

Sistema de gestión del estado de la militancia de la Unión de Jóvenes Comunistas en la Universidad de Ciencias Informáticas

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

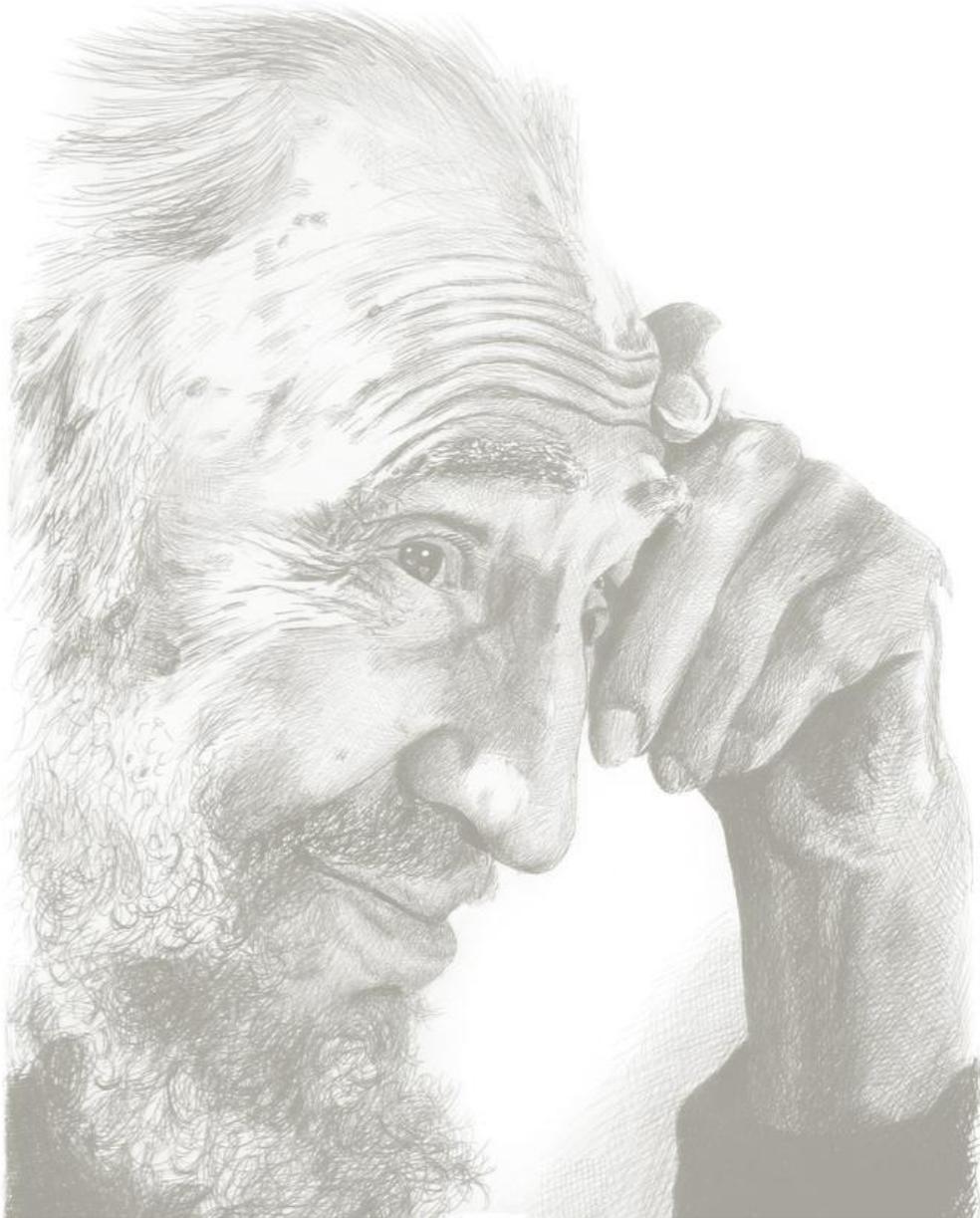
Autor: Erick Cabrera Quintana

Tutores: MSc. Miguel Jaeger Rodriguez Lazo

Ing. Ramón Paumier Samón

La Habana, curso 2013-2014

Frase



"Crear en la juventud es ver en ellos la generación del mañana, una generación mejor que nuestra propia generación, una generación con muchas más virtudes y muchos menos defectos que las virtudes y los defectos de nuestra propia generación."

Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 1 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año 2014.

Autor: Erick Cabrera Quintana

Tutor: MSc. Miguel Jaeger Rodriguez Lazo

Tutor: Ing. Ramón Paumier Samón

Datos de contacto

Miguel Jaeger Rodriguez Lazo

Graduado de Ingeniería en Ciencias Informáticas (2007) en la Universidad de las Ciencias Informáticas (UCI).

Profesor Asistente y Máster en Informática Aplicada (Universidad de las Ciencias Informáticas, 2013).

Actualmente se desempeña como profesor y jefe de la asignatura de Programación V; además ejerce como líder de proyecto en el Centro de IdeoInformática.

jaeger@uci.cu

Ramón Paumier Samón

Graduado de Ingeniería en Ciencias Informáticas (2007) en la Universidad de las Ciencias Informáticas (UCI).

Profesor Asistente.

Imparte la asignatura de Programación I.

rpaumier@uci.cu

Agradecimientos

A todas las personas que de una manera u otra colaboraron con la realización de este trabajo, que me apoyaron en todo momento y supieron darme fuerzas para seguir adelante y culminar esta etapa tan importante de mi vida. A quienes saben ser amigos, a los que son como familia.

Resumen

Se presenta el sistema para la gestión del estado de la militancia de la Unión de Jóvenes Comunistas en la Universidad de las Ciencias Informáticas. El proceso de desarrollo estuvo guiado por la metodología de desarrollo OpenUP, se utilizó como marco de trabajo el Sistema de Administración de Contenidos Drupal en su versión 7.26, el cual acotó los lenguajes de programación empleados. El gestor de bases de datos seleccionado fue PostgreSQL en su versión 9.1. La implementación del sistema viabiliza el control y seguimiento de los militantes dentro y fuera de la Universidad. El principal impacto del trabajo es de índole social pues permitirá humanizar el trabajo realizado por el Departamento de Documentación y Control de la organización en esta casa de altos estudios.

Palabras clave: Drupal, Gestión de información, Sistema de Gestión de Contenidos, Unión de Jóvenes Comunistas.

Índice general

INTRODUCCIÓN	1
CAPÍTULO 1: SISTEMAS DE GESTIÓN, TENDENCIAS Y TECNOLOGÍAS ACTUALES.	8
1.1. ESTUDIO DE SISTEMAS HOMÓLOGOS	8
1.2. TECNOLOGÍAS DE DESARROLLO, HERRAMIENTAS Y LENGUAJES A UTILIZAR	11
CAPÍTULO 2: CARACTERIZACIÓN DEL SISTEMA DE GESTIÓN DE INFORMACIÓN DE LOS MILITANTES.	26
2.1. PROPUESTA DEL SISTEMA DE GESTIÓN DE LA MILITANCIA EN LA UJC DE LA UNIVERSIDAD.....	26
2.2. MODELO DE DOMINIO	26
2.3. REQUERIMIENTOS DEL SISTEMA	28
2.4. ARQUITECTURA DEL SISTEMA.....	37
2.5. PATRONES DE DISEÑO DE DRUPAL	38
2.6. DISEÑO DEL SISTEMA	40
2.7. MODELO DE DESPLIEGUE	45
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DEL SISTEMA DE GESTIÓN DE LA MILITANCIA EN LA UJC DE LA UCI.....	47
3.1. DIAGRAMA DE COMPONENTES	47
3.2. ESTÁNDARES DE CODIFICACIÓN	48
3.3. MODELO DE DATOS ENTIDAD-RELACIÓN	49
3.4. PRUEBAS AL SISTEMA	50
CONCLUSIONES.....	56
RECOMENDACIONES.....	57
REFERENCIAS BIBLIOGRÁFICAS	58
BIBLIOGRAFÍA	61
ANEXOS	65

Índice de tablas

TABLA 1 MÉTRICAS APLICADAS A LOS SISTEMAS HOMÓLOGOS	11
TABLA 2 REQUISITOS FUNCIONALES	28
TABLA 3 DESCRIPCIÓN DE REQUISITOS. ADICIONAR MILITANTE.	31
TABLA 4 DESCRIPCIÓN DE REQUISITOS. MODIFICAR MILITANTE.....	33
TABLA 5 DESCRIPCIÓN DE REQUISITOS. ELIMINAR MILITANTE.	34
TABLA 6 DESCRIPCIÓN DE REQUISITOS. BUSCAR MILITANTE.....	35
TABLA 7 DESCRIPCIÓN DE REQUISITOS. LISTAR DATOS DEL MILITANTE.	35
TABLA 8 RESULTADOS DE LAS PRUEBAS DE SEGURIDAD.....	52
TABLA 9 DESCRIPCIÓN DE REQUISITOS. ADICIONAR COMITÉ PRIMARIO.	66
TABLA 10 DESCRIPCIÓN DE REQUISITOS. ELIMINAR COMITÉ PRIMARIO.	67
TABLA 11 DESCRIPCIÓN DE LA TABLA NODE.	72
TABLA 12 DESCRIPCIÓN DE LA TABLA NODE_TYPE.....	73
TABLA 13 DESCRIPCIÓN DE LA TABLA USERS.....	74
TABLA 14 DESCRIPCIÓN DE LA TABLA ROLE.	75
TABLA 15 DESCRIPCIÓN DE LA TABLA FIELD_DATA_FIELD_COMIT_PRIMARIO.	76
TABLA 16 DESCRIPCIÓN DE LA TABLA FIELD_DATA_FIELD_A_O.....	76
TABLA 17 DESCRIPCIÓN DE LA TABLA FIELD_DATA_TIPO.	77
TABLA 18 PRUEBAS DE AUTORIZACIÓN.....	78
TABLA 19 PRUEBAS DE GESTIÓN DE SESIONES.	78
TABLA 20 COMPROBACIÓN DEL SISTEMA DE AUTENTICACIÓN.	79
TABLA 21 VALIDACIÓN DE DATOS.....	81

Índice de figuras

FIGURA 1 MODELO DE DOMINIO.	27
FIGURA 2 ARQUITECTURA EN CAPAS DE DRUPAL.....	37
FIGURA 3 DIAGRAMA DE CLASES DEL DISEÑO DE DRUPAL.	41
FIGURA 4 DIAGRAMA DE CLASES DEL DISEÑO. PAQUETE GESTIONAR MILITANTE.....	43
FIGURA 5 DIAGRAMA DE SECUENCIA. RF ADICIONAR MILITANTE.....	45
FIGURA 6 DIAGRAMA DE SECUENCIA. RF MODIFICAR MILITANTE.	45
FIGURA 7 DIAGRAMA DE SECUENCIA. RF ELIMINAR MILITANTE.....	45
FIGURA 8 MODELO DE DESPLIEGUE.....	46
FIGURA 9 DIAGRAMA DE COMPONENTES.	47
FIGURA 10 DIAGRAMA ENTIDAD-RELACIÓN.	49
FIGURA 11 RESULTADOS DE LAS PRUEBAS FUNCIONALES.	50
FIGURA 12 REPORTE RESUMEN GENERADO POR LA APLICACIÓN JMETER.	54
FIGURA 13 DIAGRAMA DE CLASES DE DISEÑO. PAQUETE GESTIONAR COMITÉ DE BASE.	68
FIGURA 14 DIAGRAMA DE CLASES DE DISEÑO. PAQUETE GESTIONAR COMITÉ PRIMARIO.	69
FIGURA 15 DIAGRAMA DE SECUENCIA. RF ADICIONAR COMITÉ DE BASE.	69
FIGURA 16 DIAGRAMA DE SECUENCIA. RF MODIFICAR COMITÉ DE BASE.	70
FIGURA 17 DIAGRAMA DE SECUENCIA. RF ELIMINAR COMITÉ DE BASE.....	70
FIGURA 18 DIAGRAMA DE SECUENCIA. RF ADICIONAR COMITÉ PRIMARIO.....	71
FIGURA 19 DIAGRAMA DE SECUENCIA. RF MODIFICAR COMITÉ PRIMARIO.	71
FIGURA 20 DIAGRAMA DE SECUENCIA. RF ELIMINAR COMITÉ PRIMARIO.	72

Introducción

En las últimas décadas la industria de la informática y las comunicaciones ha escalado peldaños en la sociedad, adentrándose en los más variados escenarios como consecuencia de las crecientes necesidades de las empresas y organizaciones. Los sistemas de gestión automatizada han contribuido en gran parte a ese avance, agilizando el trabajo y permitiendo obtener mayores resultados en las actividades productivas. Además de las ganancias en cuanto a eficiencia, los sistemas de gestión automatizada de la información garantizan la confidencialidad, la integridad y la disponibilidad de la información que se maneja.

Nuestro país no está exento del desarrollo informático alcanzado en el mundo y complementando el nuevo programa de desarrollo trazado por la Revolución cubana, destinado a elevar el nivel cultural de la población y su calidad de vida, tomando como referencia modelos eficientes de infraestructuras productivas, llevadas a cabo en universidades visitadas por el comandante Fidel Castro, el 23 de septiembre del año 2002 abre sus puertas la Universidad de las Ciencias Informáticas (UCI), con el firme propósito de informatizar el país, a la par de crear una industria de software capaz de comercializar en el mercado internacional. Sobre el objetivo del centro Fidel Castro Díaz-Balart planteó:

“El propósito fundamental es lograr un centro de excelencia para la formación masiva de profesionales de nivel superior. Ello debe alcanzarse con la ejecución de ambiciosos programas curriculares y de producción y con la aplicación de las más modernas tecnologías en la docencia” (CASTRO DIAZ-BALART, 2004).

Desde sus inicios ha formado capital humano especializado, encargado de la investigación, producción de software y servicios informáticos para la sociedad cubana y para el mundo. Existen diversas organizaciones de masas que agrupan a quienes estudian o laboran en este centro de altos estudios, entre la que se destaca la Unión de Jóvenes Comunistas (UJC), que ha sido la cantera formadora de los revolucionarios en nuestro país desde su creación; agrupando en sus filas a estudiantes, trabajadores y demás jóvenes que comparten nuestro ideal revolucionario y de lucha antiimperialista.

La UJC la forman más de 500.000 militantes (REVÉ, 2009), agrupados en comités de base, estos conforman comités primarios y cuentan con un Comité nacional, un Buró nacional y un Congreso que se reúne cada cuatro años para tratar estatutos y reglamentos internos de la organización. El centro ha apoyado a la organización y sus tareas en todo momento, proyectando su cumplimiento en disímiles ocasiones. Un ejemplo de ello lo constituyó, sin

Introducción

dudas, el desfile popular del primero de mayo del 2009, en el que nuestro Comandante en Jefe expresara en una Reflexión publicada en la prensa nacional:

“(...) los colores rojo, azul y blanco de nuestra bandera, ondeada por las manos laboriosas de miles de jóvenes de la Universidad de las Ciencias Informáticas que cerraban el desfile(...)” (CASTRO RUZ, 2009).

La UCI cuenta con más de 3000 militantes organizados en los distintos comités de base. El Departamento de Documentación y Control de la UJC del centro es el encargado de registrar todos los movimientos de los militantes y su estado dentro de la organización. Realiza además los procesos de incorporación y traslado desde y hasta los municipios, provincias, exterior u organismos tales como el Ministerio del Interior de la República de Cuba (MININT) y las Fuerzas Armadas Revolucionarias (FAR); descuenta del estado de la militancia a los jóvenes separados, expulsados o desactivados, así como a los que pasan a formar parte de las filas del Partido Comunista de Cuba (PCC).

Este Departamento se encarga del cómputo mensual de la cantidad de militantes por cada uno de los comités de base, por facultades y los totales a nivel de centro, generando reportes en los que se muestran el estado de la militancia por comités, los cambios realizados en los mismos, así como los traslados o desactivaciones, controlando con ello los cambios realizados dentro de la organización. La información generada es utilizada por las instancias municipales y provinciales de la UJC para la planificación y control de las estadísticas, a la vez que es manejada por el centro para valorar, mes tras mes, aspectos del funcionamiento de la organización juvenil, como lo son la asistencia a las reuniones ordinarias y el pago de la cotización.

La UJC Nacional encargó al Centro de Tecnologías de Gestión de Datos (DATEC) de la Facultad 6, la creación de un sistema para el control del estado de la militancia, para ser desplegado por todo el país; sin embargo, los reportes generados no satisfacen las necesidades de la UJC en la UCI, pues está enfocado a instancias municipales, provinciales y nacionales: almacena datos de los militantes como: misiones internacionalistas, escuelas administrativas y políticas cursadas, clasificador ocupacional, entre otros; en el tratamiento de las altas por ajuste y en los traslados que se realizan, no aparecen algunos datos de interés para la UJC en la UCI, ejemplo de ello es el traslado para provincia, donde falta el comité de base y comité primario que traslada. El sistema consta de informaciones muy generales, datos de interés a niveles superiores, que distan del trabajo diario en el Departamento de Documentación y Control del centro. En el 2009, tratando de darle solución a estas deficiencias, se desarrolló otro sistema para automatizar el proceso de gestión de la información, el Sistema Integral de Gestión de la UJC UCI: Módulo Militante, sin embargo, no

Introducción

cumplió con las especificaciones de los trabajadores, presentando diversos problemas de funcionalidad, ejemplo de ello es la imposibilidad del sistema para gestionar los cambios realizados en la información sin el uso de otras aplicaciones para manejar hojas de cálculo.

En la versión existente implementada por DATEC y sin perspectivas de desarrollo, el sistema solo permite ingresar los militantes e impide todo tipo de traslado a menos que sea eliminado, lo que constituye una limitación del sistema. La obtención de datos que se necesitan a diario, como la cantidad de comités de base, el número de militantes, el estado actualizado de la militancia en la UCI, el número de traslados realizados, así como de incorporaciones, no es posible. La labor se realiza en la actualidad de manera manual y con la ayuda de programas destinados a la gestión de información mediante hojas de cálculo, lo cual dificulta el trabajo del personal de la oficina y comprometen las estadísticas reales de cada comité de base.

Teniendo en cuenta esta necesidad social identificada se plantea como **problema de la investigación**: ¿Cómo facilitar el control y seguimiento del estado de la militancia en el Departamento de Documentación y Control de la UJC en la Universidad de las Ciencias Informáticas?

Teniendo como **objeto de estudio** los procesos de gestión de la información, siendo el **campo de acción** el proceso de gestión de la información de los militantes de la UJC en la UCI.

Se define como **objetivo general**, para dar solución al problema científico:

Diseñar un sistema de gestión del estado de la militancia en la Unión de Jóvenes Comunistas dentro de la Universidad de Ciencias Informáticas, que permita el almacenamiento y tratamiento de la información sobre los militantes del centro, así como la generación de los reportes requeridos por la organización.

Del objetivo general se derivan los siguientes **objetivos específicos**:

1. Caracterizar las limitaciones y ventajas de software existentes con propósitos similares en el ámbito nacional e internacional, e identificar las principales tendencias.
2. Determinar las tecnologías, las herramientas y la metodología para la implementación del sistema para la gestión del estado de la militancia de la UJC en la UCI.
3. Diseñar las funcionalidades del sistema para la gestión del estado de la militancia de la UJC en la UCI.
4. Implementar las funcionalidades del sistema para la gestión del estado de la

Introducción

militancia de la UJC en la UCI.

5. Validar el sistema obtenido mediante pruebas al código, a la interfaz y a sus funcionalidades.

Para cumplir los objetivos y dar solución a la situación problemática planteada, se presentan las **tareas de la investigación** siguientes:

1. Estudiar las tendencias en el desarrollo de sistemas homólogos.
2. Seleccionar las tecnologías, herramientas y estándares que se necesitan para implementar la propuesta de solución.
3. Seleccionar la metodología de desarrollo de software.
4. Definir los requisitos funcionales y no funcionales de la propuesta de solución.
5. Implementar la propuesta de solución.
6. Realizar y documentar las pruebas funcionales, de aceptación y de seguridad.

Los **principales resultados** a obtener se desglosan en:

- Gestionar el estado de la militancia de la Universidad de las Ciencias Informáticas.
- Generar reportes personalizados por la organización.
- Generar las plantillas de cada uno de los comités de base.
- Generar el cómputo mensual de militantes por comité de base, por comité primario y a nivel de la UCI.

Para regir la investigación se plantea la siguiente **idea a defender**:

La implementación de un nuevo sistema de gestión automatizada en el Departamento de Documentación y Control de la UJC en la UCI, contribuirá a una mejor gestión de los militantes, permitiendo la generación de reportes para el control de las incorporaciones y los traslados realizados dentro de la Organización.

Diseño metodológico de la investigación

Dentro de los **métodos teóricos**:

Analítico-Sintético:

Permitió el estudio y revisión de los documentos relacionados con la gestión y el trabajo interno de la UJC en la UCI, para tener un mejor entendimiento de la situación problemática; descomponer el problema de investigación y profundizar en cada uno de sus elementos, que luego se concretan en la solución.

Histórico-Lógico:

Para el estudio crítico de los trabajos precedentes, que constituyen referencias en el desarrollo de soluciones para el problema y tomarlos como base de comparación con los resultados obtenidos. Permite determinar los más factibles para el desarrollo del sistema.

Modelación:

Para el desarrollo de modelos y diagramas que posibiliten un mayor entendimiento de las reglas del negocio y demás factores en el problema científico.

Dentro de los **métodos empíricos:**

Entrevista:

Posibilitó conocer las necesidades del cliente, permitiendo obtener los requisitos del sistema. Se realizó al Jefe del Departamento de Documentación y Control de la UJC y se encuentra anexada al trabajo de diploma.

Observación:

Posibilitó la obtención de las características de los procesos que tienen lugar en el Departamento de Documentación y Control de la UJC, captar los requisitos del sistema y abundar en la información necesaria para el desarrollo del mismo, la guía de observación se encuentra anexada al trabajo de diploma.

La presente investigación consta de Introducción, tres capítulos, Conclusiones, Recomendaciones, Referencias Bibliográficas, Bibliografía y Anexos. Los elementos abordados en cada uno de los capítulos se describen a continuación:

Capítulo 1: Sistemas de gestión, tendencias y tecnologías actuales.

Se define el marco teórico conceptual, realizando un análisis de los principales conceptos y tecnologías que pueden ser apropiados para el desarrollo del sistema, llevando a cabo un estudio de las herramientas más utilizadas en el diseño e implementación de los sistemas de gestión, señalando la importancia del uso de herramientas para el desarrollo web y gestores de base de datos. Se realiza un estudio crítico de las soluciones existentes a nivel nacional e internacional, como punto de partida para el desarrollo de la solución al problema. El capítulo termina explicando las herramientas escogidas y las causas que llevaron a esa elección.

Capítulo 2: Caracterización del sistema de gestión de información de los militantes.

Se hace una descripción de las reglas del negocio que serán objeto de automatización; se procede con el análisis y diseño de la solución propuesta; se describen los requisitos funcionales y no funcionales del sistema a desarrollar.

Capítulo 3: Implementación y prueba del Sistema de Gestión de la Militancia en la UJC de la UCI.

Se describe el diagrama de componentes del sistema, son reflejados aspectos esenciales en el desarrollo de la solución, los estándares de codificación y diagrama de componentes. Las pruebas realizadas al sistema son diseñadas y registrados sus resultados con el propósito de validar la solución desarrollada.

Control semántico

Con el propósito de lograr una mejor comprensión de los temas a tratar, a continuación son definidos algunos conceptos:

Gestión

Acción y efecto de administrar; planificación, organización, dirección y control de los recursos de una organización. Entre los diferentes tipos de gestión se encuentran: la ambiental, la tecnológica, la pública, de proyecto, de conocimiento, de la información, etc. La gestión tiene el fin de obtener el máximo beneficio posible; este beneficio puede ser económico o social, dependiendo esto de los fines que persiga la organización (GISBERT y GROSS, 1974).

Sistemas de información

Han existido diversos conceptos acerca de los sistemas de información, basándose en los objetivos y las necesidades que satisfacen. En un principio eran independientes de la informática, pues no existía un auge de la misma que posibilitara su uso en esta área; constituían formas de simplificar el trabajo en una empresa. El desarrollo paulatino alcanzado por esta ingeniería, dio paso a la implementación de dichos sistemas, utilizando esas nuevas tecnologías, lo que permitió un mejor desempeño de esas empresas en relación a sus competidores. La informática y las telecomunicaciones impulsaron el desarrollo de los sistemas de información, ahorrando tiempo en la realización de actividades, permitiendo almacenar mayor información en menor espacio. En la actualidad, y atendiendo al campo de la informática, es tratado como un conjunto de elementos relacionados entre sí, con el propósito de apoyar y mejorar las actividades de una empresa, haciendo uso de sistemas computacionales que aseguren eficiencia y posibiliten obtener, almacenar, administrar y

transmitir datos; incluye además el factor humano que utiliza el sistema (ANDREU et al., 1991).

Gestión de la información

Se refiere al proceso para la obtención y utilidad de los recursos que manejan la información en un contexto dado. Responsable del ciclo de vida de la misma, pretendiendo garantizar su disponibilidad, integridad y confidencialidad. Según Gloria Ponjuán:

“...es todo lo relacionado con la obtención de información adecuada, en la forma correcta, para la persona indicada, al costo adecuado, en el tiempo oportuno, en el lugar apropiado, para tomar la acción correcta...” (PONJUÁN DANTE, 2004).

Capítulo 1: Sistemas de gestión, tendencias y tecnologías actuales.

En el presente capítulo se expone la fundamentación teórica de la investigación: se incluyen los elementos homólogos a nivel nacional e internacional, se caracterizan las principales herramientas utilizadas en la solución propuesta, así como la metodología empleada y su correspondiente justificación.

1.1. Estudio de sistemas homólogos

En el ámbito internacional no existen sistemas destinados a la gestión de la información de la UJC puesto que es una organización que solo radica en Cuba, aunque sí se encuentran algunas soluciones con características similares a la misma en cuanto al manejo de la información y gestión de reportes.

1.1.1. Sistemas de gestión analizados a nivel internacional:

1.1.1.1. DocCF: Software de Gestión Escolar

Constituye un sistema para automatizar los procesos administrativos, escolares y académicos de instituciones educativas. Desarrollado por Grupo CF *Developer* radicado en Colombia, es encargado de la distribución y el soporte del sistema en Latinoamérica y España a través de su sitio web. El sistema actual, en una versión 2.6, presenta mejoras en cuanto al uso de roles para autenticar usuarios, corrección de errores, consultar datos históricos, gestión escolar web, restauración de datos, mejora de carga y procesamiento de informes, opciones de visualización, registros de pagos, entre otros (AGUDELO RESTREPO y CARMONA RUIZ, 2013; GRUPO CF DEVELOPER, 2013; VELASQUEZ y ROMAN, 2009). Aunque presentan similitudes con las operaciones a desarrollar en el Departamento de Documentación y Control de la UJC, no satisface las necesidades de la organización por estar encaminada a la administración de centros educativos, ser un software propietario e imposibilitar la reestructuración del mismo para adecuarlo a la situación de la UJC en la Universidad.

1.1.1.2. Software *Entropy*

Es un sistema desarrollado por BSI *Action Manager* con el propósito de mejorar el desempeño empresarial de una entidad, mantener la sustentabilidad y reducir los riesgos. Una aplicación web conformada por cinco módulos principales: gestión de auditorías y cumplimiento, gestión de incidentes, gestión del desempeño, gestión del riesgo y gestión del conocimiento (capacitación); con la facilidad de funcionar de forma independiente para asuntos específicos o como un todo íntegro. Genera reportes en tiempo real, lo que posibilita la toma de decisiones (THE BRITISH STANDARDS INSTITUTION (BSI), 2013).

1.1.1.3. Sistema Automatizado de Gestión Judicial (SAGJ)

Constituye un sistema desplegado en Panamá con el propósito de informatizar el Órgano Judicial y proporcionar una estructura organizativa para las jurisdicciones del país, amparada en una ley que posibilita tomar medidas para la informatización de los procesos judiciales. La aplicación administra datos referentes a: la corte suprema de justicia, tribunales superiores, juzgados, escuelas judiciales, y brinda servicios de directorio telefónico, calendario judicial, informativos judiciales y formularios. Proporciona contraseñas a los abogados idóneos para ingresar al sistema y obtener información de las firmas forenses a las que pertenecen y otros datos de interés (ORGANO JUDICIAL DE PANAMÁ, 2013).

1.1.2. Sistemas de gestión analizados en el ámbito nacional:

1.1.2.1. Sistema Integral de Gestión de la UJC Nacional: Módulo Militante

Constituye un sistema desarrollado en la Facultad 6 de la Universidad de las Ciencias Informáticas, como parte del trabajo de diploma de Yaisel Avilés Ramirez y Alexis Molina González. El mismo permite administrar la información y estadísticas de la Unión de Jóvenes Comunistas a nivel nacional, estructura de sus comités primarios, plantillas por cada uno de los comités de base, etc.

Tiene básicamente dos módulos: Módulo de Reportes y Módulo Militante; el primero genera tablas para el control y monitoreo de aspectos esenciales para la organización, mientras que el segundo gestiona todo lo relacionado con el estado de la militancia: incorporaciones, sanciones, traslados, etc.

Fue desarrollado utilizando Java como lenguaje de programación, aprovechando las características y potencialidades que brindan los lenguajes orientados a objetos (RAMIREZ y GONZÁLEZ, 2009).

1.1.2.2. Sistema Integral de Gestión de la UJC UCI: Módulo Militante

Es una solución creada en el año 2009 por estudiantes de la Facultad 1 de la Universidad, constituyendo un trabajo de Diploma de Marlen Fajardo Díaz y Eraldo Miguel Fera Sánchez, con el objetivo de llevar a cabo la gestión de la militancia de la UJC dentro de la UCI, diseñada con una arquitectura Cliente/Servidor y desarrollada en el *Framework Symfony* en su versión 2.0 (DÍAZ y SÁNCHEZ, 2009). Aunque en principio el sistema resolvería los problemas existentes en el Departamento de Documentación y Control, lugar donde sería aplicado, la realidad demostró las deficiencias en el trabajo partiendo de las siguientes características:

- La aplicación genera tablas en hojas de cálculo, que posteriormente deben ser modificadas manualmente.

Capítulo I.

- No permite el traslado de ningún militante luego de ingresarlo a menos que este sea eliminado.
- No brinda la información diaria que necesita el Departamento, como la cantidad de comités de base, el número de militantes, el estado actualizado de la militancia de la Universidad, el número de traslados para las distintas categorías, así como de incorporaciones.
- Es de lento funcionamiento a pesar de ser usada por solo dos clientes.

1.1.3. Resultados de la investigación:

Para un mejor análisis de estos sistemas, se aplicaron varias métricas atendiendo a los criterios enunciados por (GONZÁLEZ *et al.*, 2010):

Facilidad de uso, ergonomía:

Si requiere de conocimientos técnicos (bajos, medios o altos) para su utilización.

Administración y monitoreo:

Si proporciona herramientas de administración y monitoreo de los procesos llevados a cabo dentro del sistema.

Arquitectura:

Cliente/Servidor, *Desktop*.

Modularidad:

- Monolítico, todos los procesos son realizados a un mismo nivel y vinculados.
- Modularidad de primer nivel, presenta algún módulo para realizar una tarea específica aunque la mayoría las desarrolla el propio sistema.
- Completamente modular, dividido en diferentes módulos, los cuales realizan tareas que complementadas conforman las funcionalidades del sistema.

Licencia:

Características y clasificación de la licencia, si es software libre o propietario.

Capítulo I.

Servicios:

Si brinda entrenamiento, soporte y/o consultoría a los clientes.

Documentación:

Si cuenta con documentación disponible y/o actualizada.

Se realizó un estudio de estos sistemas con el objetivo de extraer información útil. Garantizando un mejor entendimiento de los resultados, se conformó una tabla con las métricas evaluadas:

Tabla 1 Métricas aplicadas a los sistemas homólogos

Métricas/Sistemas	DocCF	Entropy	SAGJ	Sist UJC Nacional	Sist UJC UCI
Facilidad de uso	Bajo	Medio	Bajo	Medio	Bajo
Administración y monitoreo	Sí	Sí	Sí	Sí	Sí
Arquitectura	Cliente/Servidor	Cliente/Servidor	Cliente/Servidor	Desktop	Cliente/Servidor
Modularidad	completamente modular	completamente modular	modularidad de primer nivel	modularidad de primer nivel	modularidad de primer nivel
Licencia	Propietaria/No caduca	Propietaria	No se conoce	GPL	GPL
Servicios	Sí	Sí	No	Sí	No
Documentación	Sí	Sí	Sí	Sí	Sí

Luego de un estudio de estos sistemas se decide llevar a cabo la implementación de un nuevo sistema que satisfaga las necesidades del Departamento de Documentación y Control de la UJC, por la complejidad que conlleva modificar uno de los existentes, que reflejan incorrectamente el proceso del negocio. Se toman como referencia aspectos de gestión, organización de los contenidos y diseño, según las métricas aplicadas.

1.2. Tecnologías de desarrollo, herramientas y lenguajes a utilizar

Como parte del proceso investigativo que se lleva a cabo, es necesario abundar en las principales tendencias en cuanto a tecnologías y herramientas que serán utilizadas en el desarrollo de la solución que se propone.

En el desarrollo de aplicaciones web y sitios de gestión de información suelen utilizarse gran variedad de herramientas, entre las que se encuentran los *Frameworks* y los Sistemas de Gestión de Contenidos; a continuación se describen cada uno de ellos para una mejor comprensión.

1.2.1. Framework

Constituyen estructuras informáticas que dan soporte a la creación de aplicaciones, cuentan con módulos, bibliotecas¹entre otras herramientas para facilitar el desarrollo de *software*. Emplea arquitectura Modelo Vista Controlador (MVC), lo que posibilita crear código estructurado y con independencia entre las clases, engloba operaciones complejas en estructuras sencillas para su posterior uso (ESCOBAR *et al.*, 2012). *Symfony* es uno de los más utilizados a nivel mundial, a continuación se realiza una descripción y especificación del mismo.



1.2.1.1. Symfony

Es un *framework* lanzado en octubre del 2005 por Fabien Potencier, diseñado para optimizar el desarrollo de aplicaciones web, aunque en principio solo se utilizaría en Sensio Labs, compañía francesa de consultoría y servicios, donde trabajaba su creador y actual CEO². Programado en PHP³ 5.3 y compatible con diversos gestores de bases de datos como PostgreSQL, Oracle, MySQL entre otros, siendo además multiplataforma y distribuido bajo una Licencia MIT⁴. Se presentan a continuación algunas características que destacan (EGUILUZ, 2013):

- Utiliza buenas prácticas y patrones de diseño para la web.
- Cuenta con una potente línea de comandos para ahorrar tiempo en la generación de código.
- Es independiente del gestor de base de datos, implementa una capa de abstracción que posibilita cambiar el gestor de base de datos en cualquier etapa del desarrollo.

¹ En informática, biblioteca o *library* en inglés, forman un conjunto de programas que brindan servicios y datos a otros para su uso, pasando a formar parte de ellos.

² Encargado de la dirección administrativa en una organización o institución.

³Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

⁴ Empleada por el *Massachusetts Institute of Technology* (MIT), permite utilizar el software así licenciado tanto para ser software libre como privativo, compatible con la Licencia GPL.

Capítulo I.

- Mantiene la presentación de la aplicación, la lógica de negocio y la del servidor separadas.
- Código de fácil lectura, cuenta con comentarios que favorecen el mantenimiento.
- Incluye validación automática en los formularios, para facilitar la introducción de datos correctos.
- Fácil de extender mediante el uso de bibliotecas.
- Soporta autenticación y credenciales, para la gestión de usuarios y permisos.

1.2.2. Sistema de Gestión de Contenidos

El CMS (*Content Management System*) por sus siglas en inglés, consiste en una aplicación informática con potentes herramientas para la administración de contenido, fundamentalmente páginas web; brindan soporte a la confección de galerías multimedia y otros servicios, capaces de manejar una o varias bases de datos donde almacenan todo el contenido del sitio. Funciona en el servidor donde se encuentra instaurado el portal y genera dinámicamente las páginas webs, esto posibilita mayor versatilidad en la gestión del sitio respecto a páginas estáticas. La actualización, restructuración y creación de *backups*⁵ son más sencillas al agrupar todo el contenido en una base de datos estructurada en el servidor. La independencia entre el contenido y el diseño del sitio posibilita realizar cambios sin modificar el formato del contenido nuevamente. Se reconocen diferentes tipos de CMS y pueden clasificarse atendiendo a varios criterios (RODRÍGUEZ, 2012; TOMLINSON, 2010):

- Según el lenguaje de programación empleado, como por ejemplo *Active ServerPages*, *Python*, *Java*, *PHP*, *Ruby On Rails*, *ASP.net*
- Por su funcionalidad.
- *Blogs*: pensados para páginas personales.
- Comercio electrónico: surgido para la gestión de usuarios, productos y compras.
- Foros: creados para el intercambio de opiniones.
- Enseñanzas: plataforma para la gestión *on-line* de programas educativos.
- Publicaciones digitales: para la publicación de información en formatos digitales.
- Wikis: para fomentar el desarrollo colaborativo.
- Según la licencia: Código abierto, *Software* privativo.

Con el auge de Internet ha surgido gran variedad de CMS, capaces de desarrollar las más diversas aplicaciones. Su obtención puede variar en cuanto al costo, existiendo algunos de libre distribución y otros privativos. Entre los más utilizados de código abierto a nivel mundial

⁵ En informática constituye una copia de seguridad, para una posterior restauración situaciones de pérdidas de datos.

se encuentran WordPress, Joomla y Drupal, que pese a su aparente similitud, reúnen diferentes características. Para la presente investigación se estudiará el CMS Drupal al ser el más utilizado y conocido por el equipo de desarrollo y contar con variedad de módulos que facilitan la realización del sitio web.



1.2.2.1. Drupal

Es un sólido y robusto CMS, lanzado en enero del 2001 y ha llegado a convertirse en uno de los favoritos para los desarrolladores. Adecuado para sitios amplios y con abundante información. Desarrollado en PHP y capaz de soportar base de datos MySQL, PostgreSQL y SQLite, además de ser distribuido bajo una licencia GPL⁶. El uso de listas de control de acceso le ha proporcionado una seguridad altamente confiable. La última versión estable en el mercado es la 7.26, con diversas mejoras de seguridad. Se reconocen otras características, tales como (RODRÍGUEZ, 2012; TOMLINSON, 2010):

- Módulos altamente editables para cada plantilla.
- Activa comunidad que favorece el desarrollo de nuevos módulos y la actualización de los ya desarrollados.
- Abundante cantidad de módulos existentes, brindando herramientas para el desarrollo de los más diversos sitios.
- Código complicado pero limpio, algo que los desarrolladores agradecen.
- Gran flexibilidad en la creación de contenidos y de sus campos.

1.2.3. Selección de la herramienta para el desarrollo de la propuesta

Atendiendo a las características descritas, valorando la necesidad de gestionar información con diversos formatos y teniendo en cuenta la flexibilidad que se requiere, se elige para dar solución a la problemática, el CMS Drupal en su versión 7.26 que presenta mejoras de seguridad y administración: facilita la administración de usuarios, el tratamiento de roles y permisos, cuenta además con una completa y flexible gestión de los contenidos; la especificidad de sus módulos aseguran la obtención de funcionalidades concretas para las más variadas exigencias, necesarios en el sistema; la gestión del panel de administración

⁶Licencia orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre.

permite acomodar el entorno de desarrollo y simplificar la complejidad y tiempo de desarrollo. Se adecua para desarrollar las gestiones que se realizan en el Departamento.

1.2.4. Lenguajes de modelado y programación utilizados

Existen diversos lenguajes de programación para el desarrollo en la web. Estos han ido surgiendo como resultado de las necesidades de las plataformas para una mayor flexibilidad y fortaleza de las aplicaciones. A continuación se muestran las características de los lenguajes que se utilizarán para el diseño y construcción del sitio, teniendo en cuenta la selección del CMS Drupal como herramienta para el desarrollo del sistema.

1.2.4.1. HTML

Lenguaje estático para el desarrollo de sitios web; es el acrónimo de *HyperText Markup Lenguaje*, en español Lenguaje de Marcas Hipertextuales. Conformado por etiquetas, de las cuales la mayor parte son semánticas e interpretadas por el navegador, permiten la conformación de páginas web mediante la estructuración de texto. Es un lenguaje de fácil aprendizaje; lo admiten todos los navegadores webs; es extensible, lo que posibilita la incorporación de características y funciones adicionales para el diseño de la página (MORA, 2002).

1.2.4.2. Javascript

Lenguaje de programación que funciona fundamentalmente del lado del cliente. Es interpretado y basado en prototipos. Creado por Brendan Eich en la empresa Netscape *Communications*, publicado el 4 de diciembre de 1995. Sintaxis similar al lenguaje Java, aunque no es un lenguaje orientado a objetos. El *World Wide Web Consortium* (W3C) diseñó el *Document Object Model* (DOM), traducido al español como Modelo de Objetos del Documento, estándar para evitar incompatibilidades. Permite la creación de interfaces de usuario mejoradas y páginas web dinámicas. Todos los navegadores en sus últimas versiones interpretan código javascript integrado dentro de las páginas web (GOODMAN *et al.*, 2007; MANGER *et al.*, 1997).

1.2.4.3. PHP

Es un lenguaje de programación utilizado para la creación de sitios web dinámicos. El nombre es el acrónimo de *Hypertext Pre-processor* y surge en 1995 bajo una licencia de software libre. Consiste en un lenguaje interpretado del lado del servidor, encapsulado dentro del código HTML y cuenta entre sus funciones con la consulta a bases de datos. Su sintaxis ha sido influida por C, C++, Perl, Java y *Python*. Requiere de un servidor web que soporte librerías PHP para su funcionamiento; entre los más populares se encuentran Internet *Information Services* (IIS) y Apache. Es un lenguaje multiplataforma, soporta los paradigmas imperativo, orientado a objetos, procedural y reflexivo; capaz de conectarse con la mayoría

de los manejadores de bases de datos, tales como: PostgreSQL, ORACLE, MySQL entre otros; posibilita la expansión de sus capacidades con la utilización de módulos y presenta gran cantidad de documentación en su página web que incluye descripciones y ejemplos de cada una de sus funciones.

Teniendo en cuenta que cualquier código que se ejecute en el lado del servidor puede ser poco fiable al tratar con los archivos, información y conexiones de red, es importante resaltar que PHP está diseñado para ser un lenguaje más seguro en la creación de programas CGI⁷ que C o Perl, y una correcta configuración en tiempos de compilación y ejecución (GRANADO, 2004; PHP GROUP, 2006).

1.2.4.4. UML

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés), herramienta utilizada para definir un sistema, modelar los procesos de negocios, detallar artefactos, documentar y modelar gráficamente el sistema. Agrupa componentes de ingeniería de software y facilita el modelado orientado a objetos, de componentes, de flujos de trabajo, etc. Se reconocen en los gráficos UML: las Entidades (elementos abstractos que conforman los diagramas, presentan atributos que los describen y caracterizan) y las Relaciones (vínculos entre las Entidades; denotan el tipo de relación que guardan) (JACOBSON, 2000).

1.2.5. Sistemas de Gestión de Base de Datos (SGBD)

Los SGBD constituyen programas informáticos encargados de la creación y administración de bases de datos; estos sistemas brindan una interfaz al usuario con herramientas para el manejo estructurado y eficiente de grandes volúmenes de información. Tienen como objetivos garantizar la seguridad de los contenidos, la independencia de los esquemas físicos y lógicos de las bases de datos respecto a las aplicaciones que la usan, minimizar la demora en la obtención de la información solicitada, entre otras. Teniendo en cuenta que se usará el CMS Drupal para el desarrollo de la solución, el cual soporta MySQL y PostgreSQL como bases de datos, se procede a caracterizar cada uno de ellos (MATO, 2005).

1.2.5.1. MySQL

Es un sistema de administración de Base de Datos, programado mayormente en C y C++; su objetivo es gestionar bases de datos relacionales. Funciona en una arquitectura cliente/servidor; es multihilos y multiusuario. MySQL es un software de código abierto, publicado bajo licencia GNU GPL⁸, soportado por múltiples plataformas. MySQL facilita una

⁷ Tecnología que permite a un navegador web solicitar datos de un programa ejecutado en un servidor web, especifica un estándar para transferir datos entre el cliente y el programa.

⁸ Licencia orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre.

conectividad segura, la replicación, transacciones, la indexación de campos de texto, entre otras características que se han ido incorporando en las últimas versiones. Es de uso factible en cualquier situación por su flexibilidad y las herramientas que brinda (COBO, 2005; GILFILLAN, 2003).

1.2.5.2. PostgreSQL

Sistema de gestión de base de datos relacionales, distribuido bajo una licencia BSD⁹ permitiendo así el uso del código fuente. Es conocida su alta concurrencia: permite la consulta a tablas en proceso de actualización sin necesidad de bloqueos. PostgreSQL cuenta con una comunidad de desarrolladores en todo el mundo colaborando gratuitamente. Los tipos nativos presentan una amplia variedad, encontrándose: texto de largo ilimitado, Arreglos (*Arrays*), direcciones IP¹⁰, variedad de números, entre otros; el usuario puede además elaborar sus propios tipos de datos. Los códigos que se ejecutan en el servidor pueden ser escritos en diferentes lenguajes, por ejemplo: PL/PgSQLC (lenguaje propio), C++, Java y PL/Perl. PostgreSQL se considera multiplataforma por ser capaz de funcionar sobre varios sistemas operativos; emplea las Llaves Foráneas (*foreign keys*), los Disparadores (*triggers*), las Vistas, herencia de tablas y otras características que aseguran la fortaleza del sistema para la administración de bases de datos (FERNÁNDEZ *et al.*, 2011; POSTGRESQL DEVELOPMENT TEAM, 2013).

1.2.5.3. Selección del Sistema de Gestión de Bases de Datos

Por las características expuestas en relación a los SGBD usados por Drupal, considerando la amplia comunidad de desarrolladores y libertades que brinda la licencia BSD, la consistencia y la escalabilidad que provee, se escoge PostgreSQL como el Sistema para la Gestión de la Base de Datos del sitio a crear.

1.2.6. Servidor Web

Consiste en una aplicación informática instalada en una computadora, capaz de recibir peticiones HTTP de los clientes a través de sus navegadores web, interpretarlas y enviar código HTML para que el navegador web del cliente genere la página web. Los servidores web pueden ejecutar aplicaciones que devuelven código HTML, el cual es enviado al cliente; o enviar las aplicaciones y ejecutarlas en el navegador del cliente, generalmente constituyen programas sencillos escritos en los lenguajes Java o Javascript, almacenados en archivos de

⁹ Licencia de software libre permisiva, tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público, permite el uso del código fuente en software no libre.

¹⁰ Etiqueta numérica que identifica, de manera lógica y jerárquica, a un elemento de comunicación de un dispositivo, generalmente una computadora; dentro de una red que utilice el protocolo IP (*Internet Protocol*).

texto plano¹¹. Se abordarán las características fundamentales de los servidores HTTP: Apache, Internet *Information Services* (IIS) y Cherokee (KUROSE y ROSS, 2004).

1.2.6.1. Apache

Es un servidor web de código abierto, surgido en 1995. Desde su aparición ha alcanzado una amplia aceptación. Es un proyecto de la Fundación de Software Apache para proporcionar servicios HTTP seguros, extensibles y eficientes. Cuenta con miles de Web masters en el mundo que mantienen sitios webs críticos y corrigen los errores por ellos mismos o por otros miembros de la Fundación. Es multiplataforma, modular (puede ser adaptado según las necesidades, con diferentes módulos de apoyo que proporciona, y con la API de programación de módulos), extensible (al ser modular se han desarrollado diversas extensiones, destacándose PHP) (THE-APACHE-SOFTWARE-FOUNDATION, 2013).

1.2.6.2. Internet *Information Services* (IIS)

Servidor web desarrollado para los sistemas operativos de Microsoft. Inicialmente lanzado para *Windows* NT 3.51. Constituye un software privativo, basado en módulos para procesar páginas que incluyen los *Active Server Pages* (ASP¹²) y ASP.NET, aunque es capaz de soportar páginas de PHP y Perl. Ofrece fundamentalmente los siguientes servicios: FTP, SMTP, NNTP, HTTP y HTTPS, y permite administrar un servidor de manera sencilla al contar con herramientas y funciones que le brindan seguridad (DOMÍNGUEZ-DORADO, 2004).

1.2.6.3. Cherokee

Un servidor creado en 2001 bajo la dirección de Álvaro López Ortega, con la intención de crear un servidor más ligero que Apache sin que perdiera rapidez o funcionalidad. Distribuido bajo la Licencia Pública General de GNU, constituye un software libre programado en C¹³, multiplataforma y actualizado por una comunidad de desarrolladores de Cherokee. Dispone de un panel de administración desde la web, soporta la configuración de servidores virtuales, balanceo de carga, archivos de *log* compatibles con apache, métodos de registro NCSA y W3C, entre otras características (ORTEGA, 2003).

¹¹ Archivo informático compuesto únicamente por texto sin formato, se pueden codificar de distintos modos dependiendo del lenguaje.

¹² Tecnología de Microsoft del tipo "lado del servidor" para páginas web, generadas dinámicamente; comercializada como un anexo a Internet *Information Services* (IIS).

¹³ Lenguaje de programación débilmente tipificado de medio nivel pero con muchas características de bajo nivel, apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas.

1.2.6.4. Selección del Servidor Web

Teniendo en cuenta la importancia que tiene para el país la migración al software libre, la comunidad de desarrolladores existente, la fortaleza y funcionalidades que brinda, se define con claridad que el Servidor HTTP Apache satisface en mayor medida las necesidades existentes en el Departamento.

1.2.7. Herramientas CASE

Acrónimo de *Computer Aided Software Engineering*: Ingeniería de Software Asistida por Computadora, conforman el conjunto de aplicaciones informáticas destinadas fundamentalmente a mejorar la productividad en el desarrollo del software, aumentar su calidad, reducir el tiempo y coste del desarrollo y mantenimiento de los sistemas informáticos. Son útiles en cualquier fase del ciclo de vida del desarrollo de un proyecto y en términos generales se pueden clasificar de la siguiente forma (DAVIS y MATA, 1992; GONZÁLEZ CABRERA y TORRES, 2012):

- Las fases del ciclo de vida del sistema que cubren.
- Su funcionalidad.
- Las plataformas que soportan.
- La arquitectura de las aplicaciones que producen.

Para el desarrollo de la solución, es necesario el uso de una herramienta CASE que cumpla y satisfaga las necesidades del sistema. Se caracterizarán a continuación, algunas de las más usadas:

1.2.7.1. Visual Paradigm

Herramienta para el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. Promueve el la rapidez y eficiencia del proceso de desarrollo del software. Es una herramienta para el desarrollo de aplicaciones utilizando modelado UML; ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de sistemas que están interesados en construcción de aplicaciones a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos. *Visual Paradigm* ofrece (MORALES, 2012; VISUAL PARADIGM, 2011):

- Navegación intuitiva entre la escritura del código y su visualización.
- Potente generador de informes en formato PDF/HTML.
- Documentación automática Ad-hoc.
- Ambiente visualmente superior de modelado.
- Sofisticado diagramador automáticamente de *layout*.
- Sincronización de código fuente en tiempo real.

1.2.7.2. MagicDraw

Es una herramienta CASE desarrollada por No Magic, con licencia de software propietario. Desarrollada en Java y permite la creación de código en diversos lenguajes de programación: Java, C# y C++, entre otros; además de la modelación de datos. Cuenta con complementos para modelar la arquitectura de empresas y para la ingeniería de sistemas, entre otros. La herramienta es compatible con los IDEs (GÉNOVA *et al.*, 2009; MAGICDRAW, 2013; ŠILINGAS y BUTLERIS, 2009):

- Oracle *Workshop* 8.1.2.
- NetBeans, desde la versión 6.
- Eclipse, desde la versión 3.1.
- IBM *Rational Application Developer*.

1.2.7.3. Microsoft Visio

Herramienta desarrollada para sistemas operativos de Windows, luego que este la adquiriera en el año 2000. Distribuido bajo una licencia propietaria, en la actualidad presenta grandes cambios relacionados a sus funcionalidades y entornos donde usarla, modificando las facilidades que brindaba a los arquitectos e ingenieros por características del desarrollo corporativo. Permite la creación de diagramas de bases de datos, compatibles con UML, diagramas de flujo de negocio, se encuentra incluido en su *Suite* Ofimática a partir de Microsoft Office 2003 (CHAVIANO GÓMEZ y HERNÁNDEZ GONZÁLEZ, 2010; EVANS *et al.*, 2003).

1.2.7.4. Selección de la herramienta CASE

Se usará la herramienta para UML *Visual Paradigm* en su versión 8.0, por brindar soporte para todo el ciclo de desarrollo del software, proporcionar medidas eficaces en el análisis y diseño de sistemas, ser multiplataforma y de fácil aprendizaje.

1.2.8. Entornos Integrados de Desarrollo

Un Entorno integrado de desarrollo (IDE, por sus siglas en inglés) constituye un sistema para la programación de aplicaciones. Brindan una interfaz amigable para un gran número de lenguajes de programación dentro de los que se encuentran: Java, C#, C++, *Visual Basic*, PHP, *Python*, Delphi, entre otros. Un IDE está compuesto fundamentalmente por (BRAGDON *et al.*, 2010):

- Un editor de código: diseñado para uno o varios lenguajes de programación, en los que identifica y corrige errores.
- Un compilador: aplicación informática capaz de traducir un programa escrito en un lenguaje de programación a otro, formando un programa equivalente que es

interpretado por la computadora; usualmente el segundo lenguaje es lenguaje de máquina¹⁴.

- Un intérprete: programa informático capaz de analizar y ejecutar otros programas, escritos en un lenguaje de alto nivel¹⁵.
- Un depurador: programa empleado para corregir los errores y probar otros programas.
- Un cliente: proceso o programa que accede a los recursos y servicios brindados.
- Posibilidad de ofrecer un sistema de control de versiones.
- Factibilidad para ayuda en la construcción de interfaces gráficas de usuario.

Existen variedad de IDEs para el desarrollo de aplicaciones, en la presente investigación se caracterizan dos de las más usadas.

1.2.8.1. NetBeans

NetBeans comenzó en 1996 como un proyecto estudiantil bajo la tutoría de la Facultad de Matemáticas y Física de la Universidad Carolina, en Praga, bajo el nombre de Xelfi. En la primavera de 1999, Netbeans DeveloperX2 fue lanzado y desde sus primeras versiones se convirtió en una alternativa viable para el desarrollo de aplicaciones. *Sun Microsystems* patrocina el proyecto desde junio del 2000; en la actualidad constituye un Entorno Integrado de Desarrollo de software libre, distribuido bajo la Licencia Pública General de GNU, conocida por su nombre en inglés *General Public License*. Es un sistema extensible y existen abundantes módulos encaminados a un desarrollo específico: vinculados a aplicaciones web con PHP5, aplicaciones de escritorio, para móviles, para la creación de módulos en Drupal y permitir la integración con *frameworks* como *Symfony*. Debido a que los módulos pueden desarrollarse independientemente, los productos de la plataforma NetBeans pueden ser ampliados fácilmente por otros desarrolladores de software. Es un IDE multiplataforma, programado en Java pero soporta cualquier otro lenguaje de programación (AGUIRRE VILLARREAL y VELASTEGUÍ CARRERA, 2012; BÖCK, 2009).

1.2.8.2. Eclipse

Es un Entorno Integrado de Desarrollo de código abierto y multiplataforma. Consiste en una aplicación modular, programada en Java aunque soporta otros lenguajes como: C, C++, *Python* y *LaTeX*, Sistema de gestión de base de datos y aplicaciones web. Cuenta con una comunidad de desarrolladores que extienden constantemente sus funcionalidades, aplicaciones y módulos. Distribuido en la actualidad bajo la *Eclipse Public License*

¹⁴ Sistema de códigos directamente interpretable por un circuito microprogramable, constituyen un conjunto de instrucciones que determinan acciones a ser tomadas por la máquina.

¹⁵ Algoritmos escritos de una manera comprensible para los humanos, capaces de dar instrucciones que luego serán interpretadas por otros programas a más bajo nivel.

desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro. El Proyecto Eclipse define a su software como: "*una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular*" (AGUIRRE VILLARREAL y VELASTEGUÍ CARRERA, 2012).

1.2.8.3. Selección del Entrono Integrado de Desarrollo

En la presente investigación se utilizará el IDE Netbeans en su versión 7.3 para el desarrollo de la solución, atendiendo a sus características y dado que se tiene mayor experiencia y familiarización con esta herramienta por parte del equipo de trabajo.

1.2.9. Administrador de Base de Datos

El pgAdmin es la herramienta más popular para administración de PostgreSQL, de código abierto, una base de datos avanzada, distribuida bajo la Licencia PostgreSQL. La aplicación puede utilizarse en sistemas operativos de Linux, FreeBSD, Solaris, Mac OSX y plataformas de Windows para administrar PostgreSQL a partir de su versión 7.3; así como las versiones comerciales y derivados de PostgreSQL, entre las que se encuentran: *Postgres PlusAdvanced Server* y base de datos *Greenplum*. Diseñado para responder a las necesidades de todos los usuarios. Cuenta con una interfaz gráfica compatible con todas las características de PostgreSQL y es el encargado de la administración. Entre sus características destaca: incluye un editor de resaltado de sintaxis SQL, un agente de planificación de tareas de SQL / *batch* / *Shell*, un editor de código del lado del servidor, las conexiones al servidor se pueden hacer a través de TCP/IP¹⁶ o *Unix Domain*, permite encriptado SSL para garantizar la seguridad. PgAdmin lo conforman una comunidad de desarrolladores en todo el mundo, por lo que está disponible en más de una decena de idiomas. La última versión liberada es la 1.18.1 (FERNÁNDEZ *et al.*, 2011).

1.2.10. Metodologías

Las metodologías de desarrollo de software conforman el conjunto de técnicas y procedimientos que rigen el desarrollo de productos software y su documentación. Se especifican todos los pasos a seguir, actividades a realizar, las personas involucradas y el papel que juegan en el proceso de desarrollo de software; se detalla además la información necesaria para cada actividad y la que debe producirse como resultado de la misma. Se combinan el empleo de UML con la práctica de procedimientos detallados. Existen diversas metodologías disponibles para aplicar en el proceso de desarrollo de software, se clasifican en robustas y ágiles, el equipo de desarrollo caracteriza a continuación una de apartado para

¹⁶ Conjunto de protocolos de red en los que se basa Internet, que permiten el intercambio de información entre ordenadores.

tener un conocimiento abarcador sobre cada tipo y seleccionar la apropiada para el desarrollo de la solución (SAMPIERI *et al.*, 2006).

1.2.10.1. OpenUP:

Constituye un marco de trabajo para procesos de desarrollo de software. Liberado por el *EclipseProcess Framework* (EPF), construido sobre una donación realizada por IBM del *Basic Unified Process* y entregada a Fundación Eclipse a finales del 2005, renombrado en 2006 como OpenUP. Distribuida bajo la Licencia Pública de Eclipse, pensada para lograr un mayor intercambio de información entre el equipo de desarrollo y promover aspectos esenciales del proyecto: objetivos, alcance y avance del mismo. Entre sus características generales se encuentran (BELL, 2010):

- Preserva la esencia del *Unified Process*:
 - Desarrollo iterativo e incremental.
 - Centrado en la Arquitectura.
 - Desarrollo dirigido por Casos de Uso.
- Está pensada para proyectos pequeños.
- Solo se incluye lo fundamental, sin dejar de ser completo y extensible (menos de 20 artefactos).

La metodología se conforma por dos dimensiones relacionadas:

- El método: define todos los elementos básicos que formarán parte del ciclo de vida del proyecto (Tareas, roles, artefactos y resultados, entre otros).
- El proceso: organiza los elementos del método en el tiempo y los aplica.

Entre los principios de OpenUP destacan:

- Colaborar para armonizar intereses y compartir el conocimiento.
- Desarrollo evolutivo continuo para obtener retroalimentación y mejorar continuamente.
- Estabilizar prioridades para extender el beneficio alcanzado por los interesados en el proyecto.
- Enfocarse en la arquitectura desde el comienzo para minimizar los riesgos y organizar el desarrollo.

1.2.10.2. RUP

El Proceso Unificado Racional, del inglés *Rational Unified Process* (RUP) constituye una metodología estándar para el análisis, documentación e implementación de proyectos. En 1995 la compañía *Rational Software* adquirió la empresa llamada Objectory AB, fundada por quien incorporara los casos de uso a los métodos de desarrollo de sistemas orientados a objetos, Ivar Jacobson. Resultado de la correlación de *Rational Approach* y Objectory (proceso de la empresa Objectory AB), surge en 1998 la primera versión del *Rational Unified Process*. Cuenta con tres características fundamentales: está centrado en la arquitectura, está dirigido por los Casos de Uso y es iterativo e incremental. RUP identifica seis prácticas deseables con las cuales define la forma efectiva de trabajo para los equipos de desarrollo de software (JACOBSON, 2000):

- Gestión de requisitos: cuenta con una guía para encontrar, documentar, organizar y seguir los cambios de los requisitos funcionales y las restricciones.
- Desarrollo de software iterativo: progreso del producto mediante iteraciones con hitos bien definidos, realizado las actividades en función de la fase en que se encuentra el producto.
- Desarrollo basado en componentes: creación de sistemas divididos en componentes con interfaces bien definidas, que posteriormente se ensamblarán; el producto se crea a medida que se desarrollan sus componentes.
- Modelado visual (usando UML): el uso de herramientas para el modelado visual facilita la gestión de los componentes, consolida los requisitos, diseños e implementaciones a desarrollarse.
- Verificación continua de la calidad: se evalúan los artefactos durante el proceso de desarrollo y al final de cada iteración, integrando pruebas en todo el período y revisiones continuas a los artefactos no ejecutables.
- Gestión de los cambios: RUP cuenta con esta disciplina para mitigar los riesgos que representan el desarrollo y mantenimiento de software. Los artefactos de software cambian durante el proceso de desarrollo, teniendo gran impacto los cambios en los requisitos.

1.2.10.3. Selección de la metodología

Atendiendo a las características del sistema a implementar y las metodologías descritas, se selecciona la metodología OpenUP, por adecuarse mejor a las condiciones de desarrollo existentes: el tiempo con que cuenta el equipo de desarrollo y la necesidad de liberar versiones al cliente mientras se implementa la solución.

Capítulo I.

La aplicación de métricas a sistemas con propósitos similares estudiados a nivel nacional e internacional, enfocando la investigación en sistemas de gestión con funcionalidades afines a las desarrolladas en el Departamento de Documentación y Control de la UJC, apoyó la decisión de implementar un sistema nuevo por los costos que requiere la modificación de alguno de los existentes. La utilización de la metodología de desarrollo OpenUP guiará el desarrollo y garantizará un correcto flujo de procesos y la liberación de versiones del sistema al cliente durante su desarrollo. El empleo de Drupal en su versión 7.26 agilizará la creación de versiones garantizando su calidad. Será usado Apache en su versión 2.2.21 como servidor web y PostgreSQL en su versión 9.1 como servidor de base de datos, brindando seguridad y garantizando la disponibilidad del sistema. La utilización de la herramienta *Visual Paradigm* en su versión 8.0 facilitará el modelado del sistema y sus funcionalidades. El uso IDE Netbeans en su versión 7.3 permitirá acortar el tiempo de desarrollo.

Capítulo 2: Caracterización del sistema de gestión de información de los militantes.

En el presente capítulo se exponen las características del sistema a desarrollar; haciendo uso de la metodología OpenUP, se define el modelo de dominio para el objeto de estudio. Se identifican y describen los requisitos funcionales a implementar por el sistema de gestión de la militancia en la UJC. Se explica el diseño de la propuesta de solución.

2.1. Propuesta del Sistema de gestión de la militancia en la UJC de la Universidad

Uno de los métodos empleados para la recopilación de información fue el Analítico-Sintético, posibilitando una mayor comprensión de la gestión llevada a cabo en el Departamento de Documentación y Control de la UJC; se realizó una entrevista a Ramón Paumier Samón (Jefe del Departamento de Documentación y Control de la UJC), disponible en los anexos del trabajo de diploma, aportando diversos requisitos. Se identificaron los diferentes contenidos a tratar: el comité primario, comité de base, el militante, la relación que presentan y la distribución de los mismos dentro del sistema; la estructura del sitio y de la información que lo compone; los reportes que genera el departamento y sus componentes.

El sistema almacenará datos de interés para la UJC vinculados a los militantes, comités de base y primarios, mantendrá un registro de las causas de incorporaciones y de bajas, y generará reportes mensuales con todos los cambios realizados en los diferentes niveles: comités de base, comités primarios y a nivel UCI; permitirá además la impresión de los reportes generados y sus respaldos.

La propuesta de solución mejora el tratamiento de roles, la gestión de los contenidos, propone una administración intuitiva, lo cual facilitará el trabajo a personas inexpertas en el área de la informática.

2.2. Modelo de Dominio

El modelo de dominio es el encargado de representar los conceptos fundamentales para el desarrollo de una aplicación y las diferentes relaciones que existen entre ellos. Son presentados como clases los aspectos esenciales y las interrelaciones denotan una estructura de funcionamiento, brindan una mejor comprensión del medio actual para la concepción de un futuro sistema. Para el desarrollo del sitio de gestión de la militancia en la UJC dentro de la Universidad se decide representar este modelo y no un modelo de negocio, por no ser tan complejo el proceso llevado a cabo en el Departamento de Documentación y Control de la UJC.

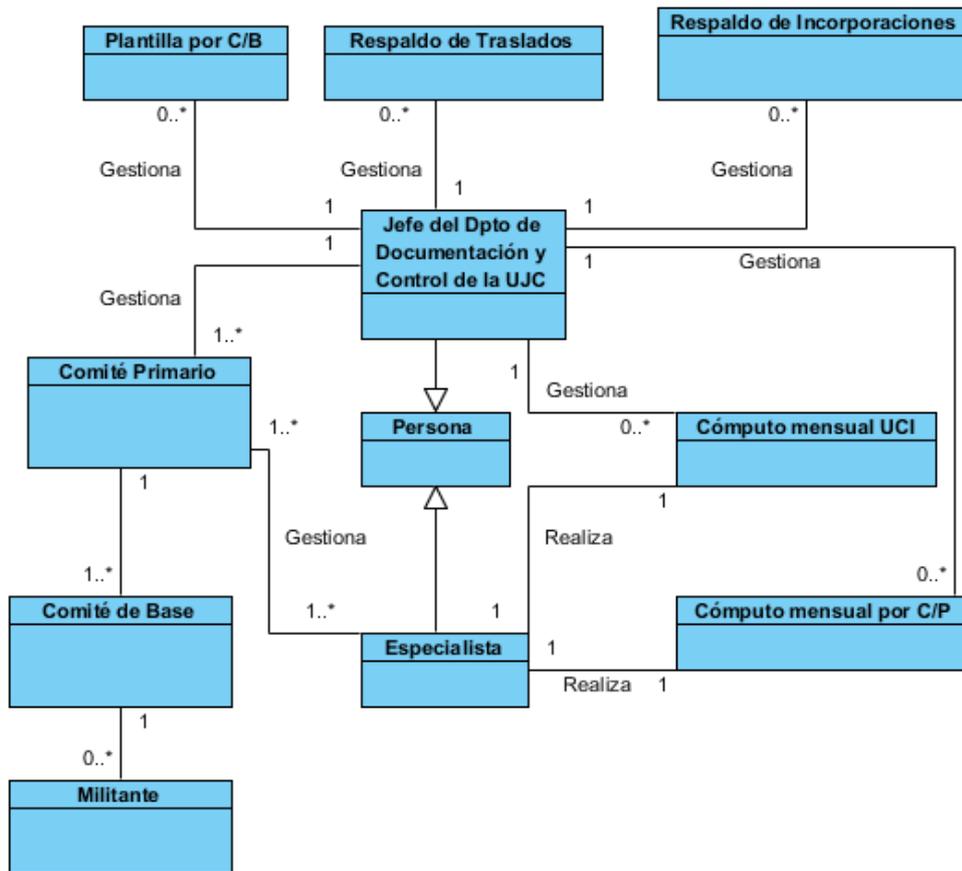


Figura 1 Modelo de dominio.

2.2.1. Descripción de clases y roles del diagrama

A continuación son descritos los diferentes objetos que participan en el proceso y relaciones que guardan.

Militante: el centro de todo el proceso de gestión, se almacenan datos de interés por la Organización y el estado de su militancia.

Jefe del Departamento de Documentación y Control de la UJC: persona que gestiona los contenidos del Sistema de gestión de la militancia.

Persona: entidad genérica que representa tanto al jefe del departamento como al especialista.

Especialista: persona del Departamento de Documentación y Control, encargada de la gestión de los militantes, los comités de base, comités primarios y los reportes mensuales.

Comité de Base: estructura organizativa a la que pertenecen los militantes de la UJC.

Comité Primario: instancia que agrupa a los diferentes comités de base existentes en la Universidad.

Plantilla por Comité de Base: documento donde se reflejan los militantes por comité de base al que pertenecen, contiene toda la información asociada a ellos.

Respaldo de Traslados: son tablas que almacenan la información vinculada a los traslados, pueden realizarse hacia provincias, municipios, exterior, FAR y MININT, además de los internos.

Respaldo de Incorporaciones: son tablas que almacenan la información relacionada con la causa de incorporación de los militantes, estas pueden ser: provincia, municipio, FAR, MININT y exterior.

Cómputo mensual UCI: representan el resultado de las operaciones realizadas por la Organización por mes, en él se agrupan las incorporaciones, los traslados, las sanciones, para toda la Universidad.

Cómputo mensual por C/P: refleja las operaciones realizadas por la UJC en el mes, incluye las incorporaciones, los traslados, las sanciones, pero a nivel de los comités primarios.

2.3. Requerimientos del Sistema

Para el desarrollo del sitio de gestión de la militancia en la UJC es necesario contar con las funcionalidades y características que el cliente demanda, dichas características se definen como requisitos que debe cumplir el sistema. Se identifican los requisitos funcionales y no funcionales: siendo los funcionales las tareas que debe cumplir el producto y los no funcionales constituyen criterios para evaluar las operaciones de los sistemas, no su comportamiento.

Tabla 2 Requisitos funcionales

Requisitos funcionales	Nombre de los requisitos
RF1	Autenticar Usuario
RF2	Añadir Usuario
RF3	Modificar Usuario
RF4	Eliminar Usuario
RF5	Listar Usuario
RF6	Adicionar Comité Primario
RF7	Modificar Comité Primario
RF8	Eliminar Comité Primario
RF9	Mostrar Comité Primario
RF10	Buscar Comité de Base
RF11	Adicionar Comité de Base
RF12	Modificar Comité de Base

Capítulo II.

RF13	Eliminar Comité de Base
RF14	Listar plantilla del Comité de Base
RF15	Adicionar militante
RF16	Buscar militante
RF17	Modificar militante
RF18	Eliminar militante
RF19	Listar datos del militante
RF20	Crear traslado interno
RF21	Eliminar traslado interno
RF22	Listar traslado interno
RF23	Imprimir traslado interno
RF24	Crear traslado para provincia
RF25	Eliminar traslado para provincia
RF26	Listar traslado para provincia
RF27	Imprimir traslado para provincia
RF28	Crear traslado para municipio
RF29	Eliminar traslado para municipio
RF30	Listar traslado para municipio
RF31	Imprimir traslado para municipio
RF32	Crear traslado para el exterior
RF33	Eliminar traslado para el exterior
RF34	Listar traslado para el exterior
RF35	Imprimir traslado para el exterior
RF36	Crear traslado para las FAR
RF37	Eliminar traslado para las FAR
RF38	Listar traslado para las FAR
RF39	Imprimir traslado para las FAR
RF40	Crear traslado para el MININT
RF41	Eliminar traslado para el MININT
RF42	Listar traslado para el MININT
RF43	Imprimir traslado para el MININT
RF44	Crear pases al PCC
RF45	Eliminar pases al PCC
RF46	Listar pases al PCC
RF47	Imprimir pases al PCC
RF48	Crear baja por sanción
RF49	Eliminar baja por sanción
RF50	Listar baja por sanción
RF51	Imprimir baja por sanción

Capítulo II.

RF52	Crear incorporación de provincia
RF53	Eliminar incorporación de provincia
RF54	Listar incorporación de provincia
RF55	Imprimir incorporación de provincia
RF56	Crear incorporación de municipio
RF57	Eliminar incorporación de municipio
RF58	Listar incorporación de municipio
RF59	Imprimir incorporación de municipio
RF60	Crear incorporación del exterior
RF61	Eliminar incorporación del exterior
RF62	Listar incorporación del exterior
RF63	Imprimir incorporación del exterior
RF64	Crear incorporación de las FAR
RF65	Eliminar incorporación de las FAR
RF66	Listar incorporación de las FAR
RF67	Imprimir incorporación de las FAR
RF68	Crear incorporación del MININT
RF69	Eliminar incorporación del MININT
RF70	Listar incorporación del MININT
RF71	Imprimir incorporación del MININT
RF72	Crear incorporación por crecimiento
RF73	Eliminar incorporación por crecimiento
RF74	Listar incorporación por crecimiento
RF75	Imprimir incorporación por crecimiento
RF76	Crear defunción
RF77	Eliminar defunción
RF78	Listar incorporación por crecimiento
RF79	Imprimir defunción
RF80	Generar cómputo mensual por comités de base
RF81	Generar cómputo mensual a nivel UCI
RF82	Generar tabla #10

2.3.1. Descripción de los requisitos funcionales

A continuación se describen algunos de los requisitos funcionales de importancia para el cliente, los restantes se encuentran en los anexos del trabajo de diploma.

Tabla 3 Descripción de requisitos. Adicionar militante.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF15	Adicionar militante	El sistema debe permitir adicionar un militante	Media	Alta
Prototipo				
Campos	Tipos de Datos		Reglas o Restricciones	
Nombre del militante	Texto		Puede contener solo letras	
Apellidos	Texto		Puede contener solo letras	
Fecha de Alta	Fecha		Puede contener solo valores numéricos	
Causa	Texto		Solo se admiten los valores: Exterior, FAR, MININT, Municipio, Provincia y Crecimiento	
Carné de Identidad	Número		Puede contener solo valores numéricos	
Fecha de inicio en la UJC	Fecha		Puede contener solo valores numéricos	
Sexo	Texto		Solo se admiten los valores: Masculino y Femenino	

Capítulo II.

	Raza	Texto	Solo se admiten los valores: Blanca, Mestiza y Negra
	Clasificación / Ocupación	Texto	Solo se admiten los valores: Estudiante, Profesor o Trabajador no docente
	Comité de base	Texto	Puede contener cualquier cadena alfa-numérica
	Nivel Cultural	Texto	Solo se admiten los valores: Doce Grado, Técnico Medio Universitario
	Dirección Particular	Texto	Puede contener cualquier cadena alfa-numérica
	Provincia-Municipio	Texto	Solo se admiten los valores presentados
	Observaciones	Los * de color rojo indican campos obligatorios.	

Capítulo II.

Tabla 4 Descripción de requisitos. Modificar militante.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF17	Modificar militante	El sistema debe permitir modificar un militante	Media	Alta
Prototipo				
<div style="border: 1px solid #ccc; padding: 10px;"> <p>Nombre del militante: *</p> <input type="text" value="Pepe"/> <p>Apellidos: *</p> <input type="text" value="Acosta"/> <p>FECHA DE ALTA</p> <input type="text" value="06/04/2014"/> <small>E.g., 07/04/2014</small> <p>Causa: *</p> <p> <input checked="" type="radio"/> Crecimiento <input type="radio"/> Exterior <input type="radio"/> FAR <input type="radio"/> MININT <input type="radio"/> Municipio <input type="radio"/> Provincia </p> <p>Carné de identidad: *</p> <input type="text" value="96123485743"/> <p>FECHA DE INICIO EN LA UJC</p> <p>Date</p> <input type="text" value="2014/04/07"/> <small>E.g., 2014/04/07</small> <p>Sexo: *</p> <p> <input type="radio"/> Femenino <input checked="" type="radio"/> Masculino </p> <p>Raza: *</p> <p> <input checked="" type="radio"/> Blanca <input type="radio"/> Mestiza <input type="radio"/> Negra </p> <p>Clasificación / Ocupación: *</p> <p> <input checked="" type="radio"/> Estudiante <input type="radio"/> Profesor <input type="radio"/> Trabajador no docente </p> <p>Comité de base: *</p> <input type="text" value="4301"/> <p>Nivel Cultural: *</p> <p> <input checked="" type="radio"/> Doce grado <input type="radio"/> Técnico Medio <input type="radio"/> Universitario </p> <p>Dirección Particular: *</p> <input type="text" value="Ave 43 entre 34 y 36"/> <p>Provincia-Municipio: *</p> <p> <input type="text" value="Artemisa"/> <input type="text" value="Caimito"/> </p> <p>Causa de baja</p> <p> <input checked="" type="radio"/> N/D <input type="radio"/> Sanción <input type="radio"/> FAR <input type="radio"/> Exterior <input type="radio"/> MININT <input type="radio"/> Municipio <input type="radio"/> Provincia <input type="radio"/> Defunción </p> <p>FECHA DE BAJA</p> <p>Date</p> <input type="text" value="04/22/2014"/> <small>E.g., 04/07/2014</small> <p style="text-align: center;"> <input type="button" value="Guardar"/> <input type="button" value="Vista previa"/> <input type="button" value="Eliminar"/> </p> </div>				
	Campos	Tipos de Datos	Reglas o Restricciones	
	Nombre del militante	Texto	Puede contener solo valores alfabéticos	
	Apellidos	Texto	Puede contener solo valores alfabéticos	

Capítulo II.

	Fecha de Alta	Fecha	Puede contener solo valores numéricos
	Causa	Texto	Solo se admiten los valores: Exterior, FAR, MININT, Municipio, Provincia
	Carné de Identidad	Número	Puede contener solo valores numéricos
	Fecha de inicio en la UJC	Fecha	Puede contener solo valores numéricos
	Sexo	Texto	Solo se admiten los valores: Masculino y Femenino
	Raza	Texto	Solo se admiten los valores: Blanca, Mestiza y Negra
	Clasificación / Ocupación	Texto	Solo se admiten los valores: Estudiante, Profesor o Trabajador no docente
	Comité de base	Texto	Puede contener cualquier cadena alfa-numérica
	Nivel Cultural	Texto	Solo se admiten los valores: Doce Grado, Técnico Medio Universitario
	Dirección Particular	Texto	Puede contener cualquier cadena alfa-numérica
	Provincia-Municipio	Texto	Solo se admiten los valores presentados
	Fecha de baja	Fecha	Puede contener solo valores numéricos
	Causa de baja	Texto	Solo se admiten los valores que se presentan
	Observaciones	Los * de color rojo indican campos obligatorios.	

Tabla 5 Descripción de requisitos. Eliminar militante.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente																
RF18	Eliminar militante	El sistema debe permitir eliminar un militante	Media	Alta																
Prototipo																				
<table border="1"> <thead> <tr> <th><input type="checkbox"/></th> <th>TÍTULO</th> <th>TIPO</th> <th>AUTOR</th> <th>ESTADO</th> <th>ACTUALIZADO</th> <th>IDIOMA</th> <th>OPERACIONES</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>Manolo</td> <td>Militante</td> <td>admin</td> <td>publicado</td> <td>12/03/2014 - 14:23</td> <td>Español</td> <td>editar eliminar</td> </tr> </tbody> </table>					<input type="checkbox"/>	TÍTULO	TIPO	AUTOR	ESTADO	ACTUALIZADO	IDIOMA	OPERACIONES	<input type="checkbox"/>	Manolo	Militante	admin	publicado	12/03/2014 - 14:23	Español	editar eliminar
<input type="checkbox"/>	TÍTULO	TIPO	AUTOR	ESTADO	ACTUALIZADO	IDIOMA	OPERACIONES													
<input type="checkbox"/>	Manolo	Militante	admin	publicado	12/03/2014 - 14:23	Español	editar eliminar													
	Campos	Tipos de Datos	Reglas o Restricciones																	
	No procede	No procede	No procede																	
	Observaciones																			

Capítulo II.

Tabla 6 Descripción de requisitos. *Buscar militante.*

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF13	Buscar militante	El sistema debe permitir buscar un militante	Media	Alta
Prototipo				
<h3>Buscar militante</h3> <p> Nombre <input type="text"/> Carné de identidad <input type="text"/> Comité de base <input type="text" value="1502"/> </p> <p style="text-align: right;"><input type="button" value="Buscar"/></p>				
	Campos	Tipos de Datos	Reglas o Restricciones	
	Nombre	Texto	Puede contener solo valores alfabéticos	
	Carné de identidad	Número	Puede contener solo valores numéricos	
	Comité de base	Número	Puede contener solo valores mostrados	
	Observaciones	No es necesario introducir ambos valores para realizar la búsqueda.		

Tabla 7 Descripción de requisitos. *Listar datos del militante.*

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF13	Listar datos del militante	El sistema debe permitir listar los datos de un militante	Media	Alta
Prototipo				
<h3>Manolo</h3> <p> <input type="button" value="Vista"/> <input type="button" value="Editar"/> </p> <p>Enviado por admin el Mié, 03/12/2014 - 14:23</p> <p> Apellidos: Paz Nuñez Fecha de alta: 12/03/2014 Fecha de inicio en la UJC: 12/03/2014 Comité de base: 1502 Causa: Crecimiento Carné de identidad: 12345434664 Clasificación / Ocupación : Estudiante Nivel Cultural: Universitario Sexo: Masculino Raza: Blanca Dirección Particular: ave 31 Provincia-Municipio: Artemisa Caimito </p>				
	Campos	Tipos de Datos	Reglas o Restricciones	

	No procede	No procede	No procede
	Observaciones		

2.3.2. Requisitos no funcionales

2.3.2.1. Eficiencia:

RNF1: El sistema debe permitir que varios usuarios estén conectados a la vez y que estos realicen cambios en la base de datos sin afectar el rendimiento del sitio.

RNF2: El sistema debe dar respuesta a las peticiones de los usuarios en un intervalo de dos a cinco segundos aproximadamente.

2.3.2.2. Fiabilidad:

RNF3: El sistema debe estar disponible el 98% del tiempo, confiriendo un 2% del tiempo para actualizar y dar soporte.

RNF4: Ante la ocurrencia de algún fallo, el sistema no debe mostrar ninguna información que comprometa la integridad y confidencialidad de la UJC.

2.3.2.3. Soporte:

RNF5: El sistema debe posibilitar la incorporación de servicios y actualizaciones en caso de necesitarlo el cliente.

2.3.2.4. Interfaz:

RNF6: La computadora donde se despliegue el sistema debe presentar las características que se mencionan a continuación:

Software:

- Sistema gestor de bases de datos: PostgreSQL 9.1.
- Servidor web: Apache (versión 2.0 o superior)
- Lenguaje de programación: PHP en su versión 5.3.
- Navegador web.

Hardware:

Hardware mínimo para el servidor Web:

- 2 GB de memoria RAM.
- 80 GB de disco duro.

Capítulo II.

- Procesador de Doble núcleo con velocidad (1.2GHz) por unidad.

Hardware óptimo para el servidor Web:

- 2 GB de memoria RAM.
- 160 GB de disco duro.
- Procesador de Doble núcleo con velocidad (3.0GHz) por unidad.

2.3.3. Restricciones de Diseño

RNF7: El sistema debe ser diseñado y documentado utilizando la metodología OpenUP, utilizando para el modelado la herramienta *Visual Paradigm 8.0*, debe ser programado en el entorno de desarrollo NETBEANS 7.2 y empleando el CMS Drupal como base de toda la estructura y arquitectura del sistema.

2.3.3.1. Interfaz:

RNF8: El Sistema de gestión del estado de la militancia debe poseer en su diseño los colores azul, blanco y rojo, representando los de la bandera y una apariencia amigable para amenizar el tiempo que empleen los funcionarios de la UJC en el.

2.4. Arquitectura del sistema

La arquitectura del sistema se hereda de Drupal al ser desarrollado sobre ese CMS. Drupal conforma el sistema basándose en conceptos fundamentales, tales como: Permisos, Módulos, Plantillas, Base de Datos y Menús.

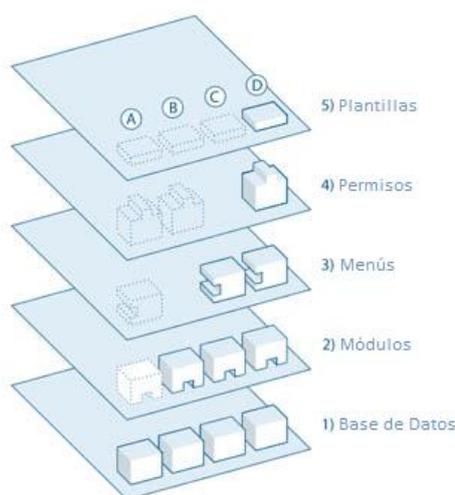


Figura 2 Arquitectura en capas de Drupal.

Capítulo II.

A continuación se describe cada uno de los elementos que componen dicha arquitectura (BUTCHER *et al.*, 2010):

La **base de datos** administra el acceso a la información almacenada, incluye la del funcionamiento del sistema y los contenidos del mismo, los nodos constituyen una parte importante de la base de datos, siendo el elemento fundamental en Drupal; almacenan toda la información de los contenidos y crece su número a la par del crecimiento del sitio.

Los **módulos** en Drupal utilizan los nodos y permiten ampliar las funcionalidades del CMS adaptándolo a las características y necesidades de cualquier sitio web. Drupal 7 concentra 44 módulos disponibles luego de su instalación y cuenta con más de 2500 módulos en el mercado.

Los **menús y bloques** en Drupal se encargan de estructurar y organizar la información dentro del sitio web, definiendo las regiones donde se mostrarán los contenidos y de establecer relaciones entre las diferentes páginas web del sistema, se adaptan al diseño gráfico del tema seleccionado y pueden constituir respuestas generada por módulos activos.

En Drupal los **roles y permisos** garantizan la integridad de la información del sitio, al delimitar las funcionalidades a las que cada usuario puede acceder, un rol constituye un conjunto de permisos y a cada usuario pueden asignársele varios roles.

Los **temas** constituyen la apariencia y estilo del sistema, separados de la información permiten ser modificados sin necesidad de cambiar el contenido del sitio web. Existen repositorios de temas libres de Drupal, permitiendo su descarga y modificación para adaptarlos a cada diseño, definen las zonas donde se mostrarán los diferentes tipos de contenidos y la apariencia de los mismos.

2.5. Patrones de diseño de Drupal

Los patrones de diseño en Drupal proporcionan soluciones eficientes para el diseño de aplicaciones, cumplen con los estándares internacionales y abonan la seguridad de los sistemas a desarrollar. En el desarrollo del sistema de gestión de la militancia en la UJC se justifican los siguientes (BUTCHER *et al.*, 2010):

2.5.1. Observador (*Observer*)

Los módulos donde se define un *hook* (gancho) para los eventos de inserción o actualización de alguna entidad, son considerados observadores de dichas entidades. Es un patrón generalizado en Drupal, evidenciado cuando se realiza alguna modificación en una entidad, se invocan en todos los módulos los *hooks* que lo implementan.

El sistema implementa este patrón en los eventos de inserción y modificación de los militantes, donde se muestran y ocultan datos según los atributos que seleccione el usuario.

2.5.2. Sencillo (*Singleton*)

Consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella. En Drupal los temas y módulos siguen este patrón, al estar separados por las funciones que implementan, pudiendo ser tomados como instancias únicas.

2.5.3. Decorador (*Decorator*)

Reside en añadir dinámicamente funcionalidad a una clase. Drupal cuenta con los *hooks* que posibilitan sobrescribir las funciones y modificar el comportamiento de los nodos en cualquier momento sin necesidad de implementar herencia para redefinir los métodos.

En el módulo desarrollado es necesario modificar atributos de los nodos dinámicamente para crear diversos tipos de contenidos que son independientes las acciones que realizan los usuarios.

2.5.4. Puente (*Bridge*)

Drupal implementa la capa de abstracción de datos siguiendo este patrón. La implementación de los módulos es independiente al gestor de bases de datos empleado, se le proporciona la capa de abstracción de datos en su lugar. Posibilita cambiar a otros gestores de bases de datos sin necesidad de modificar los módulos.

En el sistema, se realizan consultas a la base de datos empleando dicha capa y posibilita la independencia del gestor de bases de datos empleado.

2.5.5. Cadena de Responsabilidad (*Chain of Responsibility*)

Consiste en la cadena de clases receptoras por las que pasa una solicitud creada en una clase emisora. En Drupal el sistema de menús sigue este patrón: con cada petición de una página, el sistema de menús determina si existe algún módulo que requiera esa solicitud, valida el acceso al recurso por el usuario y la función que se ejecuta con dicha petición. El proceso envía la petición a cada uno de los módulos hasta que uno se encargue de ella, le niegue el acceso al usuario o se agote la cadena.

El sistema hace uso de menús que determinan la existencia de módulos que respondan a las peticiones, enviando dicha petición a todos hasta que reciba respuesta de aquel que la implemente.

2.5.6. Comando (Command)

El patrón consiste en: sin conocer el contenido de una operación ni quien la recibe, solicitar una operación a un objeto. El sistema de *hooks* en Drupal sigue este patrón, con lo que no es necesario definir cada *hook* sino aquellos que serán implementados; algunos módulos lo utilizan para minimizar funciones necesarias para la aplicación.

El módulo desarrollado utiliza los *hooks insert* y *update* que son llamados siempre que se creen o modifiquen nodos, independientemente de la clase que realice la petición serán comprobados.

2.6. Diseño del sistema

2.6.1. Diagrama de Clases de Diseño

Para una mejor comprensión del sistema a desarrollar, es necesario tener en cuenta el diagrama de paquetes de CMS Drupal. A continuación se describen los principales paquetes que lo componen:

“Themes”: lugar donde se encuentran ubicadas las plantillas que se utilizan como tema en la interfaz de Drupal, cuando se necesita incluir un nuevo diseño se copia la plantilla dentro de esta carpeta.

“Modules”: abarca todos los módulos para el funcionamiento del CMS, cuando se requiere un nuevo módulo solo debe copiarse en esa carpeta.

“Includes”: contiene ficheros indispensables para el correcto funcionamiento del sistema, entre los que se encuentran los de conexión a la base de datos.

“Scripts”: agrupa el conjunto de ficheros que posibilitan la visualización de los datos, destacándose los CSS y JavaScript.

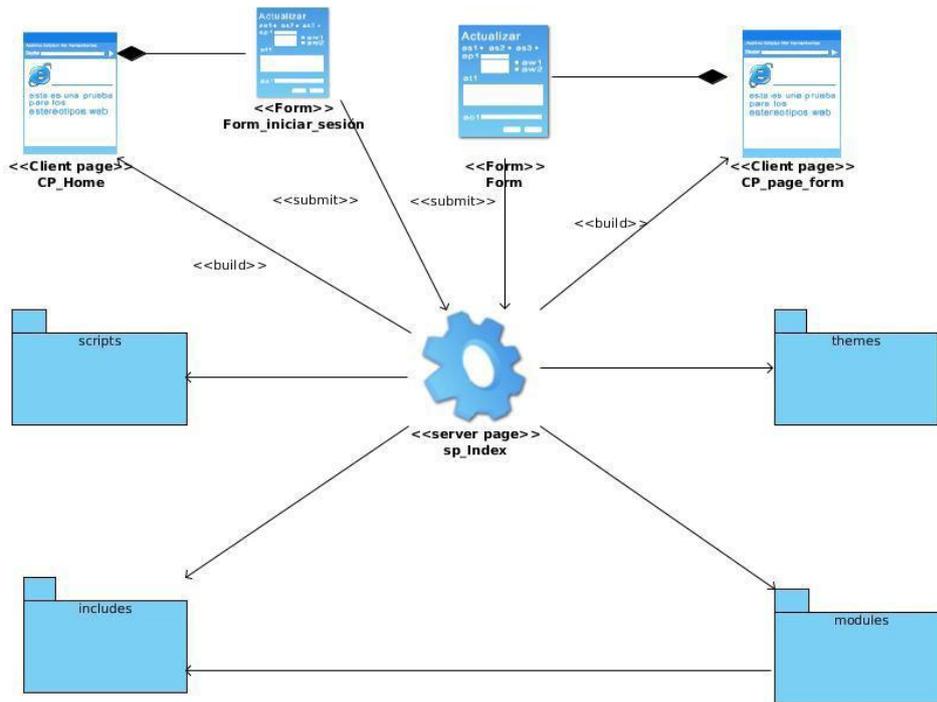


Figura 3 Diagrama de Clases del diseño de Drupal.

A continuación se describe el diagrama de paquetes para Gestionar militante, anexados al trabajo de diploma se reflejan algunos de similar importancia.

Paquete Gestionar militante

Para adicionar un militante la página servidora SP_GestionarMilitante construye la página cliente CP_AdicionarMilitante, la última se compone de un formulario Form_AdicionarMilitante donde se van a introducir los datos y que va a ser ejecutado por la página servidora. Para modificar un militante la página servidora construye la página cliente CP_ModificarMilitante, la última se compone de un formulario Form_ModificarMilitante donde se van a modificar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora CC_GestionarMilitante. Para eliminar un militante la página servidora construye la página cliente CP_EliminarMilitante, la última se compone de un formulario Form_EliminarMilitante donde se van a eliminar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora.

Para listar un militante la página servidora construye la página cliente CP_ListarDatosMilitante, la última se compone de un formulario Form_MostrarDatosMilitante donde se van a mostrar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora. Para buscar un militante la página servidora construye la página cliente CP_BuscarMilitante, la última se compone de un formulario Form_BuscarMilitante donde se van a introducir y mostrar los datos, que va a ser

Capítulo II.

ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora.

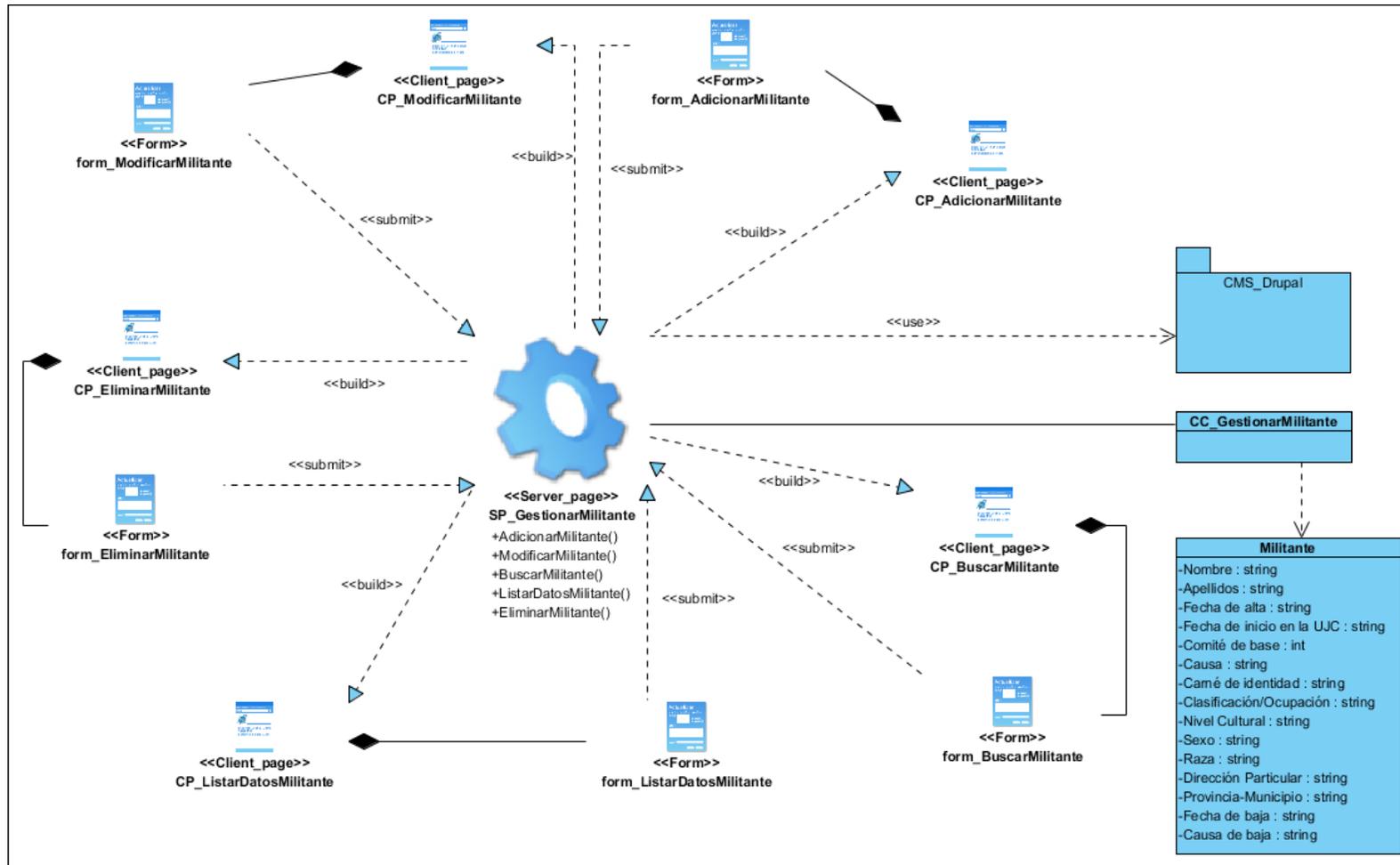


Figura 4 Diagrama de clases del diseño. Paquete Gestionar militante.

2.6.2. Diagrama de Secuencia

Un diagrama de secuencia de un sistema constituye una representación que muestra los eventos generados por actores externos, su orden y los eventos internos del sistema (LARMAN, 1999). A continuación se muestra el diagrama de secuencia del diseño, anexo al trabajo de diploma se encuentran otros de similar importancia.

Figura 5 Diagrama de secuencia. RF Adicionar militante.

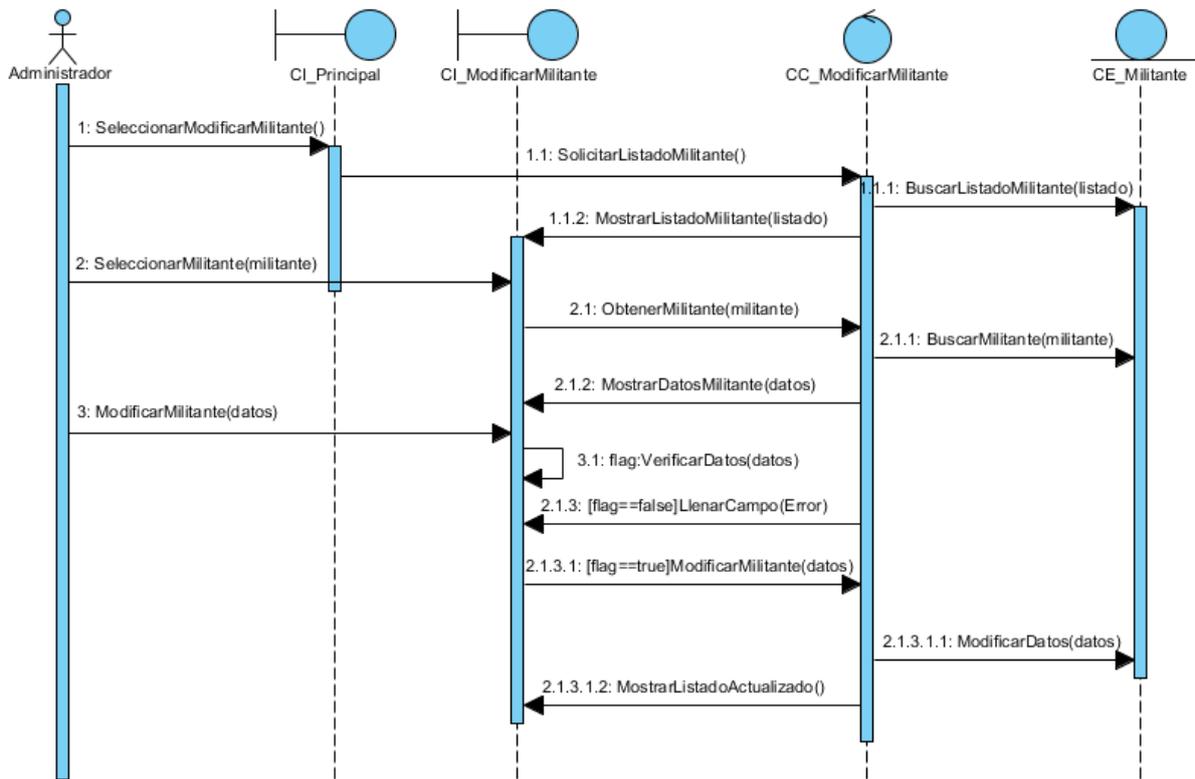


Figura 6 Diagrama de secuencia. RF Modificar militante.

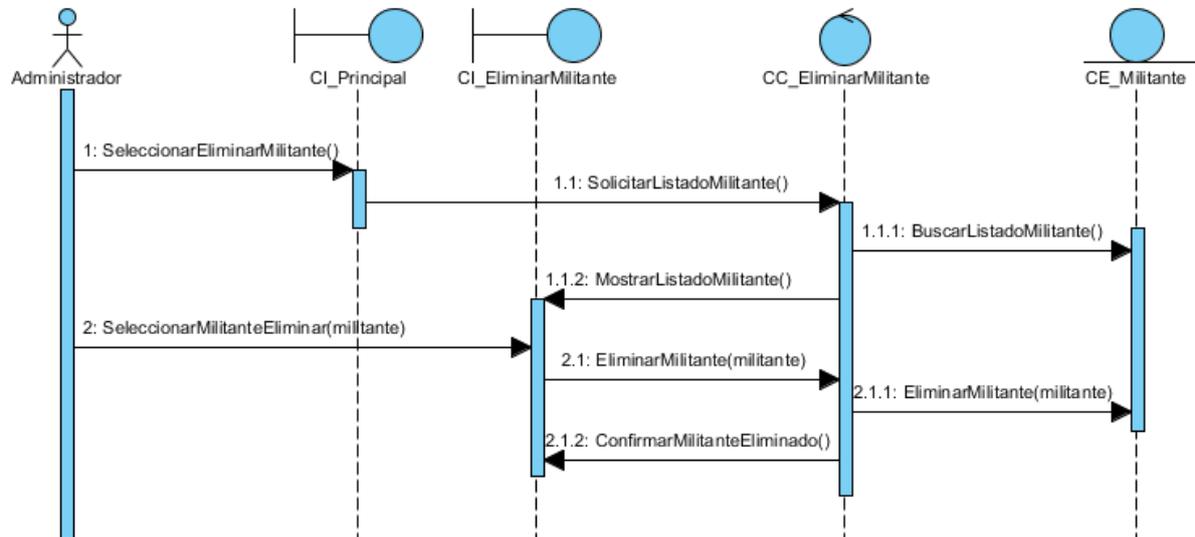


Figura 7 Diagrama de secuencia. RF Eliminar militante.

2.7. Modelo de despliegue

Se compone por un grafo donde los nodos constituyen recursos de cómputo, pudiendo ser dispositivos de hardware o procesadores. Se refleja la configuración de los componentes de hardware y objetos que existen en tiempo de ejecución. La comunicación entre las computadoras clientes y el servidor web se realiza con el protocolo HTTPS. La impresora se

Capítulo II.

conecta por un puerto USB a una computadora cliente. Se muestra un Servidor PostgreSQL para la base de datos, garantizando la disponibilidad y actualización de la información generada por el CMS Drupal. La conexión entre el servidor web y el de base de datos se realiza mediante los protocolos TCP/IP. El servidor de correo se conecta al servidor web mediante el protocolo SMTP.

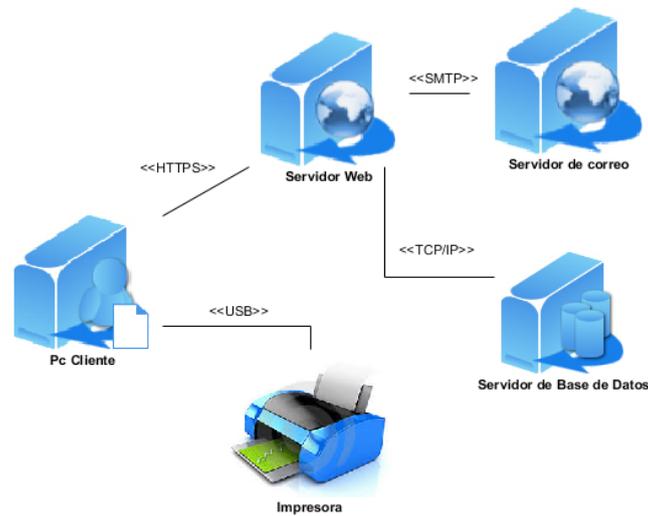


Figura 8 Modelo de despliegue.

La entrevista realizada, la observación y el análisis de los procesos llevados a cabo en el Departamento permitieron definir requisitos funcionales que servirán de guía para desarrollar las diferentes funcionalidades de la propuesta de la solución. Los artefactos generados constituyeron una guía esencial para la construcción de la propuesta de solución. El modelo de despliegue refleja los componentes físicos requeridos para la instalación del sistema.

Capítulo 3: Implementación y prueba del Sistema de Gestión de la Militancia en la UJC de la UCI.

En el presente capítulo se reflejan aspectos de la implementación del Sistema de Gestión de la militancia en la UJC dentro de la UCI, descrito en el modelo de componentes. Se presentan los casos de prueba aplicados, entre los que se encuentran: de seguridad, de funcionalidad y de carga y estrés.

3.1. Diagrama de componentes

Consiste en modelar una vista estática del sistema, se representan las dependencias existentes y la organización de los componentes que conforman el sistema. Generalmente estos diagramas son realizados para fragmentos de la aplicación sin la necesidad de contar con todos los componentes del mismo. A continuación se presenta el diagrama de componentes del sistema de gestión desarrollado:

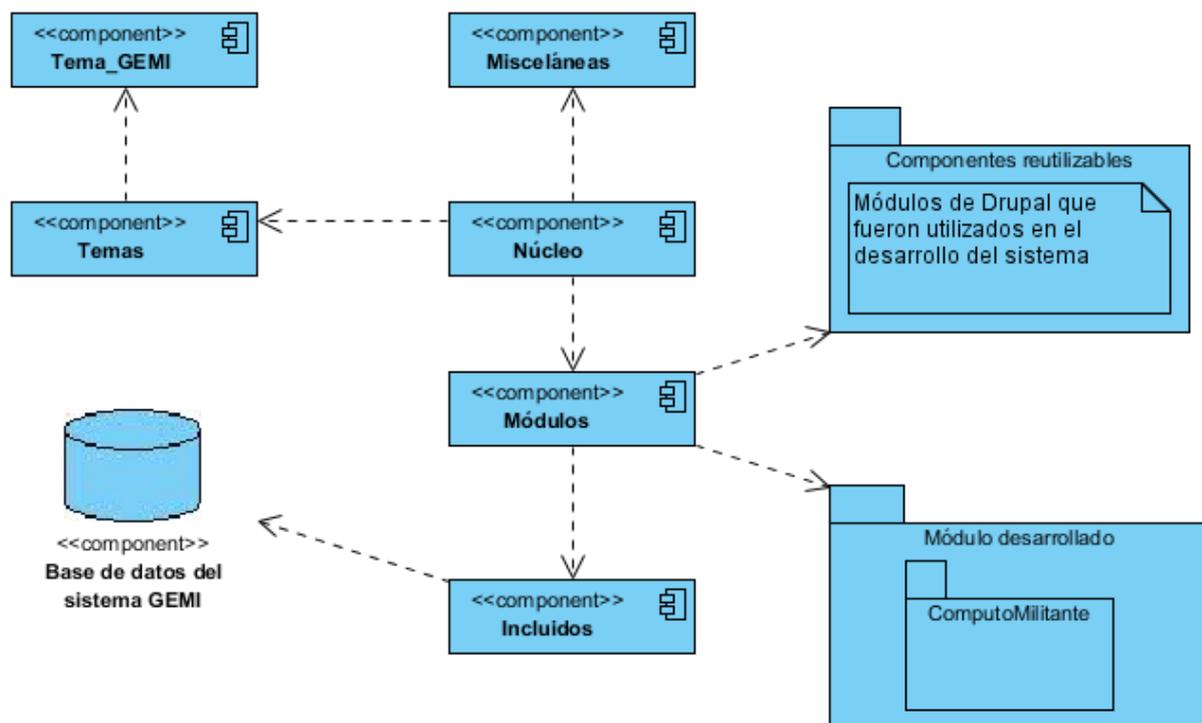


Figura 9 Diagrama de componentes.

Núcleo: constituye el punto de inicio del sistema, desde el que se accede a todos los recursos disponibles.

Temas: agrupa el conjunto de temas que vienen con la distribución de Drupal instalada.

Módulos: espacio donde se almacenan los módulos del CMS, no se deben modificar ni adicionar otros en esa carpeta.

Incluidos: forman el conjunto de ficheros necesarios para el correcto funcionamiento de Drupal.

Misceláneas: agrupa elementos del diseño y el funcionamiento, se destacan archivos JavaScript e imágenes.

Base de datos del sistema GEMI: encargada de gestionar toda la información almacenada en el sitio y vinculada a su funcionamiento.

Tema_GEMI: el tema desarrollado para el sitio de gestión de la militancia en la UJC de la Universidad.

ComputoMilitante: contiene el módulo desarrollado para gestionar los reportes necesarios para la gestión del estado de la militancia en la UJC.

3.2. Estándares de codificación

En la actualidad con el auge y constante transformación de las tecnologías, es necesario que las aplicaciones puedan actualizarse y modificarse, con el propósito de alargar el tiempo de vida de las mismas. Para favorecer las labores de mantenimiento de un software existen normas que ayudan a la comprensión del código para otros programadores. A continuación se exponen algunos estándares que aplica Drupal para una mejor organización de sus sistemas (MICROSOFT, 2003):

Etiquetas de apertura y cierre de PHP: el uso de las etiquetas de apertura y cierre usadas en el lenguaje php.

Punto y coma (;) en código PHP: de obligatorio uso al terminar cada línea, el signo (;).

Variables globales: son declaradas con un guion bajo al inicio, seguido por el nombre del tema o módulo, otro guion bajo y el nombre de la variable.

Indentación: radica en la inspección de espacios en blanco o tabuladores en el código para facilitar su comprensión. En Drupal se usan los espacios aunque se deben evitar al final de cada línea.

Funciones: los nombres de las funciones deben escribirse en minúsculas y separadas por guion bajo las palabras que los conformen. No se dejan espacios entre el nombre y el paréntesis de inicio de los argumentos en la declaración de la función; los argumentos se separan por una coma y un espacio del argumento al que precede.

Constantes: se escriben en mayúsculas y con guiones bajos para separar las palabras.

Nombres de módulos: no deben incluir guiones bajos aunque contengan varias palabras.

URL de ejemplo: por lo general debe mostrarse un ejemplo de URL antes de insertar alguna.

3.3. Modelo de datos entidad-relación

El Modelo de Entidad Relación es un modelo de datos fundamentado en una percepción del mundo real y lo conforman un conjunto de objetos básicos, denominados entidades, que se definen como objetos reales o abstractos enmarcados en un contexto determinado, del cual se almacena alguna información. El modelo consta además de las relaciones entre dichas entidades, formando diagramas o gráficos que describen un proceso determinado.

El siguiente modelo entidad-relación muestra la relación entre el esquema de Drupal y las tablas del sitio web vinculadas al contenido comité de base. Debido a que en la versión 7 de Drupal cada tipo de contenido almacena sus valores en tablas independientes que se relacionan entre sí. La descripción de la tabla se encuentra anexada al trabajo de diploma:

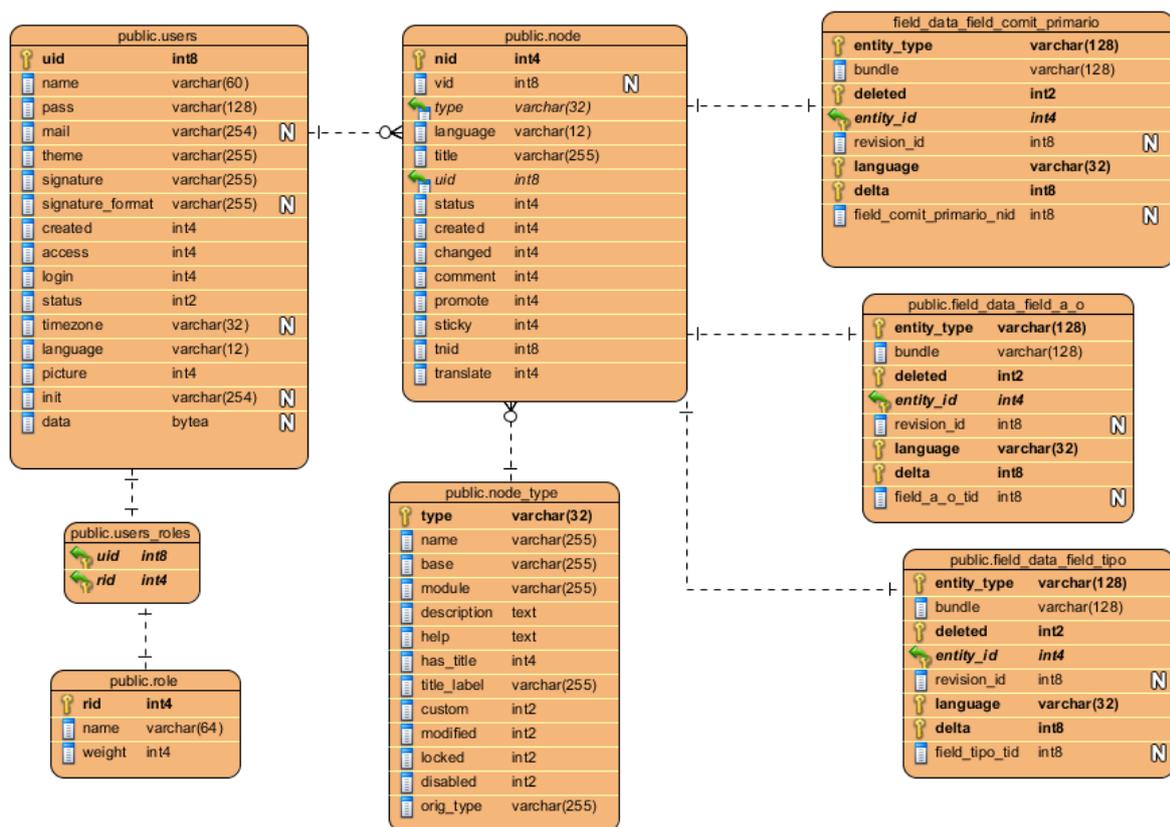


Figura 10 Diagrama entidad-relación.

3.4. Pruebas al sistema

Las pruebas realizadas al software garantizan la calidad del mismo y están encaminadas a corregir las fallas existentes antes de entregar el producto al cliente. Forman parte del ciclo de desarrollo del software y existen varias estrategias para su aplicación. En la presente investigación son documentados los resultados de las pruebas funcionales, de seguridad y de rendimiento (carga o stress), apoyándose en los casos de pruebas creados, la lista de chequeo diseñada por el Centro de Calidad de Software (CALISOFT) y las herramientas *Web Security* y *JMeter*. A continuación se describe cada una de ellas:

Pruebas de caja negra: se enfocan en la interfaz visual del software, evalúan los requisitos funcionales del sistema sin adentrarse en su funcionamiento. Se proporcionan diversas entradas para evaluar la respuesta del software y corregir cualquier deficiencia detectada en la respuesta.

3.4.1. Pruebas funcionales

En el proceso de pruebas del sistema desarrollado, se utilizan técnicas de caja negra, evaluando las funcionalidades de la aplicación desde la interfaz de usuario. Los casos de pruebas para algunos requisitos de alta prioridad aparecen anexados al trabajo de diploma. A continuación se muestran los resultados obtenidos para dichas pruebas:

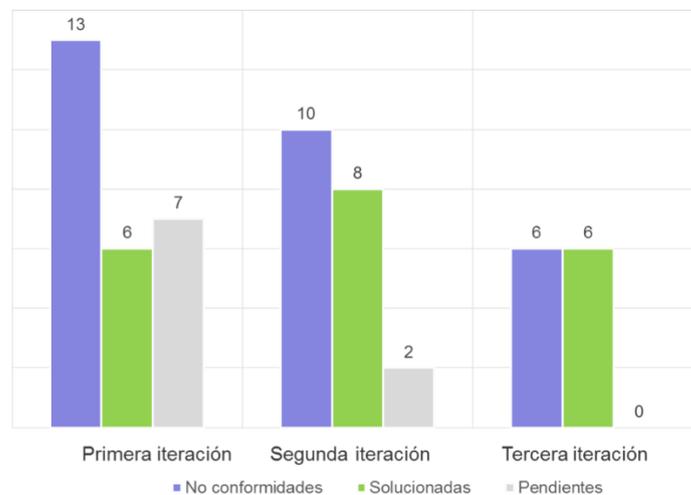


Figura 11 Resultados de las pruebas funcionales.

3.4.2. Pruebas de seguridad

Con el objetivo de garantizar la disponibilidad, integridad y confidencialidad de la información que almacena el Sistema de gestión del estado de la militancia (GEMI), identificando riesgos y amenazas que se presentan en el sistema simulando un ambiente real, evaluando el nivel

del sistema de información ante posibles ataques es necesaria la aplicación de pruebas de seguridad siguiendo la filosofía de Hacking ético¹⁷ (V&V QUALITY, 2013). Se realizan siguiendo una lista de chequeo para pruebas de seguridad diseñada por el Centro de Calidad de Software (CALISOFT), disponible en los anexos del trabajo de diploma, como pruebas de primer nivel.

La lista de chequeo divide las pruebas en 4 secciones diferentes:

Autorización

Sección que asegura que cada usuario pueda realizar solo las operaciones que le corresponden. En la realización de las pruebas referentes a esta sección se evidenció que los usuarios que no posean el rol de administrador no pueden modificar sus privilegios ni los de otros usuarios.

Gestión de sesiones

En esta sección se verifica que el sistema no permita acceder a las funcionalidades luego de haber cerrado la sección. Las pruebas realizadas no arrojaron problemas en este aspecto.

Autenticación

Estas pruebas chequean si el sistema muestra debilidades en el proceso de autenticación de usuarios ante un posible atacante, mostrando información sensible para el sistema. Con la realización de estas pruebas se demostró que el sistema no brindaba información sensible para la UJC.

Validación de datos

Sección encargada de las pruebas que garantizan la correcta entrada de datos y el cumplimiento de los requisitos que pueda tener el sistema para ellos, como pueden ser la longitud de las contraseñas o ausencia de valores alfabéticos en campos de fecha y carné de identidad. No se encontraron problemas en el software luego de aplicar dichas pruebas.

Con el objetivo de identificar vulnerabilidades y puntos de acceso a nivel de infraestructura se escogió la herramienta *Web Security*, para detectar las deficiencias relacionadas con:

- Inyección de SQL.
- *Cross Site Scripting*.

¹⁷ Acto de una persona que utiliza sus conocimientos de informática y seguridad para realizar pruebas en redes y encontrar vulnerabilidades, para luego reportarlas y que se tomen medidas, sin hacer daño.

- Inyección de CRLF.
- Salto de directorio.
- *Hacking* de autenticación.
- Seguridad de aplicaciones AJAX.

Resultados de la Prueba de Seguridad

Posteriormente al análisis del Sistema de gestión del estado de la militancia con la herramienta *Web Security*, se generó un informe con las siguientes vulnerabilidades, de las que se muestra su solución, se muestra textualmente parte del reporte en los anexos del trabajo de diploma:

Tabla 8 Resultados de las pruebas de seguridad.

Vulnerabilidad	Solución
Diferentes rutas del sistema se visualizaron dentro del código fuente de la aplicación cliente. Información que podría utilizarse por un atacante para suponer o discernir el entorno de la aplicación y las debilidades que puedan heredarse con ellas.	Configuraren Apache: directorio alias. Borrar el archivo info.php.
El sistema en determinados escenarios muestra información sensible de la base de datos ante inyecciones de código SQL.	Implantar un parche ¹⁸ para corregir problemas relacionados a inyecciones de código SQL.
Divulgación banner.	Configurar en Apache el fichero: "/etc/apache2/apache2.conf" y cambiar de <i>ServerSignature: On</i> a <i>ServerSignature: Off</i> , también cambiar de <i>ServerTokens: full</i> a <i>ServerTokens: prod</i> . También se puede configurar el fichero: "/etc/apache2/conf.d/security" con los mismos parámetros para PHP: <code>expose_php= Off</code> .

¹⁸ Código, métodos o módulos que realizan cambios que se aplican a un programa, para corregir errores, agregarle funcionalidad, actualizarlo, etc.

3.4.3. Pruebas de rendimiento

Las pruebas de rendimiento comúnmente llamadas de carga y estrés, son utilizadas para evaluar la estabilidad del software en un ambiente simulado con diferentes condiciones, atendiendo a concurrencia, operaciones sobre el sistema por usuario, entre otros. Las pruebas deberán ser lo más fidedignas posibles para poder arribar a conclusiones acertadas y precisas; la simulación se realiza utilizando las funcionalidades al alcance de los usuarios. Independientemente de que se relacionan las pruebas de carga y estrés, se diferencian en su objetivo y la forma en que evalúan el sistema. Las de carga someten al sistema a una carga de trabajo específica y estable durante un tiempo relativamente corto de alrededor de 1 hora. El nivel de carga debe ser alto y continuo para verificar que el sistema soporta esa carga sin que se afecte el servicio y con un tiempo de respuesta estable. Generalmente son aplicadas con varios niveles de carga, se prevé la carga real que tendrá el sistema y se prueba para esa estimación y para cargas mayores, pudiendo analizar el comportamiento del sistema para casos críticos y para condiciones estándares. Las pruebas de estrés en cambio están enfocadas a encontrar el punto de ruptura del sistema: determinar el nivel de carga para el que el sistema pierde el servicio y lo sucedido en los niveles anteriores. Los resultados obtenidos ayudan a tener conocimientos de las peticiones que soporta el sistema sin que se afecte el rendimiento y servicio, posibilitando implantar medidas ante situaciones de esa índole: desviar peticiones a sistemas auxiliares, denegar el acceso a usuarios o activar un protocolo que evite la pérdida del servicio (SOMMERVILLE, 2005).

La herramienta utilizada constituye un factor importante a tener en cuenta, para la realización de las pruebas de rendimiento se utilizó la herramienta *JMeter* en su versión 2.9. Constituye una herramienta diseñada para la medición del rendimiento de aplicaciones, posibilita alterar las diferentes cargas aplicadas al sistema y probar su resistencia.

El ambiente donde fueron realizadas las pruebas de rendimiento se describe a continuación: Una computadora cliente con 2Gb de memoria RAM, un microprocesador Intel *Dual Core* con velocidad 3.0 GHz por núcleo, con el sistema operativo Ubuntu 12.04.

En la elaboración del plan de pruebas de rendimiento para el Sistema de gestión del estado de la militancia se tuvieron en cuenta las acciones que pueden realizar los funcionarios de la UJC que laboran en el Departamento de Documentación y Control de la UJC en la UCI. Se utilizó una muestra de 50 usuarios conectados concurrentes, aunque eso excede en gran medida lo estimado para el ambiente real que no debe alcanzar los 15 usuarios concurrentes, a continuación se muestra el reporte que generó la herramienta *JMeter*.

Capítulo III.

Reporte resumen

Nombre: Reporte resumen

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec	Media de By...
/gemi/modu...	300	6	2	186	15,31	0,00%	4,7/sec	10,17	2223,0
/gemi/sites/...	300	13	1	133	24,87	0,00%	4,7/sec	4,30	940,8
/gemi/sites/...	300	5	2	180	10,74	0,00%	4,7/sec	6,76	1487,0
/gemi/modu...	300	6	2	184	15,09	0,00%	4,6/sec	2,16	479,0
/gemi/modu...	300	6	2	186	11,57	0,00%	4,7/sec	7,37	1623,0
/gemi/modu...	300	5	1	62	5,52	0,00%	4,6/sec	2,01	446,0
/gemi/sites/...	300	5	2	39	4,85	0,00%	4,7/sec	7,17	1559,0
/gemi/misc/...	300	5	2	23	3,09	0,00%	4,7/sec	6,56	1419,0
/gemi/sites/...	50	5	2	22	3,64	0,00%	6,4/sec	38,89	6185,0
/gemi/sites/...	50	7	2	56	10,37	0,00%	6,5/sec	3,43	540,0
/gemi/them...	50	5	3	21	3,24	0,00%	6,6/sec	7,18	1122,0
/gemi/them...	50	6	3	49	6,92	0,00%	6,6/sec	2,40	372,0
/gemi/sites/...	50	11	4	55	8,50	0,00%	6,6/sec	152,16	23553,0
/gemi/them...	300	5	1	55	5,19	0,00%	4,7/sec	3,08	676,8
/gemi/sites/...	300	5	2	42	5,00	0,00%	4,7/sec	11,28	2435,0
/gemi/node/...	100	7370	3358	10360	1480,76	0,00%	1,6/sec	6,83	4390,5
/gemi/misc/...	100	6	2	73	11,04	0,00%	1,8/sec	2,37	1354,0
/gemi/misc/...	100	6	2	56	7,11	0,00%	1,8/sec	4,18	2382,0
/gemi/modu...	100	5	2	24	3,53	0,00%	1,8/sec	3,02	1719,0
/gemi/misc/...	100	5	1	35	4,78	0,00%	1,8/sec	1,84	1043,0
/gemi/busc...	50	14856	10807	16905	1070,42	0,00%	2,0/sec	8,20	4157,9
/gemi/comp...	50	7613	5375	9583	1114,71	0,00%	2,4/sec	11,36	4947,1
/gemi/sites/...	50	147	28	293	66,01	0,00%	4,3/sec	110,21	26268,0
/gemi/sites/...	50	910	542	1294	186,13	0,00%	4,0/sec	320,19	81490,0
/gemi/sites/...	50	7	3	30	4,95	0,00%	4,3/sec	18,26	4382,0
/gemi/busc...	50	14225	10927	16120	1345,52	0,00%	1,9/sec	7,73	4254,6
Total	10100	300	1	16905	1754,81	0,00%	142,3/sec	414,67	2984,5

Incluir el nombre del grupo en la etiqueta? Guardar la cabecera de la tabla

Figura 12 Reporte resumen generado por la aplicación JMeter.

Para una mejor comprensión del reporte generado se describen los parámetros de la tabla:

Etiqueta: el nombre de la muestra (conjunto de muestras).

#Muestras: el número de muestras para cada URL.

Media: el tiempo medio transcurrido para un conjunto de resultados.

Mín: el mínimo tiempo transcurrido para las muestras de la URL dada.

Máx: el máximo tiempo transcurrido para las muestras de la URL dada.

%Error: porcentaje de las peticiones con errores.

Rendimiento: rendimiento medido en base a peticiones por segundo/minuto/hora.

Kb/sec: rendimiento medido en Kilobytes por segundo.

MediadeBytes: tamaño medio de la respuesta de la muestra medido en bytes

Como el resumen generado por la herramienta JMeter muestra, para un total de 10100 muestras que se le realizaron al sistema se alcanzó un rendimiento de 142,3 peticiones por segundo, con 0% de errores para cada petición realizada. Se arriba a que el sistema responde correctamente ante situaciones de carga y estrés.

Capítulo III.

Las funcionalidades desarrolladas en el sistema cumplen con las necesidades de gestión de la información llevada a cabo en el Departamento de Documentación y Control de la UJC. Las pruebas funcionales, de seguridad y de rendimiento realizadas al producto permitieron identificar y corregir no conformidades mejorando así la calidad del sistema en su versión final.

Conclusiones

El presente trabajo científico, dio cumplimiento a cada uno de los objetivos trazados, permitiendo arribar a las siguientes conclusiones:

- El estudio de diferentes aplicaciones en el ámbito nacional e internacional permitió obtener las características comunes en el desarrollo de las mismas, las cuales fueron tenidas en cuenta a la hora de realizar la propuesta y seleccionar las tecnologías y herramientas a utilizar en el Sistema de gestión del estado de la militancia de la Unión de Jóvenes Comunistas para la Universidad de Ciencias Informáticas.
- Las actividades y artefactos generados, basados en la metodología OpenUP, posibilitaron una correcta extracción y elaboración de los requisitos con vistas a la implementación del sistema.
- Se diseñaron casos de prueba, permitiendo evaluar, valorar y perfeccionar la calidad del sistema, logrando que la versión final cumpla con las especificaciones del diseño y de la codificación.
- Las pruebas realizadas al software permitieron validar el correcto funcionamiento del Sistema de gestión del estado de la militancia de la Unión de Jóvenes Comunistas en la Universidad de Ciencias Informáticas.

Recomendaciones

Una vez finalizado el desarrollo del Sistema de gestión del estado la militancia de la Unión de Jóvenes Comunistas en la Universidad de Ciencias Informáticas, vencidos todos los objetivos trazados y tomando el mismo como la primera versión que responde a las necesidades del Departamento de Documentación y Control de la UJC, se recomienda al Centro de Ideoinformática para futuras versiones del sistema:

- Agregar la funcionalidad de generar gráficos estadísticos.
- Realizar mejoras a la apariencia visual del sistema.

Referencias Bibliográficas

1. AGUDELO RESTREPO, D. y CARMONA RUIZ, L. M. (2013). Prototipo Web Educativo. 2013.
2. AGUIRRE VILLARREAL, D. R. y VELASTEGUÍ CARRERA, A. F. (2012). Artículo Científico-Análisis, diseño y desarrollo de un curso de Java a nivel básico (Netbeans y Eclipse) para el personal técnico del ejército utilizando la plataforma de teleformación MOODLE y herramientas Web 2.0. 2012.
3. ANDREU, R.; RICART, J., *et al.* (1991). *Estrategia y sistema de información; Information system and strategy*. McGraw-Hill, 1991. ISBN 8476156669.
4. BELL, J. (2010). *Doing your research project*. McGraw-Hill International, 2010. ISBN 0335235824.
5. BÖCK, H. (2009). *The Definitive Guide to NetBeans Platform*. Apress, 2009. ISBN 1430224185.
6. BRAGDON, A.; REISS, S. P., *et al.* (2010). Code bubbles: rethinking the user interface paradigm of integrated development environments. En *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1. 2010*. p. 455-464.
7. BUTCHER, M.; DUNLAP, G., *et al.* (2010). *Drupal 7 Module Development*. 2010, ISBN 978-1-849511-16-2.
8. CASTRO DIAZ-BALART, F. (2004). Ciencia, Tecnología y Sociedad. Hacia un desarrollo sostenible en la era de la Globalización. *Editorial Científico-Técnica*, 2004.
9. CASTRO RUZ, F. (2009). *Hay que darlo todo* Alternativa Bolivariana para las Américas (ALBA): CUBADEBATE, (Reflexiones de Fidel). Disponible en: <http://www.cubadebate.cu/reflexiones-fidel/2009/05/01/hay-que-darlo-todo/>.
10. COBO, Á. (2005). *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*. Ediciones Díaz de Santos, 2005. ISBN 8479787066.
11. CHAVIANO GÓMEZ, Y. y HERNÁNDEZ GONZÁLEZ, A. (2010). HERRAMIENTAS AUTOMATIZADAS PARA LA GESTIÓN DE PROYECTOS. *Ingeniería Industrial*, 2010, vol. 27, nº 2-3, p. 8 pág. ISSN 1815-5936.
12. DAVIS, W. S. y MATA, A. (1992). *Herramientas CASE: metodología estructurada para el desarrollo de los sistemas*. Thomson-Paraninfo, 1992. ISBN 8428319278.
13. DÍAZ, M. F. y SÁNCHEZ, E. M. F. (2009). *Sistema Integral de Gestión de la UJC UCI: Módulo Militante*. Tutor: Pérez, I. Y. W. S. Tesis de Diplomado, Universidad de las Ciencias Informáticas, 2009.
14. DOMÍNGUEZ-DORADO, M. (2004). *Introducción a las aplicaciones web con ASP e IIS*. Madrid: Editorial Iberprensa, 2004, vol. Nº 1, 24-26 p.
15. EGUILUZ, J. (2013). *Desarrollo web ágil con Symfony2*. easybook, 2013, 618 p.
16. ESCOBAR, J.; LOSAVIO, F., *et al.* (2012). Una revisión de Frameworks, Lenguajes de Modelado y Herramientas para Arquitecturas Empresariales. 2012, nº ISSN 9801255552.
17. EVANS, K.; HALLOCK, P., *et al.* (2003). *Database modeling with Microsoft visio for enterprise architects*. Morgan Kaufmann, 2003. ISBN 1558609199.
18. FERNÁNDEZ, I. G.; FIGUEROA, R. C., *et al.* (2011). Propuesta de arquitectura de una herramienta web para la administración del gestor PostgreSQL. *Revista Cubana de Ciencias Informáticas*, 2011, vol. 5, nº 1, ISSN 2227-1899.
19. GÉNOVA, G.; FUENTES, J. M., *et al.* (2009). Evaluación de herramientas CASE para UML. 2009.
20. GILFILLAN, I. (2003). *MySQL*. 2003. ISBN 8441515581.
21. GISBERT, M. D. T. Y. y GROSS, R. G.-P. Y. (1974). *Pequeño Larousse Ilustrado*. Editorial Pueblo y Educación ed. calle 15 No. 604, Vedado, La Habana: 1974, [Consultado el: 1/12/2013]. 1663 p.

Referencias bibliográficas

22. GONZÁLEZ CABRERA, L. y TORRES, E. R. P. (2012). Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información. *Serie Científica*, 2012, vol. 5, n° 10,
23. GONZÁLEZ, Z.; SANOJA, A., *et al.* (2010). UTILIZACIÓN DE MÉTRICAS DE SOFTWARE PARA APOYAR LA SELECCIÓN DE FRAMEWORKS WEB PARA EL SISTEMA DE GESTIÓN DE DATOS ACADÉMICOS DE LA FACULTAD DE CIENCIAS UCV. 2010.
24. GOODMAN, D.; MORRISON, M., *et al.* (2007). *Javascript® bible*. John Wiley & Sons, Inc., 2007. ISBN 0470069163.
25. GRANADO, L. M. C. (2004). *PHP 5*. 2004. ISBN 8441517851.
26. GRUPO CF DEVELOPER (2013). *DocCF, Software de Gestión Escolar Colombia*: [Consultado el: 11 de 2013]. Disponible en: <http://www.grupocfdeveloper.com>.
27. JACOBSON, I., BOOCH, G., RUMBAUGH, J. Y ADDISON, W (2000). *El PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE*. Traducido por: Salvador Sánchez, M. a. S., Carlos Canal, Francisco Javier Durán. <http://www.pgadmin.org/>: 2000. vol. 2014, ISBN 84-7829-036-2.
28. KUROSE, J. F. y ROSS, K. W. (2004). *REDES DE COMPUTADORAS*. 2004. ISBN 8478290613.
29. LARMAN, C. (1999). *UML y Patrones*. Pearson, 1999. ISBN 8420534382.
30. MAGICDRAW, U. (2013). No Magic Inc. URL:<http://www.magicdraw.com>, 2013.
31. MANGER, J. J.; SOTO, S. F., *et al.* (1997). *Fundamentos de JavaScript*. McGraw-Hill, 1997. ISBN 9701017633.
32. MATO, R. M. G. (2005). *Sistemas de Bases de Datos*. Editado por: Pantoja, I. J. Q. Segunda Edición ed. Editorial Pueblo y Educación, 2005. 165 p. ISBN 959-13-1273-3.
33. MICROSOFT (2003). *Revisiones de código y estándares de codificación de 2014*. Disponible en: <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.
34. MORA, S. L. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Editado por: Universitario., E. C. 2002,
35. MORALES, Y. R. (2012). Plugin de la herramienta Visual Paradigm para la evaluación del diseño Orientado a Objetos. *Serie Científica*, 2012, vol. 5, n° 9,
36. ORGANO JUDICIAL DE PANAMÁ (2013). *Sistema Automatizado de Gestión Judicial (SAGJ)* Panamá: [Consultado el: 10/12/2013 de 2013]. Disponible en: <http://www.organojudicial.gob.pa/>.
37. ORTEGA, A. L. (2003). Cherokee Web server. *Actas de las III Jornadas Andaluzas de Software Libre*, 2003, n° p. 21.
38. PHP GROUP (2006). *PHP: Hypertext preprocessor*. 2006,
39. PONJUÁN DANTE, G. (2004). Gestión de información: dimensiones e implementación para el éxito organizacional. *Rosario: Nuevo Paradigma*, 2004, n°
40. POSTGRESQL DEVELOPMENT TEAM (2013). *PostgreSQL Programmer's Guide*. 2013,
41. RAMIREZ, Y. A. y GONZÁLEZ, A. M. (2009). *Sistema Integral de Gestión de la UJC Nacional: Módulo Militante*. Tutor: Palomino, I. a. M. Tesis de Diplomado, Universidad de las Ciencias Informáticas, 2009.
42. REVÉ, G. R. (2009). Unión de Jóvenes Comunistas (UJC) 05 2009, n° [Consultado el: 07/11/2013]. Disponible en: <http://www.solvision.cu/index.php/20090805850/hipertextos/union-de-jovenes-comunistas-ujc.html>.
43. RODRÍGUEZ, F. G. (2012). *Experto en Drupal 7 Nivel avanzado*. Editado por: S.L, F. publicado el: 1/11/2013 de 2012, última actualización: 1/11/2013. vol. 3, 507 p. Disponible en: <http://www.forcontu.com>. ISBN 978-84-939410-5-5.
44. SAMPIERI, R. H.; COLLADO, C. F., *et al.* (2006). *Metodología de la Investigación*. Mexico D.F: 2006. 882 p. ISBN 907-10-5753-8.

Referencias bibliográficas

45. ŠILINGAS, D. y BUTLERIS, R. (2009). Towards implementing a framework for modeling software requirements in MagicDraw UML. *Information Technology and Control*, 2009, vol. 38, nº 2, p. 153-164.
46. SOMMERVILLE, I. (2005). *Ingeniería del software*. Pearson Educación, 2005. ISBN 8478290745.
47. THE-APACHE-SOFTWARE-FOUNDATION (2013). *HTTP Apache Server Project* [Consultado el: 20/ 12 de 2013]. Disponible en: <https://httpd.apache.org/>.
48. THE BRITISH STANDARDS INSTITUTION (BSI), A. M. (2013). *Entropy software* BSI Group, Disponible en: <http://www.bsi-entropy.com/es/explore-entropy/modules/audit-compliance-management/bsi-action-manager/>.
49. TOMLINSON, T. (2010). *Beginning Drupal 7* Editado por: Michelle Lowman, B. M. Paul Manning 2010, Disponible en: www.springeronline.com. ISBN 978-1-4302-2860-8.
50. V&V QUALITY (2013). *Pruebas de seguridad* CO231599. ed. [Consultado el: 15-04 de 2014]. Disponible en: http://www.vyvquality.com/w1/index.php?option=com_content&view=article&id=79&Itemid=151.
51. VELASQUEZ, G. y ROMAN, D. (2009). Modulo de control de avance del sistema de información de notas y calificaciones COLMIS. 2009, nº
52. VISUAL PARADIGM (2011). Visual Paradigm. *Online www. visual-paradigm. com*. Last November, 2011.

Bibliografía

Achour, M., F. Betz, et al. (2006). PHP Manual, The PHP Documentation Group.

Agudelo Restrepo, D. and L. M. Carmona Ruiz (2013). "Prototipo Web Educativo."

Aguirre Villarreal, D. R. and A. F. Velasteguí Carrera (2012). "Artículo Científico-Análisis, diseño y desarrollo de un curso de Java a nivel básico (Netbeans y Eclipse) para el personal técnico del ejército utilizando la plataforma de teleformación MOODLE y herramientas Web 2.0."

Aja Quiroga, L. (2002). "Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones." Acimed **10**(5): 7-8.

Andreu, R., J. Ricart, et al. (1991). Estrategia y sistema de información; Information system and strategy, McGraw-Hill.

Bell, J. (2010). Doing your research project, McGraw-Hill International.

Böck, H. (2009). The Definitive Guide to NetBeans Platform, Apress.

Booch, G., J. Rumbaugh, et al. (1999). El lenguaje unificado de modelado, Addison-Wesley.

Bragdon, A., S. P. Reiss, et al. (2010). Code bubbles: rethinking the user interface paradigm of integrated development environments. Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1, ACM.

Butcher, M., G. Dunlap, et al. (2010). Drupal 7 Module Development.

Cantor, M. and R. Plug (2003). "Rational Unified Process for Systems Engineering: Part 1—Introducing RUP SE Version 2.0." The Rational Edge (August 2003).

Castro Diaz-Balart, F. (2004). "Ciencia, Tecnología y Sociedad. Hacia un desarrollo sostenible en la era de la Globalización." Editorial Científico-Técnica.

Castro Ruz, F. (2009). "Hay que darlo todo." Reflexiones de Fidel. Tomado de <http://www.cubadebate.cu/reflexiones-fidel/2009/05/01/hay-que-darlo-todo/>.

Chaviano Gómez, Y. and A. Hernández González (2010). "HERRAMIENTAS AUTOMATIZADAS PARA LA GESTIÓN DE PROYECTOS." Ingeniería Industrial **27**(2-3): 8 pág.

Cobo, Á. (2005). PHP y MySQL: Tecnología para el desarrollo de aplicaciones web, Ediciones Díaz de Santos.

Corlosquet, S., R. Delbru, et al. (2009). Produce and Consume Linked Data with Drupal! The Semantic Web-ISWC 2009, Springer: 763-778.

Davis, W. S. and A. Mata (1992). Herramientas CASE: metodología estructurada para el desarrollo de los sistemas, Thomson-Paraninfo.

Díaz, M. F. and E. M. F. Sánchez (2009). Sistema Integral de Gestión de la UJC UCI: Módulo Militante. Cuba, Universidad de las Ciencias Informáticas: 411.

Bibliografía

- Domínguez-Dorado, M. (2004). Introducción a las aplicaciones web con ASP e IIS. Todo Programación. Madrid, Editorial Iberprensa. Nº 1: 24-26.
- Equiluz, J. (2013). Desarrollo web ágil con Symfony2, easybook: 618.
- Escalona, M. J. and N. Koch (2002). "Ingeniería de Requisitos en Aplicaciones para la Web— Un estudio comparativo." Universidad de Sevilla.
- Escobar, J., F. Losavio, et al. (2012). "Una revisión de Frameworks, Lenguajes de Modelado y Herramientas para Arquitecturas Empresariales."
- Evans, K., P. Hallock, et al. (2003). Database modeling with Microsoft visio for enterprise architects, Morgan Kaufmann.
- Fernández, I. G., R. C. Figueroa, et al. (2011). "Propuesta de arquitectura de una herramienta web para la administración del gestor PostgreSQL." Revista Cubana de Ciencias Informáticas 5(1).
- Génova, G., J. M. Fuentes, et al. (2009). "Evaluación de herramientas CASE para UML."
- Gilfillan, I. (2003). MySQL.
- Gisbert, M. d. T. y. and R. G.-P. y. Gross (1974). Pequeño Larousse Ilustrado. calle 15 No. 604, Vedado, La Habana: 1663.
- González Cabrera, L. and E. R. P. Torres (2012). "Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información." Serie Científica 5(10).
- González, Z., A. Sanoja, et al. (2010). "UTILIZACIÓN DE MÉTRICAS DE SOFTWARE PARA APOYAR LA SELECCIÓN DE FRAMEWORKS WEB PARA EL SISTEMA DE GESTIÓN DE DATOS ACADÉMICOS DE LA FACULTAD DE CIENCIAS UCV."
- Goodman, D., M. Morrison, et al. (2007). Javascript® bible, John Wiley & Sons, Inc.
- Graf, H. (2006). Building Websites with Joomla! Birmingham, Reino Unido, Packt Publishing, 323p.
- Granado, L. M. C. (2004). PHP 5.
- Grupo CF Developer (2013). "DocCF, Software de Gestión Escolar." Retrieved 11, 2013, tomado de <http://www.grupocfdeveloper.com>.
- Halili, E. H. (2008). Apache JMeter: A practical beginner's guide to automated testing and performance measurement for your websites, Packt Publishing Ltd.
- Jacobson, I., Booch, G., Rumbaugh, J. y Addison, W (2000). EI PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE. <http://www.pgadmin.org/>.
- Kruchten, P. (2004). The rational unified process: an introduction, Addison-Wesley Professional.
- Kurose, J. F. and K. W. Ross (2004). REDES DE COMPUTADORAS.
- Larman, C. (1999). UML y Patrones, Pearson.

Bibliografía

- Lockhart, T. (2000). "PostgreSQL User's Guide." PostgreSQL Inc.
- Magicdraw, U. (2013). "No Magic Inc." URL:{<http://www.magicdraw.com>}.
- Manger, J. J., S. F. Soto, et al. (1997). Fundamentos de JavaScript, McGraw-Hill.
- Martin, J., J. J. Odell, et al. (1994). Análisis y diseño orientado a objetos, Prentice Hall Hispanoamericana.
- Mato, R. M. G. (2005). Sistemas de Bases de Datos, Editorial Pueblo y Educación.
- Microsoft (2003). "Revisiones de código y estándares de codificación." 2014, tomado de <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.
- Momjian, B. (2001). PostgreSQL: introduction and concepts, Addison-Wesley New York.
- Montiel, M. F. (2013). "Prácticas Profesionales II."
- Mora, S. L. (2002). Programación de aplicaciones web: historia, principios básicos y clientes web. E. C. Universitario.
- Morales, Y. R. (2012). "Plugin de la herramienta Visual Paradigm para la evaluación del diseño Orientado a Objetos." Serie Científica 5(9).
- Organo Judicial de Panamá (2013). "Sistema Automatizado de Gestión Judicial (SAGJ)." Retrieved 10/12/2013, 2013, tomado de <http://www.organojudicial.gob.pa/>.
- Ortega, A. L. (2003). "Cherokee Web server." Actas de las III Jornadas Andaluzas de Software Libre: 21.
- PHP Group (2006). PHP: Hypertext preprocessor.
- Ponjuan Dante, G. (2007). "Gestión de Información en las Organizaciones."
- Ponjuán Dante, G. (2004). "Gestión de información: dimensiones e implementación para el éxito organizacional." Rosario: Nuevo Paradigma.
- PostgreSQL Development Team (2013). PostgreSQL Programmer's Guide.
- Quintero, J. B., R. A. de Páez, et al. (2012). "Un estudio comparativo de herramientas para el modelado con UML." revista universidad eafit 41(137): 60-76.
- Rahmel, D. (2008). Joomla!
- Ramirez, Y. A. and A. M. González (2009). Sistema Integral de Gestión de la UJC Nacional: Módulo Militante. Cuba, Universidad de las Ciencias Informáticas. **Trabajo de Diploma**: 399.
- Revé, G. R. (2009). "Unión de Jóvenes Comunistas (UJC) ".
- Rodríguez, F. G. (2012). Experto en Drupal 7 Nivel avanzado. Curso de creación y gestión de portales web con Drupal 7. F. S.L. 3: 507.
- Rojas, T., M. Pérez, et al. (2000). "Modelo de decisión para soportar la selección de herramientas CASE." Revista de la Facultad de Ingeniería, UCV 15(2): 117-144.

Bibliografía

- Sampieri, R. H., C. F. Collado, et al. (2006). Metodología de la Investigación. Mexico D.F.
- Sanchez, J. and J. Educación-Primaria de la Frontera (2007). "Trabajo Colaborativo." Universidad de Chile. Gobierno de Chile. Ministerio de Educación. Visited on (October 2006) www.mmpchile.cl/pag/press/Trabajo.
- Šilingas, D. and R. Butleris (2009). "Towards implementing a framework for modeling software requirements in MagicDraw UML." Information Technology and Control **38**(2): 153-164.
- Sommerville, I. (2005). Ingeniería del software, Pearson Educación.
- The British Standards Institution (BSI), A. M. (2013). "Entropy software." tomado de <http://www.bsi-entropy.com/es/explore-entropy/modules/audit-compliance-management/bsi-action-manager/>.
- The-Apache-Software-Foundation (2013). "HTTP Apache Server Project." Retrieved 20/ 12, 2013, tomado de <https://httpd.apache.org/>.
- Tomlinson, T. (2010). Beginning Drupal 7 B. M. Michelle Lowman, Paul Manning
- V&V Quality (2013). "Pruebas de seguridad." CO231599. Retrieved 15-04, 2014, tomado de http://www.vyvquality.com/w1/index.php?option=com_content&view=article&id=79&Itemid=151.
- Valdés, D. P. (2007) ¿Que es JavaScript?
- Velasquez, G. and D. Roman (2009). "Modulo de control de avance del sistema de información de notas y calificaciones COLMIS."
- Visual Paradigm (2011). "Visual Paradigm." Online [www. visual-paradigm. com](http://www.visual-paradigm.com). Last November.

Anexos

Anexo 1: Entrevista realizada el 7/11/2013 al Jefe del Departamento de Documentación y Control de la UJC de la Universidad de Ciencias Informáticas, Ing. Ramón Paumier Samón.

Entrevistador: ¿Qué informaciones gestionan los funcionarios del Departamento vinculados a la organización?

Entrevistado: El departamento de documentación y Control de la UJC de la UCI se encarga de la gestión y el control del estado de la militancia. Estadísticamente hablando, genera reportes relacionados con la cantidad de comités de base, el número de militantes en cada uno de ellos, las informaciones solicitadas por el municipio. Controla a nivel de universidad los movimientos de los militantes de la UJC, los traslados, las incorporaciones, las sanciones.

Entrevistador: ¿Han existido sistemas informáticos destinados a la automatización del proceso?

Entrevistado: Sí han existido con anterioridad sistemas para el trabajo con la UJC, pero estaban pensados para generar información y reportes a nivel provincial e incluso de país.

Entrevistador: ¿Han cumplido con las expectativas del Departamento? ¿Por qué?

Entrevistado: No han cumplido con las expectativas pues los reportes que generaba no cumplían con las necesidades del departamento, teniendo en cuenta que la UCI es un centro con derechos y por consiguiente, se encarga del control y la generación de sus propios datos. Incluso el primero de los sistemas que se utilizó, luego de haber introducido los datos de todos los militantes del centro, hubo que dejarlo a un lado pues su base de datos no era compatible con la de un nuevo sistema que se estaba desarrollando.

Entrevistador: ¿En el presente, cómo se registran los datos de los militantes, los comités de base y comités primarios?

Entrevistado: El proceso se realiza completamente de manera manual y utilizando tablas de Excel.

Entrevistador: ¿Son generadas por sistemas automatizados las tablas que reflejan el resultado de los movimientos de los militantes y su estado actual? ¿Qué costo (hombres/hora) se requiere para la realización de las mismas?

Entrevistado: Los trabajadores del departamento registran, día tras día, de manera manual, los cambios que se producen en los distintos comités de base de cada una de las facultades, en un documento o registro denominado respaldo. Una vez al mes, el respaldo se desglosa en forma numérica en las tablas del cierre y luego se chequea comité a comité, que los números finales coincidan con la cantidad de militantes de la UJC en las plantillas de cada uno de los comités.

Anexo 2: Guía de observación.

- Visitar el Departamento de Documentación y Control de la UJC, para determinar las gestiones llevadas a cabo por dicho departamento.
- Identificar los datos referentes a los militantes, comités de base y comités primarios.
- Estudiar los reportes generados por el departamento y clasificarlos según corresponda su alcance.
- Identificar la concurrencia que tendría el sistema que se instale.
- Clasificar los diferentes roles del personal que labora en el departamento para determinar sus permisos en el sistema.
- Definir las acciones que realizan los trabajadores del departamento sobre los militantes.

Anexo 3: Descripción de los requisitos funcionales.

En el presente apartado se reflejan las descripciones de los requisitos funcionales del sistema. Son presentados solo aquellos en los que los usuarios inciden directamente puesto que un número considerable de ellos son generados dinámicamente y se conforman con valores de los que se describen en el presente documento.

Tabla 9 Descripción de requisitos. Adicionar comité primario.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF1	Adicionar comité primario	El sistema debe permitir adicionar un comité primario	Media	Alta
Prototipo				

Comité de base *

Tipo *

- Seleccione un valor -

Año *

- Seleccione un valor -

Activado *

Comité primario *

- Seleccione un valor -

Campos	Tipos de Datos	Reglas o Restricciones
Comité de base	Texto	Puede contener cualquier cadena alfa-numérica
Tipo	Texto	Solo puede ser: Estudiante, Profesor o Trabajador no docente
Año	Número	Puede contener cualquier número entero entre 0 y 6
Activado	Booleano	Representa si el comité está activado o no
Comité primario	Texto	Se refiere al comité primario al que pertenece
Observaciones	Los * de color rojo indican campos obligatorios.	

Tabla 10 Descripción de requisitos. Eliminar comité primario.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente				
RF1	Eliminar comité primario	El sistema debe permitir eliminar los comités primarios	Media	Alta				
Prototipo								
<input type="checkbox"/>	TÍTULO	TIPO	AUTOR	ESTADO	ACTUALIZADO	<input type="button" value="v"/>	IDIOMA	OPERACIONES
<input type="checkbox"/>	fac3	Comité Primario	admin	publicado	20/02/2014 - 10:47		Español	editar eliminar
<input type="checkbox"/>	Fac1	Comité Primario	admin	publicado	03/12/2013 - 10:41		Español	editar eliminar
Campos		Tipos de Datos		Reglas o Restricciones				

Anexos

	No procede	No procede	No procede
	Observaciones		

Anexo 4: Diagramas de clases.

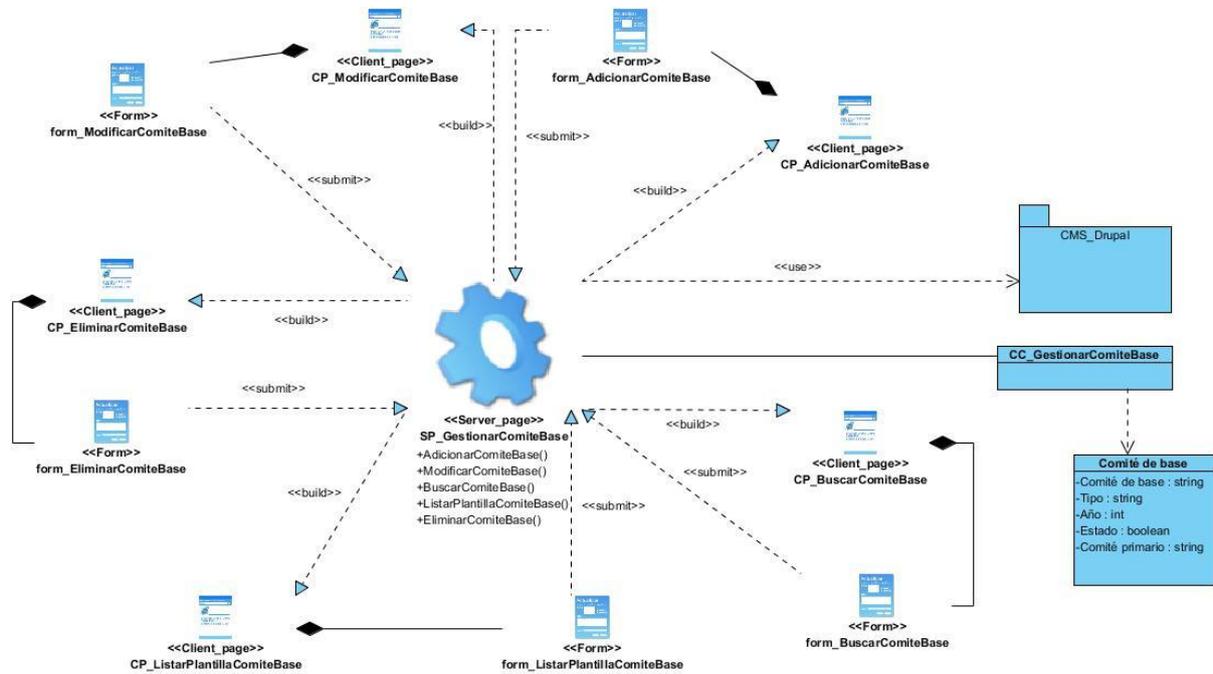


Figura 13 Diagrama de clases de diseño. Paquete Gestionar Comité de base.

Anexos

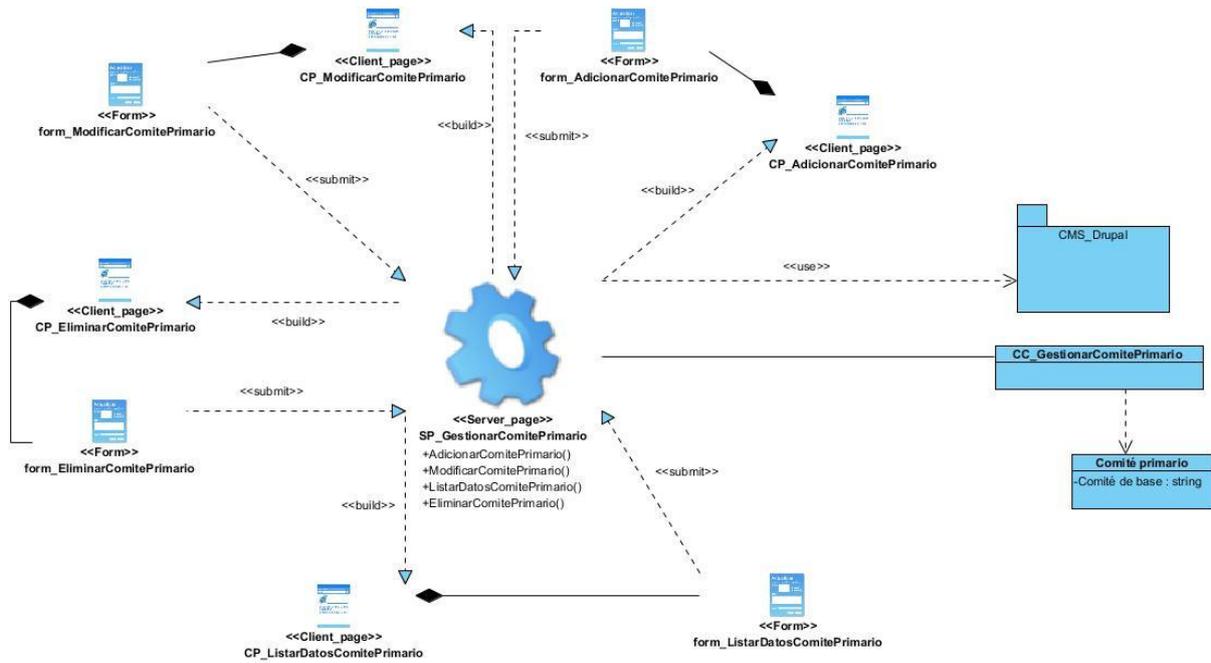


Figura 14 Diagrama de clases de diseño. Paquete Gestionar Comité primario.

Anexo 5: Diagramas de secuencia.

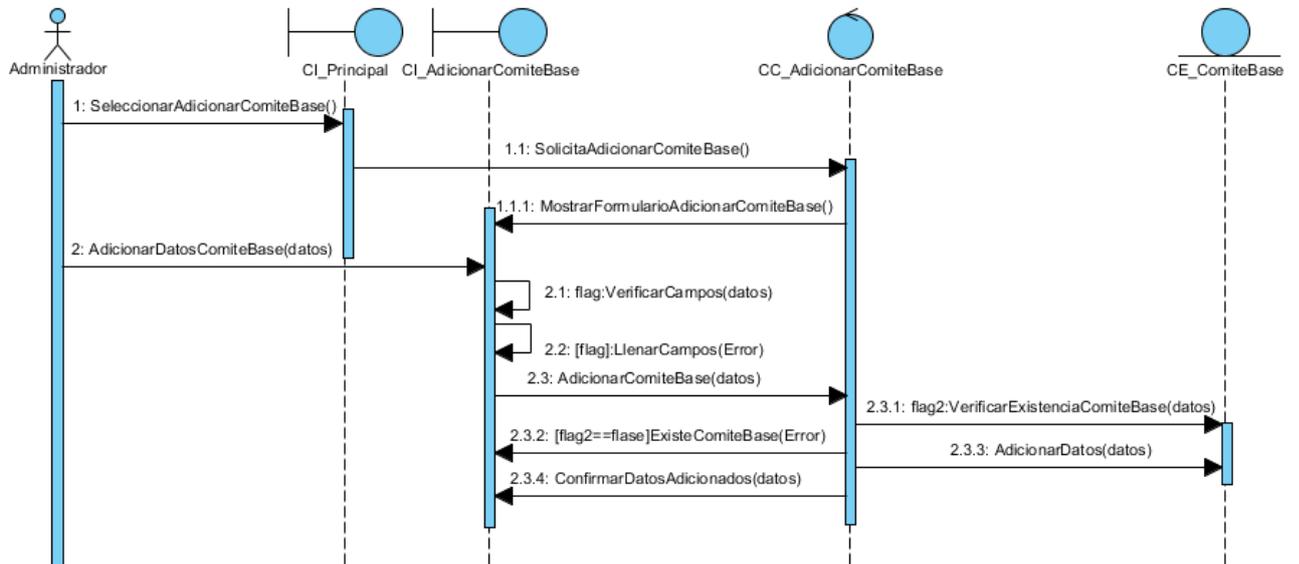


Figura 15 Diagrama de secuencia. RF Adicionar Comité de base.

Anexos

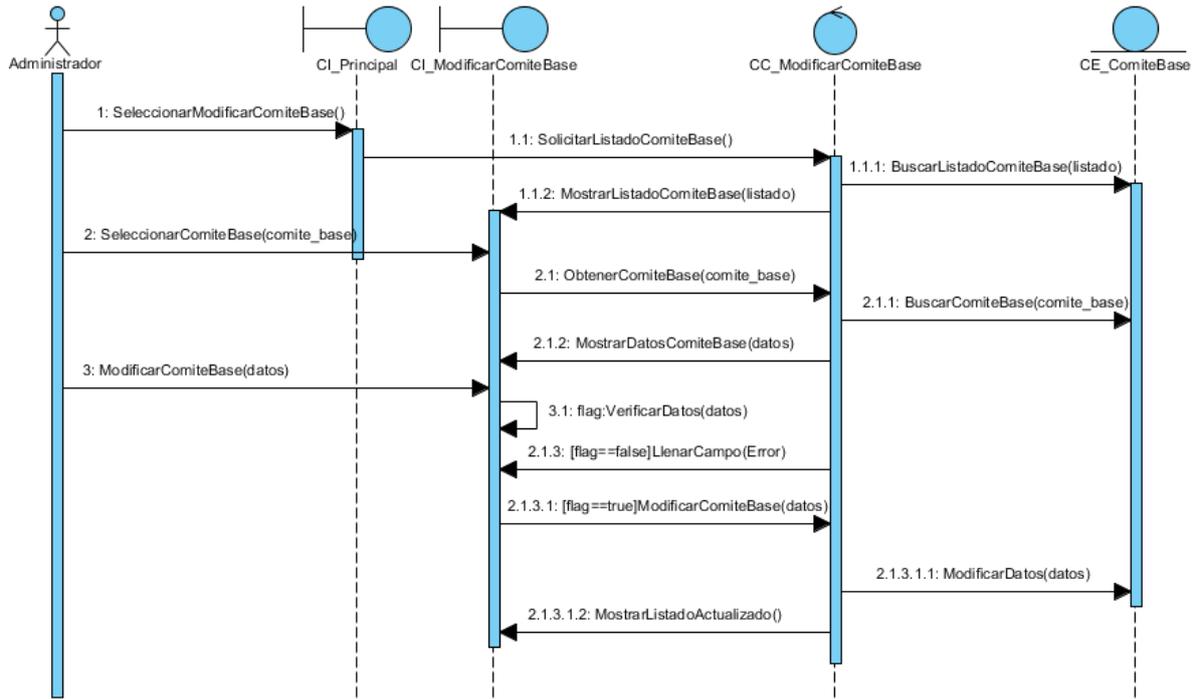


Figura 16 Diagrama de secuencia. RF Modificar Comité de base.

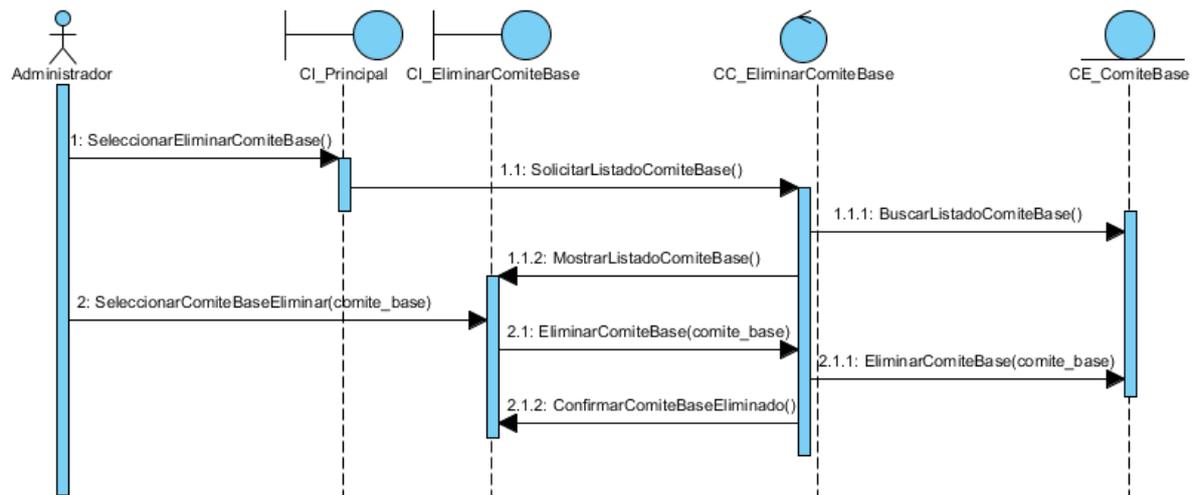


Figura 17 Diagrama de secuencia. RF Eliminar Comité de base.

Anexos

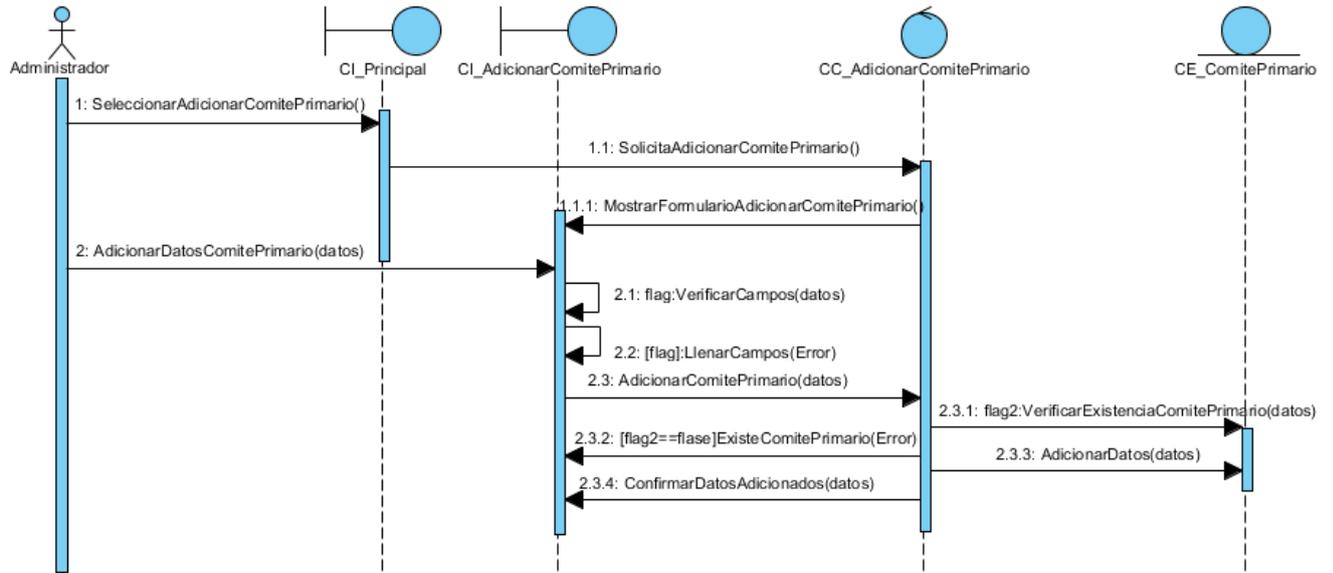


Figura 18 Diagrama de secuencia. RF Adicionar Comité primario.

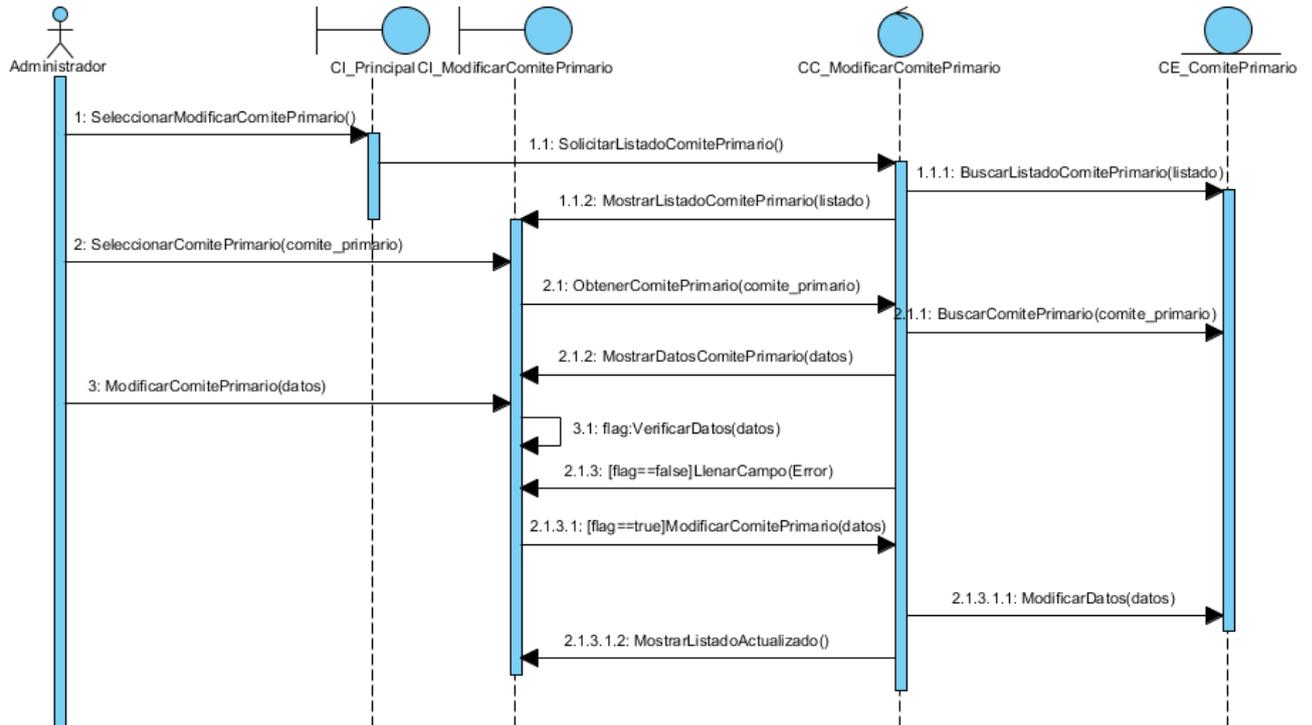


Figura 19 Diagrama de secuencia. RF Modificar Comité primario.

Anexos

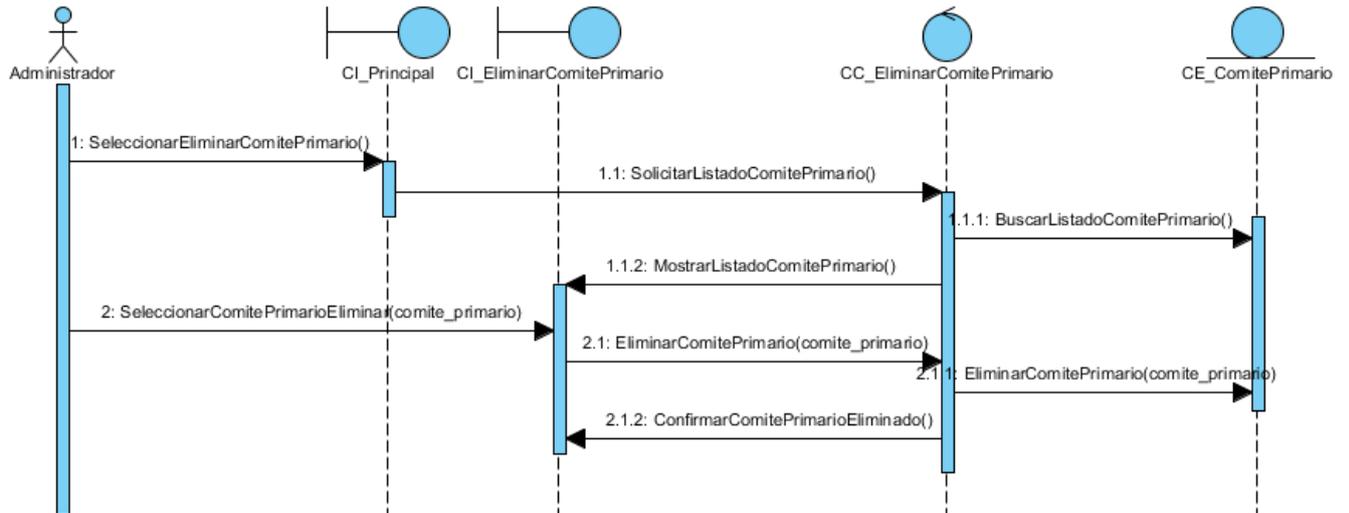


Figura 20 Diagrama de secuencia. RF Eliminar Comité primario.

Anexo 6: Descripción del modelo entidad-relación.

Nombre de la tabla: node

Descripción: la tabla almacena los datos generales de todos los nodos.

Tabla 11 Descripción de la tabla node.

Atributo	Tipo	Descripción
nid	int	Identificador primario para un nodo.
vid	int	Indica la última versión del nodo.
type	varchar	Tipo de nodo.
language	varchar	Lenguaje del nodo.
title	varchar	Título del nodo.
uid	int	Identificador de la tabla users que indica el usuario que creó el nodo
status	int	Estado de publicación del nodo. Cero para los nodos “No publicados” y uno para los “Publicados”.
created	int	Fecha de creación del nodo.
changed	int	Fecha de la última modificación hecha al nodo.

<i>comment</i>	int	Especifica las opciones de comentario. Cero para “Deshabilitado”, uno para “Solo Lectura” y dos para “Lectura/Escritura”.
<i>promote</i>	int	Indica el grado de promoción del nodo, para que aparezca o no en la página principal.
<i>sticky</i>	int	Indica la prioridad del nodo sobre otros nodos promocionados.
<i>tnid</i>	int	La identificación del grupo de traducción de este nodo.
<i>translate</i>	int	Indica si esta página de traducción necesita ser actualizada.

Nombre de la tabla: node_type

Descripción: la tabla almacena los datos generales de todos los tipos de contenidos.

Tabla 12 Descripción de la tabla node_type.

Atributo	Tipo	Descripción
<i>type</i>	varchar	El tipo de contenido del nodo e identificador primario de la tabla.
<i>name</i>	varchar	Nombre del tipo de contenido.
<i>base</i>	varchar	Nombre de sistema del tipo de contenido.
<i>module</i>	varchar	Módulo que genera el tipo de contenido que se muestra en el nodo.
<i>description</i>	text	Descripción del tipo de contenido mostrado en el nodo.
<i>help</i>	text	Texto de ayuda que se muestra al usuario al crear un nodo de este tipo de contenido.

<i>has_title</i>	int	Si el tipo de contenido tiene un título. Cero si no tiene y uno si tiene.
<i>title_label</i>	varchar	El texto que aparece como título del nodo
<i>custom</i>	int	Si este contenido ha sido generado por un módulo (cero) o añadido por un usuario (uno).
<i>modified</i>	int	Si se ha modificado el contenido del nodo por un administrador. Cero si no se ha modificado, uno si se ha modificado
<i>locked</i>	int	Si el nombre de sistema del nodo puede ser cambiado por un administrador. Cero si se puede, uno si no se puede
<i>disabled</i>	int	Si el nodo se encuentra activo. Cero si no se encuentra activo, uno si está activo
<i>orig_type</i>	varchar	El nombre del sistema original de este tipo de contenido. Puede ser diferente al actual si el valor del atributo " <i>locked</i> " es cero

Nombre de la tabla: users

Descripción: la tabla almacena los datos generales de todos los usuarios.

Tabla 13 Descripción de la tabla users.

Atributo	Tipo	Descripción
uid	int	El identificador único de cada usuario y llave primaria de la tabla.
name	varchar	Nombre de usuario único
pass	varchar	Contraseña del usuario.
mail	varchar	Dirección de correo electrónico del usuario.

Anexos

theme	varchar	Descripción del tipo de contenido mostrado en el nodo.
signature	varchar	Firma del usuario.
signature_format	varchar	El formato de texto de la firma.
create	int	Fecha de creación del usuario.
access	int	Último acceso del usuario al sitio.
login	int	Última vez que el usuario se autenticó en el sitio.
status	int	Si el usuario está activo (1) o bloqueado (0).
time_zone	varchar	Zona horaria del usuario.
language	varchar	El lenguaje del usuario.
picture	int	Identificador asociado a la imagen del usuario
init	varchar	Dirección de correo electrónico del usuario usada al crear la cuenta.
data	bytea	Datos entrados por el usuario durante la edición de su cuenta. Usados solamente por el sistema como una forma de registro de información.

Nombre de la tabla: role

Descripción: la tabla almacena los datos generales de todos los roles.

Tabla 14 Descripción de la tabla role.

Atributo	Tipo	Descripción
rid	int	El identificador único de cada rol y llave primaria de la tabla.
name	varchar	Nombre único del rol.
weight	int	El peso de este rol en las listas y la interfaz de usuario.

Anexos

module	varchar	Módulo que genera el tipo de contenido que se muestra en el nodo.
machine_name	varchar	El nombre de sistema asignado por el usuario durante la creación del rol.

Nombre de la tabla: field_data_field_comit_primario

Descripción: la tabla almacena los datos generales del campo “comité primario”.

Tabla 15 Descripción de la tabla field_data_field_comit_primario.

Atributo	Tipo	Descripción
entity_type	varchar	El tipo de contenido al que este dato está adherido y llave primaria de la tabla.
bundle	varchar	Tipo de contenido al cual pertenece.
deleted	int	Si el campo está eliminado temporalmente.
entity_id	int	Id del nodo al que pertenece.
revisión_id	int	Id de la revisión del nodo al que pertenece.
language	varchar	Idioma del nodo al que pertenece.
delta	int	Posición en la lista de id de cada nodo, usado para campos multivaluados.
field_comit_primario_nid	int	El id del contenido al cual el fichero está asociado

Nombre de la tabla: field_data_field_a_o

Descripción: la tabla almacena los datos generales del campo “año”.

Tabla 16 Descripción de la tabla field_data_field_a_o.

Atributo	Tipo	Descripción
----------	------	-------------

Anexos

<i>entity_type</i>	varchar	El tipo de contenido al que este dato está adherido y llave primaria de la tabla.
<i>bundle</i>	varchar	Tipo de contenido al cual pertenece.
<i>deleted</i>	int	Si el campo está eliminado temporalmente.
<i>entity_id</i>	int	Id del nodo al que pertenece.
<i>revision_id</i>	int	Id de la revisión del nodo al que pertenece.
<i>language</i>	varchar	Idioma del nodo al que pertenece.
<i>delta</i>	int	Posición en la lista de id de cada nodo, usado para campos multivaluados.
<i>field_a_o_tid</i>	int	El id del contenido al cual el fichero está asociado

Nombre de la tabla: field_data_field_tipo

Descripción: la tabla almacena los datos generales del campo “tipo”.

Tabla 17 Descripción de la tabla field_data_tipo.

Atributo	Tipo	Descripción
<i>entity_type</i>	varchar	El tipo de contenido al que este dato está adherido y llave primaria de la tabla.
<i>bundle</i>	varchar	Tipo de contenido al cual pertenece.
<i>deleted</i>	int	Si el campo está eliminado temporalmente.
<i>entity_id</i>	int	Id del nodo al que pertenece.
<i>revision_id</i>	int	Id de la revisión del nodo al que pertenece.
<i>language</i>	varchar	Idioma del nodo al que pertenece.
<i>delta</i>	int	Posición en la lista de id de cada nodo, usado para campos multivaluados.

field_tipo_tid	int	El id del contenido al cual el fichero está asociado
-----------------------	-----	--

Anexo 8: Lista de Chequeo para Pruebas de Seguridad.

Peso: Define si el indicador a evaluar es crítico o no.

Evaluación (Eval): es la forma de evaluar el indicador en cuestión. El mismo se evalúa de 1 en caso de mal y 0 en caso que el elemento revisado no presente errores.

Cantidad de elementos afectados: especifica la cantidad de errores encontrados sobre el mismo indicador.

Comentario: especifica los señalamientos o sugerencias que quiera incluir la persona que aplica la lista de chequeo.

N.P. (No Procede): se usa para especificar que el indicador a evaluar no se puede aplicar en ese caso.

Tabla 18 Pruebas de Autorización.

Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	Puede un usuario estándar (no administrador) modificar sus privilegios en la aplicación.	0			
Crítico	Puede un usuario estándar (no administrador) modificar los privilegios de otro usuario.	0			

Tabla 19 Pruebas de Gestión de Sesiones.

--

Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	Al copiar la URL de la aplicación después de estar autenticado, cerrar el navegador y volver a abrirlo para pegar la dirección copiada anteriormente, la aplicación permite que el usuario entre a la aplicación.	0			
Crítico	Al cerrar la sesión de un usuario y dar clic en el botón del navegador "Atrás" la aplicación vuelve a entrar a la sesión autenticada.	0			

Tabla 20 Comprobación del Sistema de Autenticación.

Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	Se bloquea la sesión del usuario después de un tiempo X (establecido por la aplicación) estando sin usar la misma.	0	NP		
Crítico	Se bloquea la cuenta del usuario después de un número X (establecidos por la aplicación) de intento de login fallidos por el usuario.	0	NP		3 intentos

	De ser así definir la cantidad de intentos en la columna Comentarios.				
Crítico	Se muestran diferentes mensajes de error al usuario “intentar autenticarse con un usuario correcto y una contraseña incorrecta” y “al autenticarse con un usuario incorrecto y la misma contraseña del intento anterior”.	1			
Crítico	El mensaje devuelto al intentar autenticarse con un usuario correcto y contraseña incorrecta, es el mismo al intento con usuario y contraseña incorrecta, si es el mismo intente verificar el tiempo de respuesta, pues podría apreciarse una media ligeramente diferente en el tiempo de respuesta, que igualmente puede usarse esta información para descubrir cuentas válidas.	0			
Crítico	El campo usuario de la autenticación al sistema tiene el autocompletamiento activado (guarda los usuarios que se autentican). Para ver esto se debe	0			

Anexos

	autenticar al sistema, luego salir y poner la primera letra del usuario autenticado para ver si muestra el nombre de usuario completo.				
Crítico	El sistema protege el envío de los datos mediante protocolo seguro (https).	0			
Crítico	El sistema usa algún certificado.	0			

Tabla 21 Validación de Datos.

Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	Se enmascaran datos confiables cuando se visualicen en la aplicación (Por ejemplo: Contraseñas)	0			
Crítico	La aplicación solamente permite contraseñas alfanuméricas, que incluyan caracteres especiales y que tengan seis caracteres mínimos de longitud.	0			
Crítico	La aplicación permitir la funcionalidad de cambio de contraseña únicamente a usuarios autenticados validando la antigua	0			

Anexos

	contraseña, la nueva contraseña y la respuesta a la pregunta de seguridad (opcional en dependencia del proyecto).				
Crítico	El sistema muestra algún mensaje indebido, al colocar en la barra de dirección o en campos de entrada los caracteres: comillas simples (``), signos de ampersand (&), signos: + - /.	0			

Anexo 9: Resumen generado por *Web Security*.

Vanilla SQL Injection

SQL Injection (SQLI) is a code injection technique that exploits a security vulnerability occurring in the database layer of a web application. The vulnerability is present when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and thereby unexpectedly executed.

solution: *Sanitize all user-supplied data before using it as part of database queries.*

database: *DB2*

request:

GET http://10.53.12.244/Paprubas/drupal.uci.cu/taxonomy/term/33. HTTP/1.1

Vanilla SQL Injection

SQL Injection (SQLI) is a code injection technique that exploits a security vulnerability occurring in the database layer of a web application. The vulnerability is present when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and thereby unexpectedly executed.

solution: Sanitize all user-supplied data before using it as part of database queries.

database: DB2

request:

```
GET http://10.53.12.244/Papuebas/drupal.uci.cu/taxonomy/term/21. HTTP/1.1
```

Vanilla SQL Injection

SQL Injection (SQLI) is a code injection technique that exploits a security vulnerability occurring in the database layer of a web application. The vulnerability is present when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and thereby unexpectedly executed.

solution: Sanitize all user-supplied data before using it as part of database queries.

database: DB2

request:

```
GET http://10.53.12.244/Papuebas/drupal.uci.cu/comment/2. HTTP/1.1
```

Vanilla SQL Injection

SQL Injection (SQLI) is a code injection technique that exploits a security vulnerability occurring in the database layer of a web application. The vulnerability is present when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and thereby unexpectedly executed.

solution: Sanitize all user-supplied data before using it as part of database queries.

database: DB2

request:

```
GET http://10.53.12.244/Papuebas/drupal.uci.cu/node/5. HTTP/1.1
```

Vanilla SQL Injection

SQL Injection (SQLI) is a code injection technique that exploits a security vulnerability occurring in the database layer of a web application. The vulnerability is present when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and thereby unexpectedly executed.

solution: Sanitize all user-supplied data before using it as part of database queries.

database: DB2

request:

```
GET http://10.53.12.244/Paprubas/drupal.uci.cu/taxonomy/term/8. HTTP/1.1
```

Vanilla SQL Injection

SQL Injection (SQLI) is a code injection technique that exploits a security vulnerability occurring in the database layer of a web application. The vulnerability is present when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and thereby unexpectedly executed.

solution: Sanitize all user-supplied data before using it as part of database queries.

database: DB2

request:

```
GET http://10.53.12.244/Paprubas/drupal.uci.cu/forum/1. HTTP/1.1
```

Autocomplete Enabled

Autocomplete should always be disabled (`autocomplete="off"`), especially in forms which process sensitive data, such as forms with password fields, since an attacker, if able to access the browser cache, could easily obtain the cached information in cleartext.

solution: Disable the autocomplete feature (`autocomplete="off"`) on forms which may hold sensitive data.

url: `http://10.53.12.244/Paprubas/drupal.uci.cu/`

form:

```
<form action="/Paprubas/drupal.uci.cu/node?destination=node" method="post" id="user-login-form" accept-charset="UTF-8">...
```

HTTP Banner Disclosure

The application discloses its type and version. This information can be used by attackers to make an educated guess about the application environment and any inherited weaknesses that may come with it.

solution: *It is recommended to prevent the application from disclosing its type and version.*

banner: `Server: Apache/2.4.6 (Ubuntu)`

request:

```
GET http://10.53.12.244/Papuebas/drupal.uci.cu/ HTTP/1.1
```

HTTP Banner Disclosure

The application discloses its type and version. This information can be used by attackers to make an educated guess about the application environment and any inherited weaknesses that may come with it.

solution: *It is recommended to prevent the application from disclosing its type and version.*

banner: `X-Powered-By: PHP/5.5.3-1ubuntu2.2`

request:

```
GET http://10.53.12.244/Papuebas/drupal.uci.cu/ HTTP/1.1
```