escala, mediante un proceso continuo de pruebas y retroalimentación, garantizando el cumplimiento de ciertos estándares de calidad.

El proceso de desarrollo constituye un marco metodológico que define en términos de metas estratégicas, objetivos, actividades y artefactos (documentación) requerido en cada fase de desarrollo. Esto permite enfocar el esfuerzo de los recursos humanos en términos de habilidades, competencias y capacidades para asumir roles específicos con responsabilidades bien definidas. (30)

Figura 5 Estructura del ciclo de vida del proceso de desarrollo unificado



Como se observa en la figura 5, la metodología de desarrollo RUP se compone de varias fases de desarrollo. A continuación se realizará una breve descripción de estas fases: (30)

Concepción o inicio: esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos potenciales asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones.

Elaboración: en la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.

Construcción: el propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.

Transición: el propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

RUP describe cómo utilizar de forma efectiva reglas de negocio y procedimientos comerciales probados en el desarrollo de software para equipos de desarrollo de software, conocidos como “mejores prácticas”. Captura varias de las mejores prácticas en el desarrollo moderno de software en una forma que es aplicable para un amplio rango de proyectos y organizaciones. Es una guía de cómo utilizar de manera efectiva UML. Provee a cada miembro del equipo fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas de desarrollo. Crea y mantiene modelos, en lugar de enfocarse en la producción de una gran cantidad de documentación. (32)

1.6.2 XP

La programación extrema (XP, por sus siglas en inglés) es una metodología de desarrollo ágil. Su autor principal es Kent Beck⁷ quien eligió algunas características de otras metodologías y las relacionó de forma que cada una complementara a la otra. XP se puede definir como un conjunto de pasos de diversas metodologías, acoplados de manera que sean flexibles a seguir utilizados con el uso común, para realizar un

⁷ Kent Beck es un ingeniero de software estadounidense, uno de los creadores de las metodologías de desarrollo de software de programación extrema.

desarrollo eficaz. Esta metodología tiene como base la simplicidad y como objetivo principal la satisfacción del cliente; para lograrlo se deben tomar en cuenta cuatro valores fundamentales: (33)

- ✓ **Comunicación:** es muy importante que haya una comunicación constante con el cliente y dentro de todo el equipo de trabajo, de esto dependerá que el desarrollo se lleve a cabo de una manera sencilla, entendible y que se entregue al cliente lo que necesita.
- ✓ **Simplicidad:** en XP se refiere que ante todo y sin importar qué funcionalidad requiera el usuario en su sistema, éste debe ser fácil. El diseño debe ser sencillo y amigable al usuario, el código debe ser simple y entendible, programando sólo lo necesario y lo que se utilizará.
- ✓ **Retroalimentación:** se refiere a la comunicación constante entre el desarrollador y el usuario.
- ✓ **Coraje:** se refiere a la valentía que se debe tener al modificar o eliminar el código que se realizó con tanto esfuerzo; el desarrollador debe saber cuándo el código que desarrolló no es útil en el sistema y, por lo mismo, debe ser eliminado. También se refiere a tener la persistencia para resolver los errores en la programación.

Es una metodología ágil, centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Entre las herramientas usadas en XP para manejar el proceso de desarrollo de software están las siguientes:

- ✓ **Historias de usuarios (HU):** son tarjetas físicas en las cuales se anota una descripción de una funcionalidad del sistema en una oración, se le asigna un número y un título para ser identificada.
- ✓ **Casos de prueba de aceptación:** son tarjetas que se elaboran para realizar las pruebas de cada historia de usuario.
- ✓ **Tarea de ingeniería:** son tarjetas que se elaboran para ayudar y simplificar la programación de una historia de usuario.
- ✓ **Tarjetas CRC:** describen las clases utilizadas en la programación de una historia.

Entre las ventajas de la programación extrema es que se adapta al desarrollo de sistemas pequeños y grandes; optimiza el tiempo de desarrollo; permite realizar el desarrollo del sistema en parejas para complementar los conocimientos; el código es sencillo y entendible, además de la poca documentación a elaborar durante el proceso. Entre las desventajas que tiene XP está la definición del costo y el tiempo de desarrollo; el sistema va creciendo después de cada entrega al cliente y nadie puede decir que el cliente no querrá una función más; se necesita de la presencia constante del usuario, lo que a veces es muy difícil de lograr. (33)

La metodología de software seleccionada para dar desarrollo al sistema es XP, pues al ser una metodología ágil se ajusta más a las necesidades del problema a resolver. Como se planteó anteriormente, XP es más efectiva en proyectos y equipos de desarrollo pequeños, pues facilita la disminución de la documentación enfocándose como método ágil en los procesos de comunicación directa entre las personas que intervienen en el proceso, resultando esta ser muy flexible a los cambios que se vayan presentando en el desarrollo del sistema.

1.7 Análisis de servidores web

A continuación se presentará el análisis realizado de los servidores web.

1.7.1 Internet Information Services

Internet Information Services (IIS) es un potente servidor Web que ofrece una infraestructura de gran fiabilidad, capacidad de manejo y escalabilidad para

aplicaciones Web sobre todas las versiones de Windows Server. IIS hace posible que las organizaciones aumenten la disponibilidad de sus sitios y aplicaciones Web y a la vez reducir sus costes administrativos. IIS soporta la Iniciativa de Sistemas Dinámicos de Microsoft (DSI, por sus siglas en inglés) con monitorización de estado de salud automático, aislamiento de procesos y capacidades de gestión mejoradas. (34)

Este servicio convierte a una PC en un servidor web para Internet o una Intranet; en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente. Se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas. Microsoft incluye los de Active Server Pages⁸ (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

Tabla 3 Principales características de IIS (35)

Característica o funcionalidad	Resumen
Certificados centralizados	Proporciona un único almacén de certificados SSL para una granja de servidores y simplifica la administración de enlaces SSL.
Restricciones de IP dinámicas	Permite a los administradores configurar IIS para bloquear el acceso a las direcciones IP que superen el número especificado de solicitudes y especificar el comportamiento cuando se bloquea una dirección IP.
Restricciones de intentos de inicio de sesión en FTP	Restringe el número de intentos de inicio de sesión infructuosos que se pueden realizar en una cuenta de FTP en un período de tiempo especificado.
Indicación de nombre de servidor (SNI)	Amplía los protocolos SSL y TSL para poder usar un nombre de dominio virtual, o un nombre de host, con el fin de identificar el extremo de red.
Inicialización de aplicaciones	Permite a los administradores web configurar IIS para inicializar aplicaciones web, de forma

⁸ También conocido como ASP clásico, es una tecnología de Microsoft del tipo "lado del servidor" para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Services.

	que la aplicación esté lista para la primera solicitud.
Escalabilidad habilitada para NUMA ⁹	Proporciona compatibilidad con el hardware NUMA, que permite núcleos de 32-128 CPU. Esta compatibilidad proporciona un rendimiento prácticamente óptimo desde el principio en el hardware NUMA.
Velocidad moderada de CPU de IIS	Limita el consumo de CPU, memoria y ancho de banda de un grupo de aplicaciones en una implementación multiempresa. IIS 8 incluye opciones de limitación adicionales.

El servidor web IIS de Windows proporciona una plataforma segura, fácil de administrar, modular y extensible para el hospedaje seguro de sitios web, servicios y aplicaciones. Con IIS se puede compartir información con usuarios en Internet, una intranet o una extranet. IIS es una plataforma web unificada que integra IIS, ASP.NET, servicios de FTP y PHP. (36)

1.7.2 Apache

Apache es un proyecto de código abierto y uso gratuito, multiplataforma (hay versiones para todos los sistemas operativos más importantes), muy robusto y que destaca por su seguridad y rendimiento. (37)

Es un servidor web con alta aceptación en el mundo tecnológico debido a su gran cantidad de funcionalidades, su robustez y rendimiento. Entre sus principales características están: (39)

- ✓ Trabaja en una multitud de sistemas operativos, lo que lo hace prácticamente universal.
- ✓ Apache es una tecnología gratuita de código fuente abierta. Esto le da una transparencia a este software de manera que sea una de las razones por las que es tan aceptado.

⁹ NUMA es un diseño de memoria utilizado en multiprocesamiento donde la memoria se accede en posiciones relativas de otro proceso o memoria compartida entre procesos.(38)

- ✓ Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades de este servidor. Actualmente existen muchos módulos para Apache desarrollados por colaboradores de todo el mundo.
- ✓ Trabaja con Perl, PHP y otros lenguajes de script¹⁰. Perl destaca en el mundo del script y Apache utiliza en gran parte este lenguaje. También trabaja con Java y páginas con extensión .jsp, teniendo todo el soporte que se necesita para tener páginas dinámicas.
- ✓ Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- ✓ Tiene una alta configurabilidad en la creación y gestión de logs. Permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor.

La licencia de software bajo la cual se distribuye el software es la licencia Apache, que permite la distribución de derivados de código abierto y cerrado a partir de su código fuente original. La Fundación de Software Libre no considera la licencia Apache como compatible con la versión 2 de la GNU GPL, en la cual el software licenciado bajo la licencia Apache no puede ser integrado con software distribuido bajo la GPL.

Respecto a la parte de seguridad de este servidor web, es destacable que la mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas, tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones o explotar por los usuarios locales con malas intenciones, respecto a las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache. (39)

Hasta aquí se ha presentado un resumen de dos de los servidores web más potentes de la actualidad. Tanto IIS como Apache tienen amplias potencialidades claves para el desarrollo de cualquier proyecto informático.

¹⁰ Script: archivo de órdenes, de procesamiento por lotes o guión.

Para la solución del sistema se seleccionó el servidor web Apache en su versión 2.4, ya que IIS no es multiplataforma y debe ser usado bajo licencia propietaria, mientras Apache es un proyecto libre.

Conclusiones

Después del análisis del estado del arte se concluye que:

- ✓ El proceso de análisis de las indisciplinas cometidas por estudiantes universitarios es complejo y la incorporación de las TIC en su gestión debe contribuir a su desarrollo.
- ✓ Joomla, WordPress y Drupal se encuentran entre los CMS más empleados, destacándose Drupal por su flexibilidad, usabilidad y posibilidad de desarrollo, siendo seleccionado para la implementación de la solución informática propuesta.
- ✓ MySQL y PostgreSQL se encuentran entre los sistemas gestores de bases de datos más utilizados, destacándose PostgreSQL por su potencia, rendimiento y condición de software libre, siendo seleccionado para la implementación de la solución informática propuesta.
- ✓ La metodología de desarrollo de software XP es la adecuada para guiar el desarrollo de la solución propuesta por su carácter ágil, efectividad y flexibilidad en equipos de desarrollo pequeños.
- ✓ IIS y Apache se encuentran entre los servidores web más empleados, pero destaca Apache por su condición de proyecto libre de licencias, lo que lo hace elegible para el desarrollo de la solución informática propuesta.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA INFORMÁTICO

En el presente capítulo se desarrollan las fases Exploración, Planificación y Diseño de la metodología de desarrollo de software XP, en las que se abordarán temas relacionados con las necesidades del programa y se presentarán las principales historias de usuario, el plan de duración y el plan de entrega. Se interpretarán las especificaciones del sistema quedando reflejadas en los requisitos funcionales y no funcionales y se generarán los artefactos correspondientes, concretando la cantidad de entregas y la propuesta de solución.

2.1 Propuesta de solución

Con el objetivo de contribuir al desarrollo del proceso de análisis de las indisciplinas cometidas por estudiantes de la Facultad 4 se propone la utilización de un sistema informático a partir del cual se pueda denunciar la indisciplina, emitir alertas de tiempo a las personas involucradas en el proceso, obtener reportes de datos de casos cerrados, así como facilitar planillas y documentos oficiales para emplear durante el proceso. También el sistema informático debe permitir al Decano y a los miembros de las comisiones disciplinarias listar los casos abiertos y ver su estado actual. Además, permitirá archivar denuncias y casos que no procedan, así como la opción de reabrirlos nuevamente. Se podrán guardar reportes sobre los casos cerrados para su posterior consulta y análisis.

2.2 Personas relacionadas con el sistema

Definir la audiencia a la que va dirigida la información que se gestiona en el sistema constituye uno de los pasos fundamentales para alcanzar el éxito en los objetivos propuestos. Se entiende como persona relacionada con el sistema toda aquella que esté vinculada al desarrollo o uso del software, que obtenga un resultado de valor de uno o varios procesos que existen en el sistema.

Tabla 4 Personas relacionadas con el sistema.

Personas relacionadas con el sistema	Fundamentación
--------------------------------------	----------------

Usuario anónimo	Es el usuario que navega por el sitio sin autenticarse, tiene acceso a todos los contenidos del sitio que no requieran autenticación, como el reglamento disciplinario para estudiantes del MES.
Usuario autenticado	Persona autenticada en el dominio <i>uci.cu</i> que podrán denunciar indisciplinas cometidas por estudiantes de la Facultad 4. También otras que actuarán en algún momento del análisis de la indisciplina, ellos son el denunciado, el representante de la FEU y la UJC, así como el profesor guía.
Miembro de la comisión	Persona que forma parte de la comisión disciplinaria y tiene acceso a la documentación que se va generando como resultado del análisis.
Usuario avanzado	Tiene privilegios avanzados en el sistema como crear, modificar y eliminar las comisiones disciplinarias que operarán durante el curso académico en la Facultad, funcionalidad que debe ser realizada por el Decano.
Administrador	Usuario encargado de la administración del sistema, encargado de su correcto funcionamiento, teniendo acceso a su código fuente para modificarlo si fuese necesario.

2.3 Exploración

En la fase Exploración de la metodología XP se define el alcance general del proyecto. El cliente define lo que necesita mediante la redacción de Historias de Usuarios, mediante las cuales los programadores estiman los tiempos de desarrollo. Las estimaciones realizadas en esta fase pueden estar sujetas a cambios cuando se analicen más en detalle en cada iteración. (40)

2.3.1 Aspectos funcionales y no funcionales

A continuación se presentan los aspectos funcionales del software.

Tabla 5 Requisitos funcionales

Requisitos funcionales	Requisitos asociados
Gestionar comisiones	Crear comisión
	Eliminar comisión
	Editar comisión
Gestionar casos	Crear caso
	Editar caso
	Eliminar caso
	Publicar caso
	Listar casos
	Buscar casos
Gestionar opiniones	Adicionar opinión
	Eliminar opinión
	Editar opinión
Gestionar denuncia	Realizar denuncia
	Editar denuncia
	Eliminar denuncia
	Archivar denuncia
	Publicar denuncia
	Listar denuncias
	Buscar denuncia
Ingresar declaración de denunciado	
Ingresar agravantes y atenuantes a casos	
Interrumpir caso	
Notificar medida	
Mostrar reportes	
Emitir alerta	

A continuación se presentan los aspectos no funcionales del sistema, en estos se definen las propiedades y cualidades que debe tener el sistema para su correcto comportamiento:

Interfaz, el diseño de la interfaz debe ser sencillo con pocas imágenes y animaciones que puedan perjudicar la rapidez del software. Debe ser fácil de utilizar permitiendo que

los usuarios sean capaces de interactuar con la aplicación aun teniendo conocimientos básicos de informática. Además debe tener colores serios pues se trata de un sistema cuyas funcionalidades están directamente relacionadas con la información recogida en el reglamento disciplinario del MES.

Usabilidad, el sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras. Debe contener menús que le permita a estos acceder a las funcionalidades de la aplicación de su interés siempre y cuando tengan acceso a ellas.

Portabilidad, la solución debe ser multiplataforma, de modo que los usuarios puedan usarlo en más de un sistema operativo.

Rendimiento, teniendo en cuenta que el producto se debe diseñar sobre una arquitectura cliente-servidor, los tiempos de respuesta del sistema deben ser rápidos, así como la velocidad de procesamiento de información.

Hardware, la solución informática debe ser desplegada en un servidor con los requerimientos mínimos que se presentan a continuación. Un microprocesador Dual Core o superior, 2 GB de RAM o superior, así como 100 GB de disco duro de almacenamiento.

Software, para instalar y acceder al sistema se debe contar con los siguientes requerimientos:

Máquina servidor:

- Sistema operativo: Windows 7 o superior o cualquier distribución de Linux.
- Servidor Web: Apache en su versión 2.4.4 o superior.
- Lenguaje PHP en su versión 5 o superior.
- PostgreSQL en su versión 9.2 o superior.

Máquina cliente:

- Sistema operativo: Cualquier distribución de Linux así como Windows XP o superior.
- Navegador Web: Firefox versión 4.0 o superior, Google Chrome versión 27.0 o superior, Internet Explorer versión 7.0 o superior.

Seguridad, las políticas de seguridad respecto al control de acceso serán garantizadas mediante roles, regulando así el acceso de los usuarios a la información concerniente a sus actividades y funciones de trabajo, garantizando así que cada cual acceda en el sistema solo a las funcionalidades e información que les corresponda utilizar y manejar.

2.3.2 Historias de usuario

Las historias de usuario son la técnica utilizada para especificar los requisitos del software. En estas el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. (41)

Existen varias planillas sugeridas para definir la información contenida en las historias de usuario, aunque no existe un consenso definitivo en este aspecto. Para la presente investigación se define la siguiente estructura:

- ✓ **Nombre:** Nombre identificativo de la HU.
- ✓ **Número:** Define el número asignado a la HU.
- ✓ **Usuarios:** Usuarios que tienen relación con la funcionalidad en cuestión.
- ✓ **Prioridad:** Grado de prioridad (Alta, Media o Baja) asignado por equipo de desarrollo en dependencia de las necesidades y exigencias del cliente.
- ✓ **Estimación:** Tiempo de duración para la implementación de la HU asignado por el equipo de desarrollo.
- ✓ **Complejidad:** Grado de complejidad (Alta, Media o Baja) asignada por el equipo de desarrollo luego de ser analizada.
- ✓ **Iteración:** Número de iteración en la cual debe ser implementada.
- ✓ **Descripción:** Breve descripción sobre lo que debe hacer la funcionalidad.
- ✓ **Observaciones:** Información sencilla que ayude a comprender alguna cuestión o dato no especificado anteriormente.

Tabla 6 Historia de usuario No. 1. Crear comisión

No 1	Nombre: Crear comisión
Usuarios:	Usuario Avanzado
Prioridad : Alta	Complejidad: Alta
Estimación: 0.3	Iteración 1
Descripción:	Crea una comisión disciplinaria que prestará servicio durante el transcurso del curso.
Observaciones:	

Tabla 7 Historia de usuario No.13. Realizar denuncia

No 13	Nombre: Realizar denuncia
Usuarios:	Usuario Avanzado, Miembro de Comisión
Prioridad : Alta	Complejidad: Alta
Estimación: 0.6	Iteración 2
Descripción:	Crea un formulario para realizar una denuncia.
Observaciones:	

Las restantes HU se presentan en el Anexo 1.

2.4 Planificación

La metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. (40)

En esta fase los clientes establecen la prioridad de las historias de usuario atendiendo a las necesidades inmediatas para poder organizarlas por iteraciones. Se toman acuerdos sobre el contenido de las entregas y se determina un cronograma en conjunto con el cliente. Al concluir esta fase, se podrá estimar con más criterio la duración total del proyecto.

2.4.1 Estimación de esfuerzo por historia de usuario.

Se realizó una estimación de esfuerzo por semanas a cada una de las HU definidas. El valor 1 equivale a una semana de trabajo, por lo que 0.5 equivale a dos días y medio

de trabajo y así para los restantes valores estimados. A continuación se muestra la estimación en semanas asignada a cada una de estas.

Tabla 8 Estimación de cada una de las HU.

Numero	Nombre	Estimación (semanas)
1	Crear comisión	0.3
2	Eliminar comisión	0.2
3	Editar comisión	0.1
4	Crear caso	0.5
5	Editar caso	0.2
6	Eliminar caso	0.2
7	Publicar caso	0.2
8	Listar casos	0.2
9	Buscar casos	0.2
10	Adicionar opinión	0.4
11	Eliminar opinión	0.2
12	Editar opinión	0.2
13	Realizar denuncia	0.6
14	Editar denuncia	0.5
15	Eliminar denuncia	0.2
16	Archivar denuncia	0.3
17	Publicar denuncia	0.2
18	Listar denuncias	0.2
19	Buscar denuncias	0.2
20	Ingresar declaración de denunciado	0.5
21	Ingresar agravantes y atenuantes a casos	0.5
22	Interrumpir caso	0.5
23	Notificar medida	0.5
24	Mostrar reportes	0.5
25	Emitir alertas	0.5

2.4.2 Plan de iteraciones

En cada iteración se implementa una determinada cantidad de historias de usuario arrojando como resultado una entrega del producto. El plan de iteraciones define cuántas iteraciones serán necesarias para su terminación, así como las historias de usuario que se desarrollarán en cada una. Teniendo en cuenta el análisis antes expuesto y el sistema informático que se pretende desarrollar, se realizarán tres iteraciones, las cuales se describen a continuación:

- ✓ **Iteración 1:** Para la primera iteración se tiene como principal objetivo implementar las HU de mayor prioridad para el cliente, con estas implementaciones se conforma la base de la estructura del proyecto, brindando al sistema sus funcionalidades básicas y obteniendo una primera versión de prueba.
- ✓ **Iteración 2:** En esta iteración se realizará la implementación de las restantes funcionalidades de prioridad alta así como las de prioridad media, se corrigen errores o no conformidades del cliente con las HU ya implementadas, obteniendo de esta manera la segunda versión del software.
- ✓ **Iteración 3:** En esta última iteración serán implementadas las HU de prioridad baja para el cliente y se corrigen los errores encontrados en las iteraciones anteriores.

2.4.3 Plan de duración de las iteraciones

En el plan de duración de las iteraciones se construye detalladamente el orden de desarrollo de las HU en cada iteración. Se realiza luego de tener el estimado que demora implementar cada historia de usuario. Se tiene en cuenta la prioridad que el cliente le asigna a cada historia y el nivel de complejidad que estas poseen. A continuación se describe el plan de duración de las iteraciones definido para el desarrollo del sistema.

Tabla 9 Plan de duración de las iteraciones

Iteraciones	Orden de las HU a implementar	Duración total
	✓ Crear comisión	

1	<ul style="list-style-type: none"> ✓ Eliminar comisión ✓ Editar comisión ✓ Crear caso ✓ Editar caso ✓ Eliminar caso ✓ Publicar caso ✓ Listar casos ✓ Buscar casos ✓ Adicionar opinión ✓ Eliminar opinión ✓ Editar opinión 	2.9 semanas
2	<ul style="list-style-type: none"> ✓ Realizar denuncia ✓ Editar denuncia ✓ Eliminar denuncia ✓ Archivar denuncia ✓ Publicar denuncia ✓ Listar denuncia ✓ Buscar denuncia ✓ Ingresar declaración de denunciado ✓ Ingresar atenuantes y agravantes a casos 	3.2 semanas
3	<ul style="list-style-type: none"> ✓ Interrumpir caso ✓ Notificar medida ✓ Mostrar reportes ✓ Emitir alertas 	2 semanas

2.4.4 Plan de entregas

En el plan de entregas se establece el orden y las HU que serán definidas para conformar una entrega. Este plan tiene como finalidad definir la duración de cada iteración, la cual contribuye a tener una idea aproximada de la duración de la entrega final del producto.

Este plan se elabora a partir de una reunión entre el cliente y el equipo de desarrollo en la cual se identifican las funcionalidades referentes a un mismo tema, permitiendo un mayor entendimiento en la fase de implementación.

Tabla 10 Plan de entregas

Historia de Usuario	Iteración 1	Iteración 2	Iteración 3
Crear comisión	V 1.0		
Eliminar comisión	V 1.0		
Editar comisión	V 1.0		
Crear caso	V 1.0		
Editar caso	V 1.0		
Eliminar caso	V 1.0		
Publicar caso	V 1.0		
Listar casos	V 1.0		
Buscar casos	V 1.0		
Adicionar opinión	V 1.0		
Eliminar opinión	V 1.0		
Editar opinión	V 1.0		
Realizar denuncia		V 1.0	
Editar denuncia		V 1.0	
Eliminar denuncia		V 1.0	
Archivar denuncia		V 1.0	
Publicar denuncia		V 1.0	
Listar denuncias		V 1.0	
Buscar denuncias		V 1.0	
Ingresar declaración de denunciado		V 1.0	
Ingresar atenuantes y agravantes a casos		V 1.0	
Interrumpir caso			V 1.0
Notificar medida			V 1.0
Mostrar reportes			V 1.0
Emitir alertas			V 1.0

2.5 Diseño

La metodología XP hace especial énfasis en los diseños simples y claros. Entre los conceptos más importantes de diseño en esta metodología están: (40)

Simplicidad, un diseño simple se implementa más rápido que uno complejo, por ello XP se propone implementar el diseño más simple que funcione.

Soluciones “spike”, cuando aparecen problemas técnicos, o cuando es difícil estimar el tiempo para implementar una historia de usuario, pueden usarse pequeños programas de pruebas llamados “spike¹¹” para explorar distintas soluciones.

Recodificación, consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de hacerlo más simple, conciso y entendible.

Para el diseño se utilizan esquemas como son el modelado de diagramas utilizando UML o las tarjetas Clase, Responsabilidad y Colaboración (CRC), siempre que tributen a la comprensión del proceso de desarrollo, no necesiten mucho tiempo para su realización y sea útil para el desarrollo del sistema informático.

2.5.1 Tarjetas CRC

El objetivo de las tarjetas CRC es hacer un inventario de las clases que se necesitan implementar en el sistema y la forma en que van a interactuar, de esta forma se pretende facilitar su análisis y discusión por parte de varios actores del equipo de proyecto con el objetivo de que el diseño sea lo más simple posible verificando las especificaciones del sistema. (42)

Estas trabajan con la técnica de modelado basado en objetos, representando cada una un objeto, identificando las clases y sus responsabilidades. Son un buen método para mostrar, de manera visual y clara las clases, sus roles y relaciones.

A continuación se muestran dos de las tarjetas CRC generadas para la solución, las restantes pueden ser consultadas en el Anexo 3.

Tabla 11 Tarjeta CRC para gestionar comisiones.

Gestionar comisiones	
Responsabilidades	Colaboradores
Crear comisión	node_add_page
Eliminar comisión	node_delete_confirm
Editar comisión	node_page_edit

¹¹ Spike se traduce como “punta” o “clavo” o también como “frustrar”.(40)

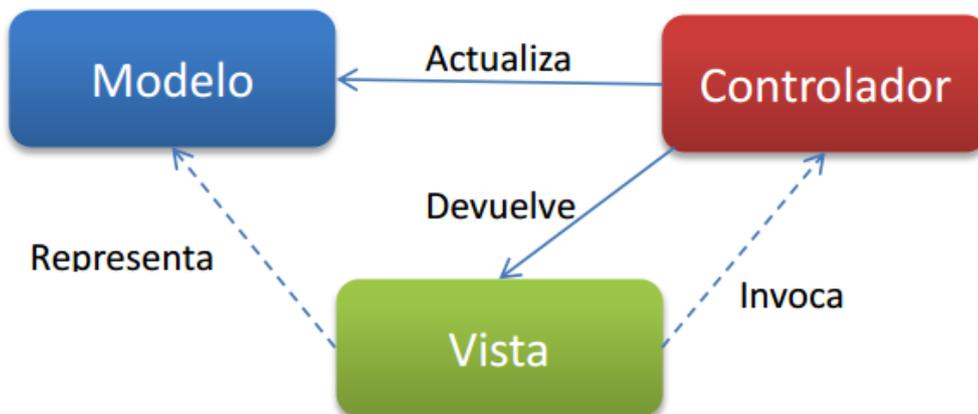
Tabla 12 Tarjeta CRC para gestionar casos.

Gestionar casos	
Responsabilidades	Colaboradores
Crear caso	node_add_page, entity, entityreference
Editar caso	node_add_page, entity, entityreference
Eliminar caso	node_delete_confirm, entity, entityreference
Publicar caso	view, Bulk Operations, Node
Listar casos	view, entity, entityreference
Buscar casos	view, entity, entityreference

2.5.2 Patrón arquitectónico

El patrón arquitectónico Modelo-Vista-Controlador (MVC) ha sido uno de los más importantes e influyentes de la historia de la programación, en la actualidad sigue siendo un patrón vigente y ampliamente utilizado. Este patrón pertenece a un conjunto de patrones agrupados en el estilo arquitectural de presentación separada. El patrón divide la capa de presentación en tres tipos de objetos básicos: modelos, vistas y controladores. La utilidad del patrón reside en que describe los flujos de comunicación entre estos tres tipos de objetos, como se muestra en la figura 6: (43)

Figura 6 Patrón arquitectónico Modelo Vista Controlador. (43)



El objetivo de este patrón es realizar y mantener la separación entre la lógica de la aplicación, los datos y la presentación. Esta separación tiene muchas ventajas, entre las que destacan: (44)

- ✓ Se puede identificar más fácilmente en qué capa se está produciendo un problema con solo saber su naturaleza.
- ✓ Se pueden crear varias presentaciones sin necesidad de escribir varias veces la misma lógica de la aplicación.
- ✓ Cada parte funciona de manera independiente y cualquier cambio centraliza el efecto sobre las demás, así que es posible estar seguros que una modificación en un componente realizará bien las tareas en cualquier parte de la aplicación.

Para mayor entendimiento de los componentes, a continuación se brinda una breve descripción de las responsabilidades de cada uno en el sistema informático:

El modelo

El modelo representa la información sobre la cual la aplicación opera, su lógica de negocios. Son un conjunto de clases encargadas de representar la información con que trabaja el usuario. (43) Este funcionamiento se logra en Drupal mediante la *capa de abstracción*, que realiza consultas a la base de datos, convirtiendo las instrucciones genéricas proporcionadas por Drupal en instrucciones particulares de la base de datos que se esté utilizando.

Las vistas

Las vistas son las encargadas de representar gráficamente el modelo y de ofrecer las acciones de los controladores para que el usuario pueda interactuar con este. (43) En Drupal las vistas son representadas por temas, que pueden ser los que trae Drupal por defecto, contribuidos, los cuales pueden ser modificados, o desarrollados. Para la solución se empleó el tema responsive_blog el cual fue modificado y adaptado a las necesidades del sistema.

Los controladores

Responden a acciones de usuario e invocan cambios en las vistas o en los modelos según sea necesario. Los controladores orquestan la interacción entre las vistas y el

modelo, recibe las peticiones del usuario, interactúa con el modelo realizando consultas y modificaciones a este, además de controlar qué vista se mostrará en respuesta y qué datos requiere para su funcionamiento.(43) Esto se logra en Drupal mediante módulos instalados, los cuales rigen el funcionamiento del sistema.

Conclusiones

El uso de la metodología de desarrollo de software XP brindó las principales herramientas para guiar la implementación del sistema. Teniendo en cuenta esta metodología, en este capítulo se definieron los cinco usuarios que interactúan con el sistema, así como se identificaron las 25 HU y la prioridad de cada una de ellas. El orden de la implementación de estas HU se reflejó en el plan de entrega en las iteraciones correspondientes. En este capítulo se presentaron los 25 requisitos funcionales definidos y entre los requisitos no funcionales del sistema se presentaron los relacionados a la seguridad, usabilidad, portabilidad, rendimiento, así como características que debe tener la interfaz del sistema informático, entre otros. Se generaron también las tarjetas CRC correspondientes a los requisitos funcionales definidos, tributando de esta forma a que el diseño sea lo más simple posible verificando las especificaciones del sistema. Se definió Modelo-Vista-Controlador como patrón arquitectónico a emplear en la propuesta de solución.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA INFORMÁTICO

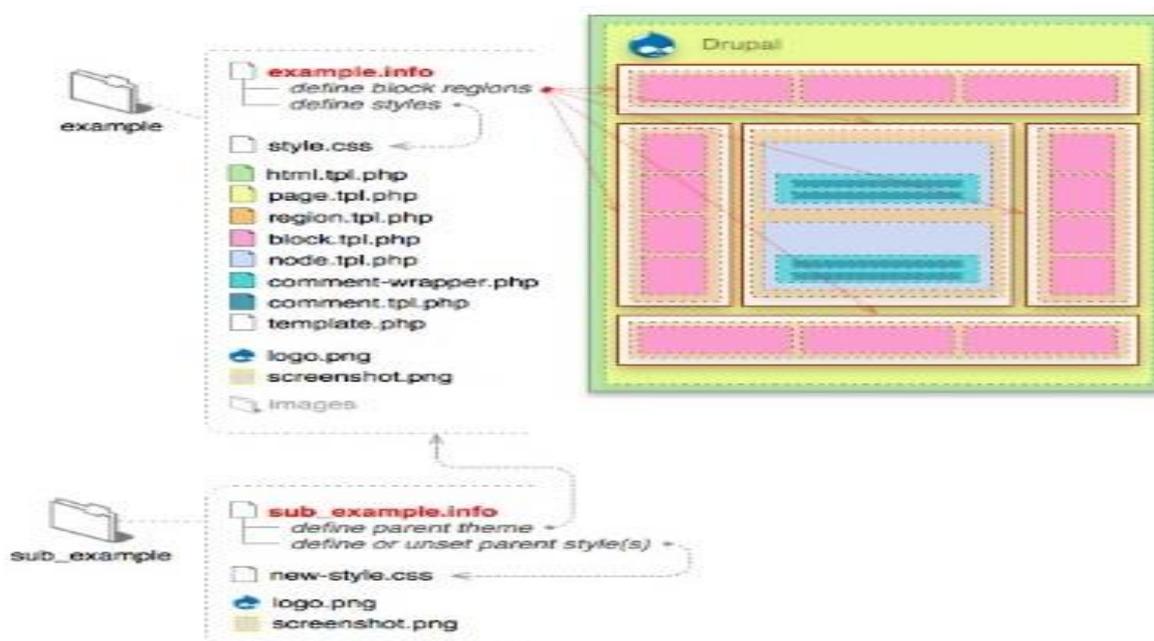
Los artefactos generados en la fase de análisis y diseño constituyen la entrada principal para el flujo de trabajo de la implementación y desarrollo, donde se describe cómo se organizan e implementan las funcionalidades definidas en el sistema. Luego de implementado el sistema se efectúan las pruebas necesarias a las funcionalidades implementadas para verificar que estén correctamente desarrolladas, dando paso a comprobar que el sistema funciona adecuadamente antes de ser usado por los usuarios finales. En este capítulo se presentan los resultados arrojados de estos procedimientos.

3.2 Temas en Drupal

Los temas permiten cambiar el aspecto gráfico o diseño del sitio en cuestión. Es posible modificar un tema de manera sencilla cambiando las hojas de estilo CSS, los archivos de plantilla o sobrescribiendo las funciones del tema. Cada plantilla o función acepta un conjunto predeterminado de variables que serán sustituidas por sus valores reales al mostrar el sitio web. (6)

Drupal trabaja con el motor de plantillas PHPTemplate por defecto pero se pueden incorporar otros motores. La figura 7 describe la estructura típica de la carpeta de un tema en Drupal.

Figura 7 Estructura de un tema en Drupal. (6)



De los archivos que contiene un tema en Drupal, solo el archivo *.info* es obligatorio. En él se incluye la definición del tema y algunas opciones de configuración, así como de archivos CSS y JavaScript. El resto de los archivos son opcionales en su instalación, sin que esto afecte el funcionamiento del sitio.

Otros de los archivos presentes en el directorio del tema son los archivos de plantilla (*.tpl.php*) que son los encargados de definir la estructura del tema, contienen código HTML y variables PHP que serán sustituidas por sus valores correspondientes, al presentar el contenido de la página, el archivo *template.php* que permite añadir lógica adicional de programación para obtener el valor final de las variables que se presentarán en las plantillas, los archivos de estilo (*style.css* y *otros ficheros .css*) que son el archivo principal que contiene los estilos y otros para una mejor organización de estos. (6)

Otros componentes son la carpeta *images*, que contendrá las imágenes del tema y los archivos *región.tpl.php* y *user-picture.tpl.php* que se encargan de la generación de la presentación de cualquiera de las regiones del tema así como la presentación de la imagen del usuario, respectivamente.

Para la propuesta de solución se usó el tema contribuido *Responsive blog*, al cual se le realizaron modificaciones para adaptarlo al sistema implementado. Este tema tiene un diseño limpio y elegante y es de tipo responsive, lo que significa que tiene la capacidad de adaptarse a dispositivos móviles.

En la UCI existe un amplio movimiento de desarrollo de aplicaciones móviles, y por ende la tendencia de la comunidad universitaria a realizar las gestiones en la web desde dispositivos móviles, se seleccionó el tema atendiendo a este criterio ya que el sistema utiliza una arquitectura cliente-servidor. De este modo la interfaz propuesta será agradable y fácil de usar para el usuario independientemente del recurso desde donde se acceda al sitio de gestión.

En la figura 8 se muestra el diseño de interfaz realizado.

Figura 6 Interfaz autenticar usuario

Sistema de análisis de las disciplinas
Facultad 4

Inicio Reglamento del MES para estudiantes universitarios Actas de comisión

Inicio de sesión

Nombre de usuario *

Contraseña *

[Crear nueva cuenta](#)
[Solicitar una nueva contraseña](#)

Iniciar sesión

Bienvenido

 Universidad de las Ciencias Informáticas

La carrera de Ingeniería en Ciencias Informáticas forma profesionales integrales, comprometidos con la patria y con el desarrollo del modelo socialista cubano, cuya función esté asociada al desarrollo de la informatización de la sociedad cubana desde tres aristas importantes: el desarrollo de la industria de software nacional, las transformaciones de procesos en las entidades para asumir su informatización y el soporte necesario para su mantenimiento.

[Leer más](#)

3.3 Implementación

Uno de los requerimientos de XP es tener al cliente disponible durante todo el proyecto. No solamente como apoyo a los desarrolladores, sino formando parte del grupo. El involucramiento del cliente es fundamental para que pueda desarrollarse un proyecto con la metodología XP. (40)

La implementación se realiza atendiendo a los estándares de codificación y de forma paralela con el diseño. Para esta fase se propone tener en cuenta aspectos muy importantes como son la disponibilidad del cliente, para lograr buenos resultados en la implementación del software.

3.3.1 Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe

reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. La mantenibilidad del código es la facilidad con que el sistema de software puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el rendimiento. Aunque la legibilidad y la mantenibilidad son el resultado de muchos factores, una faceta del desarrollo de software en la que todos los programadores influyen especialmente es en la técnica de codificación. (46)

Drupal define diferentes estándares de codificación como son: (6)

Identación, la indentación consiste en insertar espacios en blanco o tabuladores en determinadas líneas de código para facilitar su comprensión. En programación se utiliza la indentación para anidar elementos.

Etiquetas de apertura y cierre de PHP, cuando se está escribiendo en PHP, siempre se deben utilizar las etiquetas `<?php` y `?>`, y en ningún caso la versión corta `<? y ?>`.

Operadores, los operadores binarios, que se utilizan entre dos valores, deben separarse de estos valores, a ambos lados del operador, por un espacio.

Uso de comillas, Se pueden usar tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres.

Los estándares descritos anteriormente son recomendados para el desarrollo en Drupal, por esta razón se aplicaron correctamente en el código de los módulos desarrollados, como se verá más adelante en las pruebas realizadas con el módulo Coder.

3.3.1 Módulos utilizados y desarrollados en la solución

Un módulo en Drupal se compone de varias funciones escritas en PHP que permiten agregar funcionalidades al sistema. Estas funciones actúan como enganche en la construcción de la página y gestionar su contenido. La arquitectura de los módulos de Drupal es basada en ganchos (*hooks*), los cuales son la vía más factible de adicionar una nueva funcionalidad al núcleo de Drupal, llamados por este en el momento que los necesite.

Para el desarrollo de un módulo en Drupal son necesarios tres tipos de ficheros PHP de los cuales uno es de tipo *.info*, que tendrá información asociada al módulo como son la versión y el autor. Las tablas en la base de datos relacionadas con el módulo, estarán definidas en un fichero de extensión *.install*, así como un fichero *.module* que será donde se desarrollen funciones que actúen de ganchos (*hooks*) para ser llamadas por Drupal a la hora de construir la página web y administrar su contenido.

A continuación se describen algunos de los módulos empleados en el sistema, aunque fueron estudiados muchos otros para determinar cuáles serían los utilizados.

Módulos del núcleo de Drupal:

El núcleo de Drupal incorpora 44 módulos disponibles. Algunos de ellos son obligatorios para el funcionamiento de Drupal por lo que no pueden ser desactivados. Entre ellos están:

- ✓ **Filter:** Realiza acciones de filtrado sobre los contenidos.
- ✓ **Image:** Permite al sistema el manejo sobre imágenes.
- ✓ **Node:** Realiza acciones necesarias para la gestión y publicación de contenidos.
- ✓ **Field y Field SQL Storage:** Permiten conjuntamente añadir campos a entidades, tales como nodos y usuarios.
- ✓ **System:** Encargado de la administración general del sitio.
- ✓ **Taxonomy:** Permite la categorización del contenido.
- ✓ **Text:** Permite la gestión de tipos de campo de texto simple.
- ✓ **User:** Necesario para el registro, acceso y gestión de usuarios.

En la distribución de Drupal existen otros módulos que aunque forman parte del núcleo no son imprescindibles (6). Entre los que se seleccionaron para la solución se encuentran:

- ✓ **Comment:** Permite a los usuarios realizar comentarios sobre los contenidos publicados.
- ✓ **Help:** Permite que se visualice ayuda sobre los módulos en forma de mensaje.
- ✓ **List:** Permite crear campos adicionales de tipo lista de elementos, que podemos añadir a las entidades.
- ✓ **Menu:** Facilita una interfaz para el manejo de los menús en Drupal.

- ✓ **Toolbar:** Controla la barra de herramientas que muestra el menú de administración en la parte superior de la página.

Otros módulos incluidos en la solución

La comunidad de Drupal destaca por su amplio sector de desarrolladores y colaboradores a nivel mundial. Esto permite que existan numerosos módulos que faciliten diferentes funcionalidades necesarias y atractivas para cualquier proyecto. Para el desarrollo de la solución se incluyeron módulos contribuidos entre los que destacan:

- ✓ **Entity y entityreference:** Permiten crear relaciones entre entidades dígame nodos, usuarios, taxonomías, etc.
- ✓ **Rules:** Permite la creación de reglas en el sistema para acciones condicionadas.
- ✓ **View:** Permite la creación de vistas.
- ✓ **Conjunto de módulos de tipo Field (Field_collection, Field_validation):** Permiten la creación de nuevos tipos de campos.
- ✓ **Smtip:** Se usa para las gestiones de envíos de correos electrónicos.
- ✓ **LDAP:** Permite el manejo de los usuarios registrados en el dominio *uci.cu*.
- ✓ **View_data_export:** Permite la exportación de vistas a diferentes formatos.

Módulos desarrollados para la solución

Para enriquecer las funcionalidades del sistema están incluidos ciertos módulos desarrollados, los que se explican a continuación:

Operaciones_caso: Este módulo contiene un conjunto de funcionalidades para el manejo y tratamiento de los casos dentro de las que se destacan *Interrumpir_caso* y *Tomar_medida*. La primera de estas dos funcionalidades mencionadas altera el formulario del caso, dándole 30 días hábiles adicionales para la toma de decisiones respecto a un caso antes de que sea archivado. La segunda funcionalidad es un formulario el cual brinda la posibilidad de calificar la falta disciplinaria y proponer una medida a esta, archivando el caso automáticamente.

Este módulo también implementa una vista para facilitar el trabajo de las variables que se generan desde el modulo *View* de Drupal. Además, el módulo implementa

funcionalidades que mediante el consumo de servicios *LDAP* y *SOAP* maneja los datos de los usuarios de la universidad como nombre, número de solapín, género, municipio, entre otros.

A partir de los datos del usuario autenticado muestra a este como autor de las denuncias realizadas con su cuenta de dominio *uci.cu* sin necesidad de pedir nuevamente ningún elemento de este, mientras que de los usuarios demandados solo se precisa el usuario, consumiendo el resto de los datos mediante los servicios antes expuestos y alterando el formulario denuncia con dicha información.

Rules_tesis: Conjunto de reglas implementadas con el objetivo de cumplir con ciertos requerimientos del sistema. Usando el módulo *Rules* (reglas) se pueden definir acciones condicionales, que se ejecutarán cuando ocurran determinados eventos.

Se implementaron dos eventos, el evento "*Después de 30 días de actualizado el contenido*" que se ejecuta si un contenido determinado lleva más de 30 días de creado, y el evento "*Después de 25 días de actualizado el contenido*" que se ejecuta si el contenido lleva más de 25 días de creado, ambos fueron implementados con el objetivo de realizar alertas por correo electrónico para tributar a que los tiempos reglamentarios para los procesos disciplinarios se cumplan correctamente.

El evento "*Después de 30 días de actualizado el contenido*" se aplica para archivar un caso si este lleva 30 días de creado, transcurrido ese tiempo según lo estipulado en el reglamento el caso debe detenerse.

El evento "*Después de 25 días de actualizado el contenido*" tiene como función enviar una notificación de advertencia por correo electrónico a los usuarios relacionados con el caso de que el plazo reglamentario está por terminar.

Se implementó la condición "*Usuario pertenece a comisión disciplinaria*" el cual verifica al autenticarse un usuario que este es miembro de una comisión disciplinaria. Se utiliza para asignar roles a aquellos usuarios que son miembros de comisiones.

Agravante: Permite ingresar los elementos atenuantes y agravantes a un caso, así como la declaración del denunciado, validando para estas funcionalidades los permisos asociados a cada uno de los usuarios.

Altera en el proceso la vista de los formularios de casos asociándole los enlaces hacia el formulario de los aspectos atenuantes, agravantes y declaración de denunciado.

Dentro del módulo contribuido Field validation se implementaron dos nuevas validaciones:

Field_validation_persona_uci_validator, que permite comprobar que una persona pertenece al dominio uci.cu atendiendo a su usuario o solapín.

Field_validation_estudiante_Fac4_validator, que valida que los datos ingresados, en este caso usuario o solapín, pertenezcan a un estudiante de la Facultad 4. Ambas validaciones se realizan mediante el consumo de servicios *LDAP* y *SOAP*.

2.3.2 Patrones de diseño

Los patrones de diseño software son los que permiten describir fragmentos de diseño y reutilizar ideas de diseño, ayudando a beneficiarse de la experiencia de otros. Los patrones dan nombre y forma a heurísticas abstractas, reglas y buenas prácticas de técnicas orientadas a objetos. (47)

Un patrón de diseño define un esquema de refinamiento de los subsistemas o componentes dentro de un sistema, o las relaciones entre estos. Este describe una estructura común y recurrente de componentes interrelacionados, que resuelve un problema general de diseño dentro de un contexto particular. (48)

Patrones generales de software para asignar responsabilidades (GRASP)

Experto: Se encarga de asignar la responsabilidad de experto en información, la clase que cuenta con la información necesaria para cumplir la responsabilidad. Nos indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo.

El patrón experto fue usado en la clase implementada *LDAP*, ya que esta clase contiene todos los atributos y funcionalidades asociadas al consumo de servicios de *LDAP*. Está contenida dentro del fichero *lap.class.php*.

Creador: Este patrón permite identificar quién debe ser el responsable de la creación, o instanciación de nuevos objetos o clases.

En la solución propuesta, el patrón creador se pone de manifiesto en la clase *field_validation_estudiantes_fac4_validator*, de ahí se crea un objeto de la clase LDAP para acceder a un conjunto de funcionalidades de la clase LDAP que permiten validar que un estudiante pertenece a la Facultad 4.

Patrones Gang of Four (GOF)

Decorador: Añade responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia.

En el sistema implementado este patrón se pone de manifiesto en la validación implementada “*clase field_validation_estudiantes_fac4_validator*”, que hereda de la clase “*field_validation_validator*”, posibilitando así utilizar las funciones de su clase padre sin tener necesidad de redefinirlas.

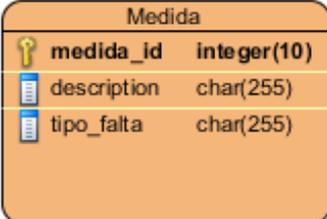
Singleton: Está diseñado para restringir la creación de objetos pertenecientes a una clase.

La clase LDAP usa este patrón ya que las funcionalidades creadas en esta clase solo son implementadas en esta clase.

3.3.2 Tablas adicionales creadas en la base de datos

Durante la instalación de Drupal se crean 74 tablas que son utilizadas para almacenar y manejar datos de contenidos, nodos y menús. Para la implementación de los módulos Operaciones_caso y Agravante se crearon las tablas Medida y Caso_extras, respectivamente. A continuación se presentan estas tablas:

Figura 7 Tabla correspondientes al módulo Operaciones_caso.



Medida	
 medida_id	integer(10)
 description	char(255)
 tipo_falta	char(255)

Figura 8 Tabla correspondientes al módulo Agravante

caso_extras	
🔑 caso_extras_id	integer(10)
📄 agravante	char(255)
📄 atenuante	char(255)
📄 declaracion	char(255)

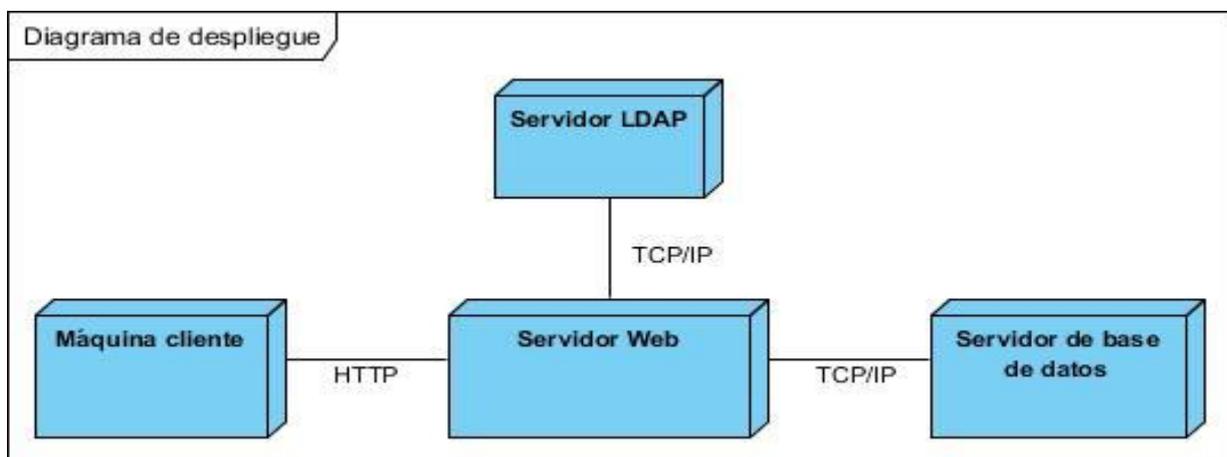
3.4 Diagrama de despliegue

El diagrama de despliegue describe las relaciones entre componentes de software y hardware en el sistema, así como visualiza como se encuentran relacionados físicamente los componentes de la aplicación. Un diagrama de despliegue muestra cómo se configuran las instancias de los componentes y los procesos para la ejecución run-time en las instancias de los nodos de proceso (algo con memoria y servicios de proceso). (49)

La metodología XP no define el modelado de este artefacto, pero se decidió la realización de este para una mejor descripción y entendimiento del problema propuesto.

El sistema informático se debe instalar en un servidor web, comunicándose con un sistema gestor de base de datos PostgreSQL. Para la autenticación de los usuarios del dominio *uci.cu* existe un servidor *LDAP* conectado a la aplicación para brindar este servicio. HTTP es el protocolo de comunicación utilizado entre el cliente y el servidor web debido a que está basado en el modelo cliente-servidor. En la figura 9 se muestra este diagrama para la solución propuesta.

Figura 9 Diagrama de despliegue



3.5 Tareas de ingeniería

Las tareas de ingeniería son una agrupación de historias de usuarios que se encuentra relacionadas formando un módulo de trabajo. Las TI también conocidas como tareas de implementación permiten a los desarrolladores obtener un nivel de detalle más avanzado que el que propician las HU, permitiéndole la representación gráfica de las responsabilidades asignadas a cada miembro del equipo de desarrollo. (50)

A continuación un ejemplo de dos de las tareas de ingeniería definidas, las restantes pueden ser consultadas en el anexo 2.

Tabla 13 Tarea de Ingeniería No.1

Tarea de Ingeniería	
No. 1	No. HU: 1
Nombre de la tarea: Crear comisión	
Tipo de tarea: Desarrollo	Estimación: 0.3
Fecha inicio: 5/3/2015	Fecha fin: 7/3/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema crea y registra una nueva comisión disciplinaria haciendo uso del tipo de contenido <i>Comisión</i> .	

Tabla 14 Tarea de Ingeniería No.2

Tarea de Ingeniería	
No. 2	No. HU: 2
Nombre de la tarea: Eliminar comisión	
Tipo de tarea: Desarrollo	Estimación: 0.2
Fecha inicio: 9/3/2015	Fecha fin: 10/04/2015
Programador responsable: Gabriel Betancourt Carrasco	

Descripción: El sistema elimina una comisión disciplinaria previamente registrada.

3.5 Pruebas

La metodología XP hace especial énfasis en el proceso de pruebas, estimulando a los desarrolladores a realizarlas mientras sea posible. Estas se clasifican atendiendo a sus funcionalidades indicando quién, cuándo y cómo deben ser ejecutadas. Mediante esta filosofía se intenta reducir lo mayor posible la cantidad de errores no detectados y el tiempo entre la introducción de estos en el sistema y su detección.

Esta metodología de desarrollo de software propone las pruebas en dos grupos, las pruebas unitarias y las pruebas de aceptación. En estas últimas el usuario juega un papel fundamental definiendo los casos de pruebas para cada HU, así como identificando los resultados esperados.

3.5.1 Pruebas unitarias

Las pruebas unitarias también se conocen como pruebas de caja blanca o pruebas modulares. Las pruebas unitarias deben ser automatizables, completas, repetibles, independientes y profesionales para garantizar su éxito en el sistema. Su principal objetivo es aislar cada parte del programa y probar su correcto funcionamiento. Estas pruebas proporcionan muchas ventajas, entre las que destacan: (51)

- ✓ Fomentan el cambio, facilitan al programador encontrar errores para así poder reprogramar el código.
- ✓ Simplifica la integración, ya que si el código funciona correctamente individualmente se facilita más adaptarlo para las pruebas de integración.
- ✓ Documenta el código, donde las propias pruebas unitarias son documentación del código ya que se puede ver cómo utilizarlo.
- ✓ Separa la interfaz y la implementación
- ✓ Los errores son más fáciles de encontrar.

Es válido aclarar que las pruebas unitarias no brindan la seguridad de que descubrirán todos los errores del código. Al probar unidades independientes por si solas, no son capaces de descubrir errores de integración, problemas de rendimiento, entre otros.

3.5.2 Módulo SimpleTest

SimpleTest es un módulo que permite usar unit test (test unitarios) en la programación de Drupal.(53) Para la realización de las pruebas unitarias al sistema implementado se utilizó este módulo, el cual somete el código a pruebas de programación para verificar su corrección.

En Drupal 7 todos los módulos del núcleo presentan un conjunto de pruebas, el módulo SimpleTest viene incorporado en el núcleo como el módulo de pruebas.

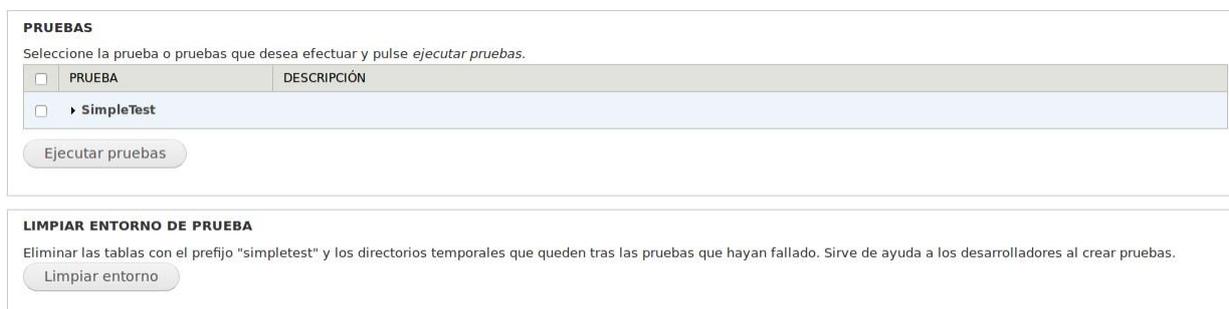
Para la realización de las pruebas primeramente se accede a la interfaz de prueba en el panel de administración de Drupal como se muestra a continuación:

Figura 10 Menú de acceso a la interfaz de prueba



Posteriormente se presenta la interfaz correspondiente al módulo SimpleTest donde se debe seleccionar a que parte del código se le realizarán las pruebas, en esta interfaz también aparece la opción *Limpiar Entorno de Prueba*, la cual hace una limpieza de los archivos temporales asociados a pruebas que hayan fallado, una buena práctica a seguir antes de realizar las pruebas.

Figura 11 Interfaz del módulo SimpleTest



Una vez seleccionadas las pruebas del módulo descrito, estas se ejecutan teniendo en cuenta que este proceso puede demorar en dependencia de la cantidad de pruebas a realizar.

La siguiente figura describe el avance de las pruebas realizadas.

Figura 12 Avance de las pruebas



Una vez concluidas las pruebas el sistema muestra los resultados obtenidos, en este caso 81 pases, 33 fallos, 46 excepciones y 24 mensajes de depuración como se muestra en la figura 13..

Posteriormente se realizó la revisión y corrección de los problemas devueltos por las pruebas.

Figura 13 Resultados de las pruebas



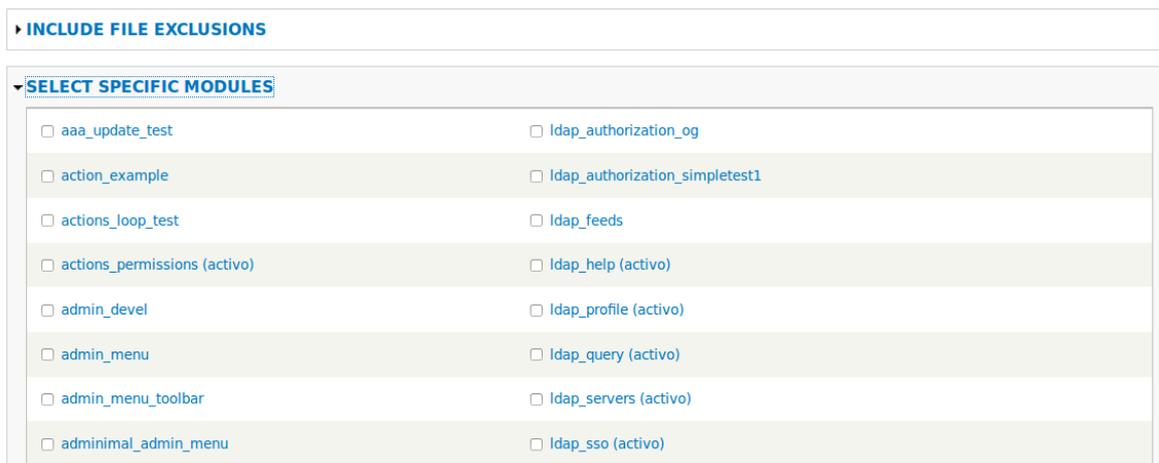
3.5.2 Módulo Coder

El módulo Coder es un módulo contribuido, que evalúa el código y señala qué parte no cumple con los estándares de codificación de Drupal, muy recomendable para mantener la coherencia del código. Permite analizar el código de cualquier archivo de Drupal, tanto del núcleo como de los módulos adicionales. Analiza la sintaxis y los comentarios del código. (6)

Se decidió usar el módulo Coder para realizar pruebas al código en busca de errores y verificando que los estándares de codificación fueron usados correctamente.

En la interfaz del módulo, aparece, entre otras opciones, la posibilidad de seleccionar los archivos sobre los que se ejecutarán las pruebas, como se muestra a continuación:

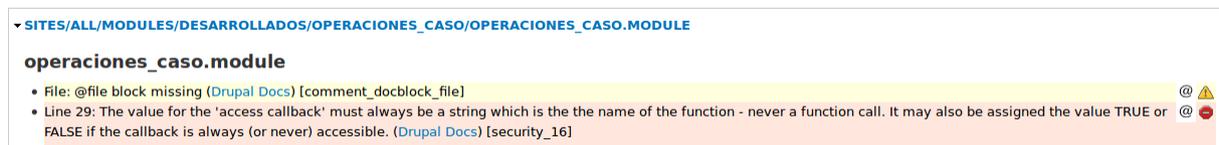
Figura 14 Interfaz del módulo Coder



A continuación se describen los resultados obtenidos de las pruebas realizadas a dos de los módulos desarrollados.

En la siguiente imagen se muestra el resultado para la prueba sobre el código del archivo *operaciones_caso.module* perteneciente al módulo *Operaciones_caso*, en la cual se encontraron irregularidades en la línea de código número 29, siendo esta corregida posteriormente.

Figura 15 Resultado de las pruebas al módulo *Operaciones_caso*



Además, se realizaron pruebas al módulo desarrollado *Agravante*, en las cuales no se encontró ninguna irregularidad, como se muestra en la siguiente figura:

Figura 16 Resultado de las pruebas al módulo *Agravante*



3.5.3 Pruebas de aceptación

Las pruebas de aceptación en XP, también llamadas pruebas del cliente son especificadas por el cliente y se centran en las características y funcionalidad generales del sistema que son visibles y revisables por parte del cliente. Las pruebas de aceptación derivan de las historias de los usuarios que se han implementado como parte de la liberación del software. (52)

Estas pruebas aseguran el comportamiento del sistema o software para poder asegurar que el sistema cumple con los objetivos trazados. También conocidas como pruebas de caja negra se definen a partir de las HU durante las respectivas iteraciones. Una HU puede definir la cantidad de pruebas de aceptación que necesite para garantizar su correcto funcionamiento. El principal objetivo de estas es garantizar que el sistema es aceptable y los requisitos han sido cumplidos satisfactoriamente.

Se realizaron las pruebas de aceptación con el cliente, arrojando estas un resultado positivo. Se realizó un acta de conformidad del cliente, que fue firmado por este, dejando constancia del éxito de las pruebas de aceptación quedando demostrado el éxito del proceso de desarrollo de la solución.

A continuación dos de las pruebas de aceptación realizadas en el sistema, las restantes se pueden consultar en el Anexo 4.

Tabla 15 Prueba de aceptación.No.1

Prueba de aceptación	
Número: 1	No. HU: 1
Nombre: Crear comisión.	
Descripción: Prueba para la funcionalidad que permite crear una comisión disciplinaria y registrarla en el sistema, haciendo uso del tipo de contenido Comisión.	
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador o usuario avanzado.	
Pasos de ejecución: El usuario selecciona la opción “Crear comisión” en el menú de comisiones. Luego la aplicación muestra los campos necesarios a llenar para crear la comisión disciplinaria.	
Resultado esperado: El sistema crea y registra una nueva comisión disciplinaria.	
Evaluación de la prueba: Satisfactoria.	

Tabla 16 Prueba de aceptación No. 2

Prueba de aceptación	
Número: 2	No. HU: 2
Nombre: Eliminar comisión.	
Descripción: Prueba para la funcionalidad que permite eliminar una comisión disciplinaria del sistema.	
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador o usuario avanzado.	
Pasos de ejecución: El usuario marca la comisión disciplinaria a eliminar y selecciona la opción "Eliminar comisión" en el menú de comisiones. Luego la aplicación elimina la comisión en cuestión del sistema.	
Resultado esperado: El sistema elimina la comisión disciplinaria seleccionada.	
Evaluación de la prueba: Satisfactoria.	

Conclusiones

En el presente capítulo se abordaron las cuestiones relacionadas con la implementación de la solución y las pruebas realizadas para garantizar su calidad. Se hizo un análisis de la arquitectura y los módulos utilizados y desarrollados del CMS utilizado para desarrollar la solución, es el caso del módulo *Operaciones_caso* y *Agravante* los cuales implementan una serie de funcionalidades muy importantes para el software. También se describieron las validaciones desarrolladas así como los eventos y condiciones implementadas para el trabajo con reglas. Se definieron los estándares de codificación y los patrones de diseño utilizados, como son el patrón Experto y el Creador de tipo GRASP así como el Decorador y el Singleton de tipo GOF. Se describieron las tablas creadas en la base de datos en la implementación de los módulos desarrollados. Se realizó el diagrama de despliegue correspondiente al diseño del sistema, así como las pruebas unitarias y de aceptación que brindan al cliente seguridad y conformidad respecto al sistema implementado. Los métodos de prueba usados permitieron validar el correcto funcionamiento del software, tanto a nivel de código como en la interacción directa con el cliente comprobándose de esta manera que el producto cumple con los objetivos propuestos.

CONCLUSIONES GENERALES

La investigación realizada permite llegar a las siguientes conclusiones:

- ✓ El análisis de las indisciplinas cometidas por estudiantes universitarios, regulado en la resolución No. 240/07 del MES es un proceso complejo que requiere la participación de varias personas e impone límites de tiempo.
- ✓ Entre los CMS más utilizados en la actualidad están WordPress, Joomla y Drupal, luego de realizar un análisis de estos sistemas gestores de contenidos se seleccionó Drupal para desarrollar el sistema informático, ya que está mejor posicionado en concepto de flexibilidad, seguridad, calidad de la documentación y posibilidad de desarrollo.
- ✓ El proceso de desarrollo de la aplicación informática propuesta fue guiado con el uso de la metodología de desarrollo de software XP, comprobando su capacidad para posibilitar obtener un producto con calidad.
- ✓ La aplicación informática para el análisis de las indisciplinas de los estudiantes de la Facultad 4, responde a las necesidades del cliente, evidenciándose esto en el resultado de las pruebas de aceptación.

RECOMENDACIONES

Para la continuidad de la presente investigación el autor recomienda:

- Incorporar nuevas funcionalidades al sistema informático propuesto como *Generar expediente* al concluir un caso.
- Hacer adaptable el contenido del sistema informático propuesto para que pueda ser visualizado en dispositivos móviles.
- Integrar el sistema informático a Akademos¹², tanto para obtener información del comportamiento académico del estudiante como para obtener alertas sobre ausencias y malas calificaciones de los estudiantes.

¹² Sistema de gestión universitaria de la UCI.

BIBLIOGRAFÍA

1. Influencia de las TIC en la gestión de la información empresarial (Julio Jiménez Martínez, y otros)
2. "Plan de estudios de la carrera Ingeniería en Ciencias Informáticas", año 2014.
3. "Resolución No. 240/07 del Ministerio de Educación Superior."
4. Acerca de: Historia de los CMS de código abierto - elWebmaster.com. [Online]. [Accedido 16 Abril 2015]. Disponible desde:
<http://www.elwebmaster.com/general/historia-cms-codigo-abierto>
5. Sistemas de Gestión de contenidos Historia y su evolución. [Online]. [Accedido 17 Abril 2015]. Disponible desde:
<http://www.cursodemarketingdigitalocc.es/sistemas-de-gestion-de-contenidos-historia-y-su-evolucion/>
6. "Aprende Drupal con Forcuntu, Experto en Drupal (2012)"
7. Mosaic | Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto. [Online]. [Accedido 14 Junio 2015]. Disponible desde:
<http://mosaic.uoc.edu/2004/11/29/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms-de-codigo-abierto/>
8. Comparativa entre WordPress, Drupal y Joomla, los tres CMS más populares. [Online]. [Accedido 13 Junio 2015]. Disponible desde:
<http://cu.globedia.com/comparativa-wordpress-drupal-joomla-cms-populares>
9. Comparativa entre Wordpress, Joomla y Drupal. | Programación y diseño web Drupal. [Online]. [Accedido 13 Junio 2015]. Disponible desde: <http://www.comunic-art.com/blog/2014/12/01/comparativa-entre-wordpress-joomla-y-drupal>
10. Qué es Joomla. [Online]. [Accedido 16 Junio 2015]. Disponible desde:
<http://www.webempresa.com/joomla.html>

- 11.** Características Principales. [Online]. [Accedido 12 Junio 2015]. Disponible desde: <http://ayuda.joomlaspanish.org/-preguntas-frecuentes-enlaces-directos-82/388-caracteristicas-principales>
- 12.** Características del gestor de contenidos WordPress | Esfera Hosting. [Online]. [Accesed 12 Junio 2015]. Available from: <http://www.esferahosting.com/wordpress>
- 13.** Características avanzadas de Drupal | Hayqueverlo.com. [Online]. [Accedido 13 Mayo 2015]. Disponible desde: <http://hayqueverlo.com/caracteristicas-drupal>
- 14.** Comparativa entre WordPress, Drupal y Joomla, los tres CMS más populares. [Online]. [Accedido 13 Junio 2015]. Disponible desde: <http://cu.globedia.com/comparativa-wordpress-drupal-joomla-cms-populares>
- 15.** Comparativa Drupal, Joomla y Wordpress | Programación Web Drupal. [Online]. [Accedido 14 Junio 2015]. Disponible desde: <http://isyourweb.com/comparativa-drupal-joomla-y-wordpress>
- 16.** Comparativa entre Wordpress, Joomla y Drupal. | Programación y diseño web Drupal. [Online]. [Accedido 10 Junio 2015]. Disponible desde: <http://www.comunic-art.com/blog/2014/12/01/comparativa-entre-wordpress-joomla-y-drupal>
- 17.** ¿Qué es PHP? y ¿Para qué sirve? Un potente lenguaje de programación para crear páginas web. [Online]. [Accedido 13 Mayo 2015]. Disponible desde: http://www.aprenderaprogramar.com/index.php?option=com_content&id=492:ique-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&Itemid=193
- 18.** Capítulo 8: Formularios (página 1). [Online]. [Accedido 15 Junio 2015]. Disponible desde: <http://www.lsi.us.es/cursos/cursoweb/cap0801.html>
- 19.** Definición de html - Qué es, Significado y Concepto. [Online]. [Accedido 13 Mayo 2015]. Disponible desde: <http://definicion.de/html/>
- 20.** CCS, Inc. - Your Source for Microchip PIC® MCU Development Tool Solutions. [Online]. [Accedido 16 Junio 2015]. Disponible desde: <http://www.ccsinfo.com/>

21. Guía Breve de CSS. [Online]. [Accedido 13 Mayo 2015]. Disponible desde: <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>
22. Estructura y reglas de CSS. [Online]. [Accedido 15 Junio 2015]. Disponible desde: <http://www.htmlhelp.com/es/reference/css/structure.html>
23. la militancia de los equivocados | very well Johnny boy. [Online]. [Accedido 14 Junio 2015]. Disponible desde: <https://lamilitanciadelosequivocados.wordpress.com/>
24. ¿Qué es MySQL? [Online]. [Accedido 13 Mayo 2015]. Disponible desde: <http://www.espestudio.com/noticias/que-es-mysql>
25. Definición de MySQL » Concepto en Definición ABC. [Online]. [Accedido 13 Mayo 2015]. Disponible desde: <http://www.definicionabc.com/tecnologia/mysql.php>
26. Ventajas y desventajas de la base de datos My SQL - Mind Map. [Online]. [Accedido 1 Junio 2015]. Disponible desde: <https://www.mindomo.com/es/mindmap/ventajas-y-desventajas-de-la-base-de-datos-my-sql-402f5022676f47919f2a15e79534330b>
27. Sobre PostgreSQL | www.postgresql.org.es. [Online]. [Accedido 12 Marzo 2015]. Disponible desde: http://www.postgresql.org.es/sobre_postgresql
28. ¿Qué es PostgreSQL? | Microbuffer. [Online]. [Accedido 13 Mayo 2015]. Disponible desde: <https://microbuffer.wordpress.com/2011/05/04/que-es-postgresql/>
29. MySQL vs. PostgreSQL. [Online]. [Accedido 13 Junio 2015]. Disponible desde: <http://www.bisente.com/documentos/mysql-postgres.html>
30. Proceso de Desarrollo Unificado (RUP). [Online]. [Accedido 13 Mayo 2015]. Disponible desde: <http://www.utvm.edu.mx/OrganoInformativo/orgJul07/RUP.htm>
31. INFORMIX-5 |: Proceso Unificado y RUP. [Online]. [Accedido 16 Junio 2015]. Disponible desde: <http://informix-5.blogspot.com/2011/08/proceso-unificado-y-up.html>
32. Por qué utilizar RUP para desarrollar aplicaciones web. [Online]. [Accedido 13 Mayo 2015]. Disponible desde: <http://www.eumed.net/libros-gratis>

- 33.** UniVerso: El Periódico de los Universitarios - No. 486. [Online].
[Accedido 13 Mayo 2015]. Disponible desde:
http://www.uv.mx/universo/486/infgral/infgral_15.html
- 34.** Internet Information Services. [Online]. [Accedido 14 Mayo 2015]. Disponible desde:
<https://www.microsoft.com/spain/windowsserver2003/technologies/webapp/iis.msp>
- 35.** Información general del servidor web (IIS). [Online]. [Accedido 13 Junio 2015].
Disponible desde: <https://technet.microsoft.com/es-es/library/hh831725.aspx>
- 36.** Información del servidor web (IIS). [Online]. [Accedido 14 Mayo 2015]. Disponible desde: <https://technet.microsoft.com/>
- 37.** ¿Qué hace un Servidor Web como Apache? Configuración | Digital Learning. [Online]. [Accedido 16 Junio 2015]. Disponible desde:
<http://www.digitalllearning.es/blog/apache-servidor-web-configuracion-apache2-conf/>
- 38.** NUMA Support (Windows). [Online]. [Accedido 17 Junio 2015]. Disponible desde:
[https://msdn.microsoft.com/en-us/library/windows/desktop/aa363804\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa363804(v=vs.85).aspx)
- 39.** Una Introducción a Apache. [Online]. [Accedido 14 Mayo 2015]. Disponible desde:
http://linux.ciberaula.com/articulo/linux_apache_intro/
- 40.** "Reglas y prácticas en eXtreme Programming" (Ing. José Joskowicz)
- 41.** "Metodologías ágiles en el desarrollo de software " (José H. Canós y otros, 2010)
- 42.** Desarrollo de software. Tarjetas CRC | Jummp. [Online]. [Accedido 22 Mayo 2015]. Disponible desde:
<https://jummp.wordpress.com/2012/01/10/desarrollo-de-software-tarjetas-crc/>
- 43.** "Guía de arquitectura N-Capas orientada al dominio con .NET 4.0" (César de la Torre Llorente, y otros, 2010)
- 44.** "Kumbia PHP Framework" (Andrés Felipe Gutiérrez)

- 45.** Drupal 7 - Unidad 43 - Arquitectura de Drupal (PDF) | Forcontu - Formación y Consultoría Drupal. [Online]. [Accedido 14 Junio 2015]. Disponible desde: <https://www.forcontu.com/>
- 46.** Revisiones de código y estándares de codificación. [Online]. [Accedido 16 Junio 2015]. Disponible desde: [https://msdn.microsoft.com/es-es/library/aa291591\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa291591(v=vs.71).aspx)
- 47.** "UML y Patrones" (Craig Larman, 2da edición)
- 48.** "Guía de Patrones, Prácticas, y Arquitectura.NET" (Ernesto Marquina y otros, 2008)
- 49.** Sparx Systems - Tutorial UML 2 - Diagrama de Despliegue. [Online]. [Accedido 17 Junio 2015]. Disponible desde: http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html
- 50.** (Ariel Erlijman Piwen, 2001)
- 51.** Pruebas Unitarias, Parte 1: Introducción y utilización de objetos simulados (Mock). [Online]. [Accedido 17 Junio 2015]. Disponible desde: <http://www.microgestion.com/index.php/mg-developers/articulos/74-unit-test-part1-mock>
- 52.** "Pruebas del sistema en Programación Extrema" (J. J. Gutiérrez, y otros)
- 53.** Drutácora: Instalando SimpleTest. [Online]. [Accedido 21 Junio 2015]. Disponible desde: <http://drutacora.blogspot.com/2010/09/instalando-simpletest.html>

ANEXOS

Anexo 1. Historias de Usuarios.

No 2		Nombre: Eliminar comisión	
Usuarios		Usuario Avanzado	
Prioridad : Alta		Complejidad: Media	
Estimación: 0.2		Iteración 1	
Descripción:		Elimina una comisión ya creada en el sistema.	
Observaciones:			

No 3		Nombre: Editar comisión	
Usuarios		Usuario Avanzado	
Prioridad : Alta		Complejidad: Media	
Estimación: 0.1		Iteración 1	
Descripción:		Modifica alguna comisión disciplinaria ya registrada en el sistema.	
Observaciones:			

No 4		Nombre: Crear caso	
Usuarios		Usuario Avanzado, Miembro de Comisión	
Prioridad : Alta		Complejidad: Alta	
Estimación: 0.5		Iteración 1	
Descripción:		Adiciona al sistema un caso para su posterior desarrollo.	
Observaciones:			

No 5		Nombre: Editar caso	
Usuarios		Usuario Avanzado, Miembro de Comisión	
Prioridad : Alta		Complejidad: Alta	
Estimación: 0.2		Iteración 1	
Descripción:		Modifica la información asociada a algún caso en desarrollo.	
Observaciones:			

No 6		Nombre: Eliminar caso
Usuarios	Usuario Avanzado	
Prioridad : Alta	Complejidad: Alta	
Estimación: 0.2	Iteración 1	
Descripción:	Elimina algún caso en desarrollo del sistema.	
Observaciones:		

No 7		Nombre: Publicar caso
Usuarios	Usuario Avanzado	
Prioridad : Alta	Complejidad: Alta	
Estimación: 0.2	Iteración 1	
Descripción:	Publica algún caso que haya sido archivado.	
Observaciones:		

No 8		Nombre: Listar casos
Usuarios	Usuario Avanzado	
Prioridad : Alta	Complejidad: Alta	
Estimación: 0.2	Iteración 1	
Descripción:	Despliega un listado de todos los casos registrados en el sistema.	
Observaciones:		

No 9		Nombre: Buscar casos
Usuarios	Usuario avanzado, Miembro de Comisión	
Prioridad : Alta	Complejidad: Alta	
Estimación: 0.2	Iteración 1	
Descripción:	Realiza una búsqueda en el sistema de algún caso en específico.	
Observaciones:		

No 10		Nombre: Adicionar opinión
Usuarios	Usuario Avanzado, Miembro de Comisión	

Prioridad : Alta	Complejidad: Alta
Estimación: 0.4	Iteración 1
Descripción:	Adiciona y registra las opiniones de las personas relacionadas con el caso, dígase representante de la FEU, testigos, etc.
Observaciones:	

No 11	Nombre: Eliminar opinión
Usuarios	Usuario Avanzado, Miembro de Comisión
Prioridad : Alta	Complejidad: Alta
Estimación: 0.2	Iteración 1
Descripción:	Elimina alguna opinión relacionada con algún caso previamente registrado en el sistema.
Observaciones:	

No 12	Nombre: Editar opinión
Usuarios	Usuario Avanzado, Miembro de Comisión
Prioridad : Alta	Complejidad: Alta
Estimación: 0.2	Iteración 1
Descripción:	Modifica alguna opinión relacionada con algún caso previamente registrado en el sistema.
Observaciones:	

No 14	Nombre: Editar denuncia
Usuarios	Usuario Avanzado, Miembro de Comisión
Prioridad : Alta	Complejidad: Alta
Estimación: 0.5	Iteración 2
Descripción:	Edita una denuncia previamente realizada.
Observaciones:	

No 15	Nombre: Eliminar denuncia
Usuarios	Usuario Avanzado, Miembro de Comisión
Prioridad : Alta	Complejidad: Alta

Estimación: 0.2	Iteración 2
Descripción:	Elimina una denuncia previamente realizada.
Observaciones:	

No 16	Nombre: Archivar denuncia
Usuarios	Todos
Prioridad : Alta	Complejidad: Alta
Estimación: 0.3	Iteración 2
Descripción:	Posibilidad de archivar una denuncia que no procede en un caso.
Observaciones:	

No 17	Nombre: Publicar denuncia
Usuarios	Todos
Prioridad : Alta	Complejidad: Alta
Estimación: 0.2	Iteración 2
Descripción:	Posibilidad de publicar una denuncia que haya sido archivada previamente.
Observaciones:	

No 18	Nombre: Listar denuncias
Usuarios	Todos
Prioridad : Alta	Complejidad: Alta
Estimación: 0.2	Iteración 2
Descripción:	Despliega un listado de las denuncias registradas en el sistema.
Observaciones:	

No 19	Nombre: Buscar denuncia
Usuarios	Usuario Avanzado, Miembro de Comisión
Prioridad : Media	Complejidad: Media
Estimación: 0.2	Iteración 2

Descripción:	Realiza una búsqueda sobre las denuncias registradas en el sistema.
Observaciones:	

No 20	Nombre: Ingresar declaración de denunciado
Usuarios	Todos
Prioridad : Alta	Complejidad: Alta
Estimación: 0.5	Iteración 2
Descripción:	Posibilidad de ingresar opcionalmente la declaración del denunciado en el caso en cuestión.
Observaciones:	

No 21	Nombre: Ingresar agravantes y atenuantes a casos
Usuarios	Todos
Prioridad : Alta	Complejidad: Alta
Estimación: 0.5	Iteración 2
Descripción:	Posibilidad de ingresar elementos atenuantes y agravantes relacionados con el caso en cuestión.
Observaciones:	

No 22	Nombre: Interrumpir caso
Usuarios	Usuario Avanzado, Miembro de Comisión
Prioridad : Media	Complejidad: Media
Estimación: 0.5	Iteración 3
Descripción:	Interrumpe el caso con la opción de reanudarlo en algún momento.
Observaciones:	

No 23	Nombre: Tomar medida
Usuarios	Usuario Avanzado, Miembro de Comisión

Prioridad : Media	Complejidad: Media
Estimación: 0.5	Iteración 3
Descripción:	Se define la medida a tomar del caso disciplinario moviendo este último a la lista de casos archivados automáticamente.
Observaciones:	

No 24	Nombre: Mostrar reportes
Usuarios	Usuario Avanzado, Miembro de Comisión
Prioridad : Media	Complejidad: Media
Estimación: 0.5	Iteración 3
Descripción:	Muestra reportes de casos archivados en el sistema.
Observaciones:	

No 25	Nombre: Emitir alertas
Usuarios	Todos
Prioridad : Baja	Complejidad: Media
Estimación: 0.5	Iteración 3
Descripción:	Emitir alertas de tiempo por correo electrónico a modo de aviso para que se cumplan con los tiempos establecidos para el proceso.
Observaciones:	

Anexo 2. Tareas de Ingeniería

Tarea de Ingeniería	
No. 3	No. HU: 3
Nombre de la tarea: Editar comisión	
Tipo de tarea: Desarrollo	Estimación: 0.1
Fecha inicio: 11/3/2015	Fecha fin: 12/3/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema edita una comisión disciplinaria previamente registrada.	

Tarea de Ingeniería	
No. 4	No. HU: 4
Nombre de la tarea: Crear caso	
Tipo de tarea: Desarrollo	Estimación: 0.5
Fecha inicio: 16/3/2015	Fecha fin: 18/3/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema le despliega una lista con las comisiones disciplinarias creadas, así como una de las denuncias registradas para que sean asignadas por el usuario y posteriormente dar paso a la creación de un caso.	

Tarea de Ingeniería	
No. 5	No. HU: 5
Nombre de la tarea: Editar caso	
Tipo de tarea: Desarrollo	Estimación: 0.2
Fecha inicio: 18/3/2015	Fecha fin: 19/3/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema modifica los campos del caso en cuestión, guardando los cambios de manera satisfactoria.	

Tarea de Ingeniería	
No. 6	No. HU: 6
Nombre de la tarea: Eliminar caso	
Tipo de tarea: Desarrollo	Estimación: 0.2
Fecha inicio: 19/3/2015	Fecha fin: 20/3/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema elimina algún caso previamente registrado.	

Tarea de Ingeniería	
No. 7	No. HU: 7
Nombre de la tarea: Publicar caso	
Tipo de tarea: Desarrollo	Estimación: 0.2
Fecha inicio: 23/3/2015	Fecha fin: 24/3/2015
Programador responsable: Gabriel Betancourt Carrasco	

Descripción: El sistema publica un caso que haya sido desplazado a la lista de casos completados, por ende archivados. Dígase también, reabrir caso.

Tarea de Ingeniería	
No. 8	No. HU: 8
Nombre de la tarea: Listar casos	
Tipo de tarea: Desarrollo	Estimación: 0.2
Fecha inicio: 24/3/2015	Fecha fin: 25/3/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema despliega los casos registrados.	

Tarea de Ingeniería	
No. 9	No. HU: 9
Nombre de la tarea: Buscar casos	
Tipo de tarea: Desarrollo	Estimación: 0.2
Fecha inicio: 25/3/2015	Fecha fin: 26/3/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: Realiza una búsqueda mediante filtros de algún caso en particular.	

Tarea de Ingeniería	
No. 10	No. HU: 10
Nombre de la tarea: Adicionar opinión	
Tipo de tarea: Desarrollo	Estimación: 0.4
Fecha inicio: 30/3/2015	Fecha fin: 1/4/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema adiciona una opinión a un caso en particular.	

Tarea de Ingeniería	
No. 11	No. HU: 11
Nombre de la tarea: Eliminar opinión	
Tipo de tarea: Desarrollo	Estimación: 0.2
Fecha inicio: 2/4/2015	Fecha fin: 3/4/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema elimina una opinión ya ingresada.	

Tarea de Ingeniería	
No. 12	No. HU: 12
Nombre de la tarea: Editar opinión	
Tipo de tarea: Desarrollo	Estimación: 0.2
Fecha inicio: 6/4/2015	Fecha fin: 7/4/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema modifica una opinión ya ingresada, guardando satisfactoriamente los cambios realizados.	

Tarea de Ingeniería	
No. 13	No. HU: 13
Nombre de la tarea: Realizar denuncia	
Tipo de tarea: Desarrollo	Estimación: 0.6
Fecha inicio: 7/4/2015	Fecha fin: 11/4/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema despliega el formulario correspondiente a la denuncia, revisa que los datos ingresados son correctos mediante la contribución del módulo LDAP, para posteriormente registrar la denuncia.	

Tarea de Ingeniería	
No. 14	No. HU: 14
Nombre de la tarea: Editar denuncia	
Tipo de tarea: Desarrollo	Estimación: 0.5
Fecha inicio: 13/4/2015	Fecha fin: 15/4/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema modifica una denuncia previamente registrada, guardando los cambios satisfactoriamente.	

Tarea de Ingeniería	
No. 15	No. HU: 8
Nombre de la tarea: Eliminar denuncia	
Tipo de tarea: Desarrollo	Estimación: 0.2
Fecha inicio: 15/4/2015	Fecha fin: 16/4/2015

Programador responsable: Gabriel Betancourt Carrasco
Descripción: El sistema elimina una denuncia previamente registrada.

Tarea de Ingeniería	
No. 16	No. HU: 16
Nombre de la tarea: Archivar denuncia	
Tipo de tarea: Desarrollo	Estimación: 0.3
Fecha inicio: 20/4/2015	Fecha fin: 22/4/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema archiva una denuncia que no se convertirá en un caso, con la posibilidad de volverse a publicar nuevamente.	

Tarea de Ingeniería	
No. 17	No. HU: 17
Nombre de la tarea: Publicar denuncia	
Tipo de tarea: Desarrollo	Estimación: 0.2
Fecha inicio: 22/4/2015	Fecha fin: 23/4/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema publica una denuncia que haya sido desplazada a la lista de denuncias archivadas.	

Tarea de Ingeniería	
No. 18	No. HU: 18
Nombre de la tarea: Listar denuncias	
Tipo de tarea: Desarrollo	Estimación: 0.2
Fecha inicio: 23/4/2015	Fecha fin: 24/04/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema despliega un listado de las denuncias registradas.	

Tarea de Ingeniería	
No. 19	No. HU: 18
Nombre de la tarea: Buscar denuncias	
Tipo de tarea: Desarrollo	Estimación: 0.2
Fecha inicio: 27/4/2015	Fecha fin: 28/04/2015

Programador responsable: Gabriel Betancourt Carrasco
Descripción: El sistema realiza una búsqueda con opciones de filtrado en las denuncias registradas.

Tarea de Ingeniería	
No. 20	No. HU: 18
Nombre de la tarea: Ingresar declaración de denunciado	
Tipo de tarea: Desarrollo	Estimación: 0.5
Fecha inicio: 28/4/2015	Fecha fin: 1/6/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema ingresa la declaración del denunciado agregándola al caso correspondiente.	

Tarea de Ingeniería	
No. 21	No. HU: 18
Nombre de la tarea: Ingresar agravantes y atenuantes a casos	
Tipo de tarea: Desarrollo	Estimación: 0.5
Fecha inicio: 4/6/2015	Fecha fin: 7/6/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema ingresa los elementos atenuantes y agravantes agregándolos al caso correspondiente.	

Tarea de Ingeniería	
No. 22	No. HU: 22
Nombre de la tarea: Interrumpir caso	
Tipo de tarea: Desarrollo	Estimación: 0.5
Fecha inicio: 7/6/2015	Fecha fin: 10/04/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema interrumpe un caso en desarrollo con la opción de reabrirlo 30 días hábiles después.	

Tarea de Ingeniería	
No. 23	No. HU: 22
Nombre de la tarea: Tomar medida	

Tipo de tarea: Desarrollo	Estimación: 0.5
Fecha inicio: 11/6/2015	Fecha fin: 13/6/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema ingresa la medida disciplinaria relacionada con el caso correspondiente, para posteriormente archivar el caso.	

Tarea de Ingeniería	
No. 24	No. HU: 22
Nombre de la tarea: Mostrar reportes	
Tipo de tarea: Desarrollo	Estimación: 0.5
Fecha inicio: 13/6/2015	Fecha fin: 15/6/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema muestra reportes basados en los datos de las denuncias archivadas, para su posterior consulta y análisis.	

Tarea de Ingeniería	
No. 22	No. HU: 22
Nombre de la tarea: Emitir alerta	
Tipo de tarea: Desarrollo	Estimación: 0.5
Fecha inicio: 18/4/2015	Fecha fin: 21/04/2015
Programador responsable: Gabriel Betancourt Carrasco	
Descripción: El sistema emite alertas por correo electrónico mediante la implementación de reglas haciendo uso del módulo <i>Rules</i> , con el objetivo de que los tiempos reglamentarios de los procesos disciplinarios se desarrollen correctamente.	

Anexo 3. Tarjetas CRC

Gestionar opiniones	
Responsabilidades	Colaboradores
Adicionar opinión	comment_permalink
Eliminar opinión	comment_confirm_delete_page
Editar opinión	comment_edit_page

Gestionar denuncia	
Responsabilidades	Colaboradores
Crear denuncia	node_add_page
Editar denuncia	node_page_edit
Eliminar denuncia	node_delete_confirm
Archivar denuncia	view, bulkoperations
Publicar denuncia	view, bulkoperations
Listar denuncia	view
Buscar denuncia	view

Ingresar declaración de denunciado	
Responsabilidades	Colaboradores
	agravante_node_view

Ingresar atenuantes y agravantes a casos	
Responsabilidades	Colaboradores
	agravante_node_view
	agravante_form
	agravante_schema

Interrumpir caso	
Responsabilidades	Colaboradores
	operaciones_caso_interrumpir_caso
	operaciones_caso_menu
	operaciones_caso_permission

Mostrar reportes	
Responsabilidades	Colaboradores
	view
	view_data_export

Tomar medida	
Responsabilidades	Colaboradores
	operaciones_caso_schema
	operaciones_caso_views_data
	operaciones_caso_medida_form

Emitir alerta	
Responsabilidades	Colaboradores
	rules_caso_creado
	rules_tesis_crom
	rules_tesis_rules_condition_info

Anexo 4. Pruebas de aceptación

Prueba de aceptación	
Número: 3	No. HU: 3
Nombre: Editar comisión.	
Descripción: Prueba para la funcionalidad que permite editar una comisión disciplinaria registrada en el sistema.	
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador o usuario avanzado.	
Pasos de ejecución: El usuario selecciona la opción "Editar comisión". Luego la aplicación le habilita la edición a los campos de la comisión disciplinaria en cuestión.	
Resultado esperado: El sistema guarda los cambios de la edición de la comisión satisfactoriamente.	
Evaluación de la prueba: Satisfactoria.	

Prueba de aceptación	
Número: 4	No. HU: 4
Nombre: Crear caso.	
Descripción: Prueba para la funcionalidad que permite crear un caso a partir de la asignación de una comisión disciplinaria y una denuncia.	

Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado o miembro de comisión.
Pasos de ejecución: El usuario selecciona la opción "Crear caso" en el menú de casos, posteriormente el sistema le despliega una lista de las comisiones disciplinarias y denuncias realizadas para a partir de la asignación de estas crear un caso.
Resultado esperado: El sistema crea un caso a partir de la comisión disciplinaria y la denuncia seleccionada.
Evaluación de la prueba: Satisfactoria.

Prueba de aceptación	
Número: 5	No. HU: 5
Nombre: Editar caso.	
Descripción: Prueba para la funcionalidad que permite editar un caso creado en el sistema.	
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado o miembro de comisión.	
Pasos de ejecución: El usuario selecciona la opción "Editar caso" facilitada en la vista correspondiente al tipo de contenido Caso.	
Resultado esperado: El sistema guarda satisfactoriamente los cambios realizados en el caso en cuestión.	
Evaluación de la prueba: Satisfactoria.	

Prueba de aceptación	
Número: 6	No. HU: 6
Nombre: Eliminar caso.	
Descripción: Prueba para la funcionalidad que permite eliminar un caso creado en el sistema.	
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado o miembro de comisión.	
Pasos de ejecución: El usuario selecciona la opción "Eliminar caso" facilitada en la vista correspondiente al tipo de contenido Caso.	
Resultado esperado: El sistema elimina satisfactoriamente el caso del sistema.	
Evaluación de la prueba: Satisfactoria.	

Prueba de aceptación	
Número: 7	No. HU: 7

Nombre: Publicar caso.
Descripción: Prueba para la funcionalidad que permite publicar un caso dígase también reabrir caso previamente archivado.
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado o miembro de comisión.
Pasos de ejecución: El usuario define el caso a publicar y selecciona la opción "Publicar" mostrada en la vista correspondiente al tipo de contenido Caso.
Resultado esperado: El sistema publica o reabre nuevamente el caso.
Evaluación de la prueba: Satisfactoria.

Prueba de aceptación	
Número: 8	No. HU: 8
Nombre: Listar casos	
Descripción: Prueba para la funcionalidad que permite desplegar un listado de los casos registrados en el sistema.	
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado, miembro de comisión o usuario autenticado.	
Pasos de ejecución: El usuario selecciona la opción "Casos" ubicada en el menú correspondiente a los casos, posteriormente el sistema despliega un listado con los casos registrados.	
Resultado esperado: El sistema muestra un listado de los casos registrados.	
Evaluación de la prueba: Satisfactoria.	

Prueba de aceptación	
Número: 9	No. HU: 9
Nombre: Buscar casos	
Descripción: Prueba para la funcionalidad que permite realizar una búsqueda entre los casos registrados mediante filtros.	
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado, usuario autenticado o miembro de comisión.	
Pasos de ejecución: El usuario usa los criterios de filtrado que brinda el sistema, para posteriormente seleccionar la opción "Buscar casos".	
Resultado esperado: El sistema devuelve los casos atendiendo a los criterios de búsqueda ingresados.	
Evaluación de la prueba: Satisfactoria.	

Prueba de aceptación	
Número: 10	No. HU: 10
Nombre: Adicionar opinión.	
Descripción: Prueba para la funcionalidad que permite adicionar una opinión a un caso.	
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado, usuario autenticado o miembro de comisión.	
Pasos de ejecución: El usuario encuentra al final de la vista correspondiente al caso un campo de texto donde ingresar su comentario. Finaliza con la opción "Aceptar".	
Resultado esperado: El sistema agrega una opinión a un caso.	
Evaluación de la prueba: Satisfactoria.	

Prueba de aceptación	
Número: 11	No. HU: 11
Nombre: Eliminar opinión.	
Descripción: Prueba para la funcionalidad que permite eliminar una opinión ya registrada en el sistema.	
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado, o miembro de comisión.	
Pasos de ejecución: El usuario elimina la opinión seleccionando la opción "Eliminar" disponible en la vista del tipo de contenido Caso.	
Resultado esperado: El sistema elimina la opinión del caso en cuestión.	
Evaluación de la prueba: Satisfactoria.	

Prueba de aceptación	
Número: 12	No. HU: 12
Nombre: Editar opinión.	
Descripción: Prueba para la funcionalidad que permite editar una opinión registrada en un caso.	
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado miembro de comisión.	
Pasos de ejecución: El usuario selecciona la opción "Editar" disponible en la vista del tipo de contenido Caso. Finaliza con la opción "Aceptar".	
Resultado esperado: El sistema modifica la opinión y guarda los cambios correctamente.	

Evaluación de la prueba: Satisfactoria.

Prueba de aceptación

Número: 13

No. HU: 13

Nombre: Realizar denuncia.

Descripción: Prueba para la funcionalidad que permite realizar una denuncia.

Condiciones de ejecución: El usuario debe tener rol de administrador, usuario avanzado, usuario autenticado o miembro de comisión.

Pasos de ejecución: El usuario selecciona la opción "Realizar denuncia" disponible en el menú de denuncias, posteriormente el sistema despliega el formulario correspondiente a la denuncia, revisando mediante funcionalidades implementadas en el módulo LDAP que los datos ingresados sean correctos. Finaliza con la opción "Aceptar".

Resultado esperado: El sistema ingresa la denuncia correctamente.

Evaluación de la prueba: Satisfactoria.

Prueba de aceptación

Número: 14

No. HU: 14

Nombre: Editar denuncia.

Descripción: Prueba para la funcionalidad que permite editar una denuncia registrada en el sistema.

Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado, usuario autenticado o miembro de comisión.

Pasos de ejecución: El usuario selecciona la opción "Editar" disponible en la vista del tipo de contenido *Denuncia*. Finaliza con la opción "Aceptar".

Resultado esperado: El sistema edita la denuncia y guarda los cambios correctamente.

Evaluación de la prueba: Satisfactoria.

Prueba de aceptación

Número: 15

No. HU: 15

Nombre: Eliminar denuncia.

Descripción: Prueba para la funcionalidad que permite eliminar una denuncia registrada en el sistema.

Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado o miembro de comisión.

Pasos de ejecución: El usuario selecciona la opción “Eliminar” disponible en la vista correspondiente al tipo de contenido *Denuncia*.

Resultado esperado: El sistema elimina la denuncia correctamente.

Evaluación de la prueba: Satisfactoria.

Prueba de aceptación

Número: 16

No. HU: 16

Nombre: Archivar denuncia.

Descripción: Prueba para la funcionalidad que permite archivar una denuncia que no procede en caso.

Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado o miembro de comisión.

Pasos de ejecución: El usuario selecciona la opción “Archivar denuncia”, disponible en la vista correspondiente al tipo de contenido *Denuncia*.

Resultado esperado: El sistema desplaza la denuncia a la lista de denuncias archivadas.

Evaluación de la prueba: Satisfactoria.

Prueba de aceptación

Número: 17

No. HU: 17

Nombre: Publicar denuncia.

Descripción: Prueba para la funcionalidad que permite publicar una denuncia previamente archivada.

Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado miembro de comisión.

Pasos de ejecución: El usuario marca la denuncia a publicar entre el listado de denuncias archivadas y selecciona la opción “Publicar denuncia” disponible en la vista correspondiente al tipo de contenido *Denuncia*.

Resultado esperado: El sistema publica la denuncia seleccionada previamente archivada.

Evaluación de la prueba: Satisfactoria.

Prueba de aceptación

Número: 18

No. HU: 18

Nombre: Listar denuncias.

Descripción: Prueba para la funcionalidad que permite listar las denuncias registradas en el sistema.
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado o miembro de comisión.
Pasos de ejecución: El usuario selecciona la opción “Denuncias” disponible en el menú de denuncias, posteriormente el sistema despliega un listado de las denuncias registradas.
Resultado esperado: El sistema lista las denuncias registradas.
Evaluación de la prueba: Satisfactoria.

Prueba de aceptación	
Número: 19	No. HU: 19
Nombre: Buscar denuncia.	
Descripción: Prueba para la funcionalidad que busca denuncias previamente registradas en el sistema.	
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado o miembro de comisión.	
Pasos de ejecución: El usuario usa los criterios de filtrado que brinda el sistema, para posteriormente seleccionar la opción “Buscar denuncia”.	
Resultado esperado: El sistema devuelve las denuncias atendiendo a los criterios de búsqueda ingresados.	
Evaluación de la prueba: Satisfactoria.	

Prueba de aceptación	
Número: 20	No. HU: 20
Nombre: Ingresar declaración de denunciado	
Descripción: Prueba para la funcionalidad que ingresa la declaración del denunciado a su caso correspondiente.	
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado, usuario autenticado o miembro de comisión.	
Pasos de ejecución: El usuario encuentra en la vista correspondiente al caso un campo de texto donde ingresar su declaración. Finaliza con la opción “Aceptar”.	
Resultado esperado: El sistema ingresa la declaración asociada al caso.	
Evaluación de la prueba: Satisfactoria.	

Prueba de aceptación	
Número: 21	No. HU: 21
Nombre: Ingresar agravantes y atenuantes	
Descripción: Prueba para la funcionalidad que ingresa los aspectos agravantes y atenuantes relacionados a un caso.	
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado o miembro de comisión.	
Pasos de ejecución: El usuario encuentra en la vista correspondiente al caso campos de texto donde ingresar los aspectos agravantes y atenuantes. Finaliza con la opción "Aceptar".	
Resultado esperado: El sistema ingresa los elementos agravantes y atenuantes.	
Evaluación de la prueba: Satisfactoria.	

Prueba de aceptación	
Número: 22	No. HU: 22
Nombre: Interrumpir caso	
Descripción: Prueba para la funcionalidad que interrumpe un caso en desarrollo.	
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador o usuario avanzado.	
Pasos de ejecución: El usuario selecciona la opción "Interrumpir caso" en la vista correspondiente al tipo de contenido Caso.	
Resultado esperado: El sistema interrumpe el caso.	
Evaluación de la prueba: Satisfactoria.	

Prueba de aceptación	
Número: 23	No. HU: 23
Nombre: Tomar medida	
Descripción: Prueba para la funcionalidad que registra la medida disciplinaria correspondiente al caso.	
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado o miembro de comisión.	
Pasos de ejecución: El usuario selecciona la opción "Tomar medida" correspondiente a la vista del tipo de contenido Caso. Posteriormente se define la medida y el caso pasa a ser archivado en el sistema.	
Resultado esperado: El sistema ingresa la medida y archiva el caso correspondiente.	
Evaluación de la prueba: Satisfactoria.	

Prueba de aceptación	
Número: 24	No. HU: 20
Nombre: Mostrar reportes	
Descripción: Prueba para la funcionalidad que muestra reportes a partir de los datos de los casos archivados.	
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador, usuario avanzado, usuario autenticado o miembro de comisión.	
Pasos de ejecución: El usuario selecciona la opción "Mostrar reportes" disponibles en el menú principal, con la opción de exportarlo a formato Excel.	
Resultado esperado: El sistema muestra los reportes de casos cerrados y archivados.	
Evaluación de la prueba: Satisfactoria.	

Prueba de aceptación	
Número: 25	No. HU: 20
Nombre: Emitir alerta	
Descripción: Prueba para la funcionalidad que emite las alertas por correo electrónico a personas relacionadas con algún caso.	
Condiciones de ejecución: El usuario autenticado debe tener rol de administrador o usuario avanzado.	
Pasos de ejecución: El usuario recibe un correo electrónico de aviso atendiendo a los criterios valorados para llevar a cabo las alertas implementados en el sistema	
Resultado esperado: El sistema emite las alertas por correo electrónico.	
Evaluación de la prueba: Satisfactoria.	