



Universidad de las Ciencias Informáticas

Facultad 4

MÓDULO PARA LA GESTIÓN DE LA INFORMACIÓN DEL PROCESO
DIAGNÓSTICO SOCIOPOLÍTICO DE LA UNIÓN DE JÓVENES COMUNISTAS
EN LA UCI.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS
INFORMÁTICAS

Autores: Raydel Herrera García

Oriesniel Martínez Duarte

Tutores: Ing. Leonardo Rodríguez

Ing. Carlos Montenegro Amador

La Habana, junio de 2016

Declaración de Autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 4 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 _____.

Autores:

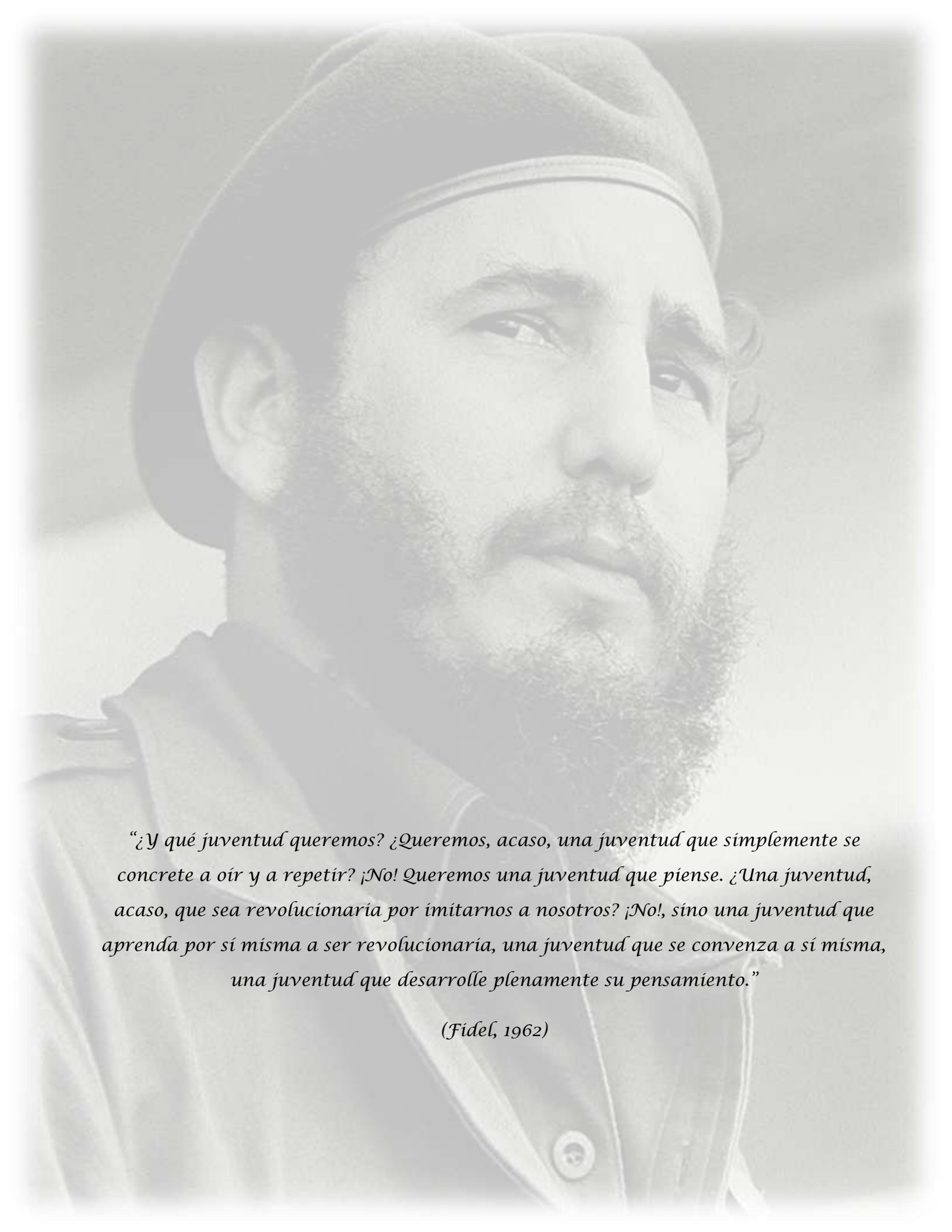
Raydel Herrera García

Oriesniel Martínez Duarte

Tutores:

Ing. Leonardo Rodríguez González

Ing. Carlos Montenegro Amador



“¿Y qué juventud queremos? ¿Queremos, acaso, una juventud que simplemente se concrete a oír y a repetir? ¡No! Queremos una juventud que piense. ¿Una juventud, acaso, que sea revolucionaria por imitarnos a nosotros? ¡No!, sino una juventud que aprenda por sí misma a ser revolucionaria, una juventud que se convenza a sí misma, una juventud que desarrolle plenamente su pensamiento.”

(Fidel, 1962)

Dedicatoria

De Raydel:

Muchas veces no nos damos cuenta de cuan importantes son esas personas que están junto a nosotros cada día para brindarnos su apoyo incondicional, por eso vale la pena recordarles en cada momento cuanto se les quiere y se les valora por reír y llorar junto a nosotros.

A mí mamá: por ser madre para guiarme y enseñarme el camino y padre para aconsejarme.

A mí hermana: por ser el regalo más grande que me ha dado la vida.

A mí novia: por tanto amor, confianza y apoyo incondicional.

A mis abuelos: porque el amor es palabra pequeña para definir el lazo que nos une.

A mis suegros: desde que llegaron a mi vida han sido como mis padres.

A mi cuñado: por ser el hermano varón que nunca tuve.

De Oríesniel:

A mi madre: por ser la fuente de inspiración que día tras día me da las fuerzas necesarias para cumplir mis propósitos en la vida.

A mi abuelita míma: por ser la personita con el alma más pura de todo el planeta y quererme como soy.

A mi tía Omarita: por todo el amor y el apoyo incondicional que me ha brindado.

A mi hermana: porque más que hermana es mi amiga, mi consejera mi todo.

A mi primíta Mary: por ser el regalito más tierno que me ha dado la vida.

A mi cuñado: por ser amigo, hermano y demostrarme que puedo contar siempre contigo.

Agradecimientos:

De Raydel:

A mis tutores Carlos y Leo sin ellos la realización de esta obra no hubiera sido posible, por todo el apoyo que recibí de su parte, por la paciencia que tuvieron y por estar ahí para nosotros en cada momento. ¡Gracias!

A mi mamá, que una tarde de junio me trajo a la vida, y de ahí en adelante me ha dedicado su vida, me educó y me enseñó el camino para convertirme en la persona que soy hoy en día, gracias también por enseñarme a no conformarme e ir a por más, no me alcanza una vida para retribuir ese amor que me has dado.

A mi novia Mairélis, por entregarme ese amor que pensé que no existía hoy en día, por atenuar muchos de mis defectos, por la confianza, por ese apoyo que siempre fue mi sostén cuando lo necesitaba, es una bendición que entre once millones de personas me halla enamorado de ella.

A mis suegros Rey y Maira, por ese cariño desinteresado que me han brindado por acogerme como un hijo, por preocuparse tanto por mí, por darle vida a la personita que más quiero en el mundo, por darme a ese hermano varón que no tuve.

A mi compañero de tesis Oriensiel, siempre ha estado ahí para mí, dentro de las mejores cosas que me sucedieron en la universidad una de ellas sin duda alguna es haber contado con su amistad, por aguantar mis regaños, por apoyarme tanto cuando lo necesitaba, el lazo que nos une solo lleva un nombre hermandad.

A mis colegas David y Lester, siempre estaré agradecido a la vida por dejarme contar con su amistad, por hacer que estos lazos de hermandad se fortalezcan cada día más, gracias por estar ahí mis hermanos.

¡Gracias a todos!

De Oríesniel:

A mi madre por darme la oportunidad de abrir los ojos un día de julio, y criarme con tanto sacrificio. Por hacer función de madre y padre al mismo tiempo, guiarme siempre por el buen camino y dedicar su vida para verme convertido en un hombre de bien. Por brindarme tanto amor a pesar de nuestras diferencias y por todo el sacrificio que ha hecho para ayudarme a cumplir mis metas.

A mi abuelita, mi tía y mi prima por demostrarme el verdadero significado de la palabra familia y por hacer de tripas corazón para siempre verme contento. Por estar siempre disponibles para mí, sea cual sea la hora y el momento. Por todo el apoyo incondicional y por todas las horas dedicadas a la causa.

A mi compañero de tesis Raydel porque a pesar de nuestras diferencias de carácter siempre has estado disponible para cuando lo he necesitado, por saber sacarme más de sonrisa cuando siempre lo necesité, por ser ante todo hombre y amigo, por acompañarme tanto en los momentos malos como en los buenos y por entrar en ese pequeño círculo de mejores momentos por los que he pasado en mi vida universitaria.

Un especial agradecimiento a la primera persona que conocí en la UCI y rápidamente se convirtió en parte de mi vida: David Nazario. Hoy más que mi amigo, más que mi hermano eres mi sangre, creo que no existen palabras para describir el lazo de hermandad que nos une. Gracias por ser tan humilde, tan sincero, tan sano y por sobre todo siempre comprenderme y apoyarme incondicionalmente.

A mis tutores Carlos y Leo por todo el apoyo y el tiempo brindado incondicionalmente.

¡Gracias a todos!

Resumen

La gestión de la información es un elemento de vital importancia para la toma de decisiones para cualquier organización. En la Universidad de las Ciencias Informáticas existen varias aplicaciones para informatizar el procesamiento de la información, dentro de ellas se encuentran las dedicadas a controlar algunos de los procesos sustantivos de la Unión de Jóvenes Comunistas. En la actualidad esta organización en la Universidad gestiona la información relacionada con el diagnóstico sociopolítico, los expedientes de las Brigadas Técnicas Juveniles para optar por el sello Forjadores del Futuro, la gestión del plan de trabajo, el aval de promoción al Partido Comunista de Cuba y la evaluación de los militantes de forma manual, provocándole problemas de almacenamiento, duplicación y pérdida de la misma debido al deterioro de los documentos.

En la presente investigación se desarrolla un módulo que se integrará al **Sistema Integral de Gestión de los Procesos Sustantivos de la UJC en la UCI (SIGPS_UCI)**, con el objetivo de agilizar la gestión de la información que se genera en el proceso del diagnóstico sociopolítico de la Unión de Jóvenes Comunistas en la UCI. Para guiar el desarrollo del sistema se utilizó como metodología de desarrollo AUP-UCI, tomando como escenario la descripción de casos de uso con el objetivo de mantener una línea base con respecto a la investigación realizada en el desarrollo del SIGPS_UCI. La implementación de la solución estuvo apoyada en el framework de desarrollo Symfony2, haciendo uso del servidor web Apache v2.6 y del lenguaje de programación PHP v5.4.12. En el modelado de los artefactos que genera el uso de la metodología se utilizó el lenguaje de modelado UML y la herramienta Visual Paradigm v8.0.

Palabras clave:

Aval de promoción al PCC, Diagnóstico sociopolítico, Evaluación, Expediente de las BTJ, Militantes, Plan de trabajo, UJC

Índice

Introducción	1
1 Capítulo 1: Fundamentación Teórica	7
1.1 Introducción	7
1.2 Sistema de información	7
1.3 Software de gestión	7
1.4 Análisis de sistemas similares	9
1.5 Lenguajes de Programación	10
1.5.1 Lenguajes de programación del lado del cliente	10
HTML v5	11
JavaScript	11
jQuery 1.10.2	11
CSS v3	12
1.5.2 Lenguajes de programación del lado del servidor	12
PHP v5.4	12
1.6 Marcos de trabajo	13
Symfony v2.5	13
Bootstrap v2.0.3	13
Doctrine	14
1.7 Servidor Web	14
Apache v2.2.6	15
1.8 Entorno de desarrollo	15
NetBeans IDE v8.0.2	15
1.9 Sistema gestor de base de datos (SGBD)	16
PostgreSQL	16
1.10 Lenguaje de modelado	16
Lenguaje de Modelado Unificado	17

1.11	Metodología de desarrollo de software	17
1.11.1	Metodologías Tradicionales	18
1.11.2	Metodologías Ágiles	18
	Proceso Unificado Ágil (AUP)	19
	SCRUM	20
	Fundamentación de la metodología a utilizar	20
1.12	Herramienta CASE	21
	Visual Paradigm v8.0	21
	Conclusiones Parciales	22
2	Capítulo 2: Propuesta de solución	23
2.1	Introducción	23
2.2	Modelo de dominio	23
2.2.1	Conceptos del dominio	23
2.2.2	Diagrama del modelo de dominio	24
2.3	Descripción del sistema propuesto	25
2.4	Requerimientos funcionales	26
2.5	Requerimientos no funcionales	30
2.6	Modelo de casos de uso del sistema	32
2.6.1	Descripción de los actores del sistema	32
2.6.2	Diagrama de actores del Sistema	33
2.6.3	Diagrama de casos de uso del sistema	33
2.6.4	Patrones de casos de uso del sistema	34
2.6.5	Descripción textual de los casos de uso del sistema	35
2.7	Modelo de análisis	40
2.7.1	Diagrama de clases del análisis	40
2.7.2	Diagrama de colaboración del análisis	41
	Conclusiones Parciales	42
3	Capítulo 3: Desarrollo de la propuesta de solución	43
3.1	Introducción	43

3.2	Patrones de diseño	43
3.3	Patrón arquitectónico Modelo – Vista – Controlador en Symfony.....	43
3.4	Patrones GOF.....	44
3.5	Patrones GRASP.....	45
3.6	Modelado de diseño	46
3.6.1	Diagrama de clases del diseño	47
3.6.2	Diagrama de secuencia	48
3.6.3	Diagrama de despliegue	49
3.6.4	Diagrama de componentes.....	50
3.7	Pruebas de software	51
3.7.1	Niveles de prueba.....	51
3.7.2	Métodos de prueba.....	53
3.7.3	Diseño de casos de prueba	53
3.7.4	Resultados obtenidos de las pruebas de caja negra.....	57
3.7.5	Resultados de las pruebas de integración	58
3.8	Validación de los resultados	59
	Conclusiones Parciales.....	60
	Conclusiones generales.....	61
	Recomendaciones	62
	Referencias Bibliográficas.....	63
	Bibliografía.....	69

Introducción

En los últimos años la ciencia ha avanzado considerablemente convirtiéndose en parte de la cultura del hombre. En este proceso de avance juega un papel importante el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC). Cuba es un país, que a pesar de las limitaciones económicas que presenta debido a las consecuencias del bloqueo, no ha quedado exento de la utilización de las mismas. Desde hace un tiempo el gobierno cubano se ha enfocado en la informatización de los distintos sectores de la sociedad, respaldado mayormente por las investigaciones, aportes y proyecciones hechas por entidades o grupos de desarrollo de software.

Una de las organizaciones que forma parte de los diferentes sectores de la sociedad cubana es la Unión de Jóvenes Comunistas (UJC), definida como la organización de avanzada de la juventud cubana, cantera y reserva combativa del Partido Comunista de Cuba (PCC). La UJC tiene la responsabilidad de formar en sus filas a la vanguardia de la juventud con el objetivo de defender el socialismo como única opción de desarrollo. Tiene la misión de participar activamente en el desarrollo económico y social del país, convencida de la importancia de la batalla económica en la sostenibilidad y preservación de nuestro sistema socialista.(2)

En la Universidad de las Ciencias Informáticas (UCI) como estructura de la UJC existe un Comité de la UJC (C_UJC) al cual se subordinan varios Comités Primarios (CP) como estructuras intermedias entre el Comité UJC y los Comités de Base (CB).

En un año tipo, los comités primarios desarrollan procesos estándares tales como: evaluación de los militantes, proceso de crecimiento, otorgamiento de estímulos, sanciones, procesos de traslado, altas y bajas a militantes. Estos procesos son dirigidos por diferentes frentes o esferas, pero tienen como objetivo garantizar el correcto funcionamiento de la organización.

Hoy día existe una aplicación nombrada: “Sistema integral para la gestión de los procesos sustantivos de la UJC en la UCI (SIGPS_UCI)”, la cual fue desarrollada para facilitar el desarrollo de varios procesos de la organización. Este sistema gestiona procesos como el cierre de funcionamiento (resumen estadístico de las reuniones de CB) y el ID2 (proceso que rige los pasos por lo que ha transitado un militante, lleva el control de las altas y bajas producidas en los CB).

Dicha aplicación sirve de apoyo al funcionamiento de la organización, pero en la misma aún no se gestionan procesos que garantizan el correcto funcionamiento de las organizaciones de base (OB) en la UJC. Ejemplo de estos procesos son el diagnóstico sociopolítico, la gestión del plan de trabajo, la evaluación de los militantes, el otorgamiento y registro de avales para el crecimiento al PCC. Estos procesos generan una carga de trabajo y mucha información debido a que estos procesos tabulan un buen número de indicadores y el desarrollo de los mismos se realiza de forma manual.

El diagnóstico sociopolítico se realiza anualmente, recoge la valoración del trabajo realizado por la organización en función del cumplimiento de su misión y del objeto social del centro o área. A través de este diagnóstico se identifican las fortalezas y debilidades que posee la organización en el área en que actúan los militantes de los CB. Caracteriza el censo del universo juvenil, o sea, todos los no militantes, y define las acciones generales para dar solución a las problemáticas identificadas, que se concretan en tareas específicas del plan de trabajo.(3)

El diagnóstico contiene tres componentes fundamentales:

1. Caracterización del centro o área de atención: en ella se exponen los resultados en el cumplimiento de su objeto social y el papel que desempeñan los jóvenes en estos. Este componente recoge los factores que dificultan o facilitan la consecución de tareas y actividades en una organización de base, recoge los criterios expresados por las representaciones de las organizaciones políticas y de masas que se encuentren en el centro acerca del área de atención para así desarrollar una buena caracterización de la misma. En este proceso deben quedar plasmados todos los criterios del universo juvenil, el resumen del análisis interno y externo, para así aprovechar las potencialidades y mitigar los riesgos que inciden directamente en el área.(3)

2. Caracterización de la UJC: en ella se realiza una valoración de los resultados del trabajo de las organizaciones de base en períodos anteriores, definiendo las debilidades y las fortalezas presentes. Esta valoración incluirá un resumen de las principales actividades desarrolladas y el estado de cumplimiento de las mismas en las organizaciones de base, además de un resumen con todo el resultado de los procesos políticos:

- ✓ Crecimiento de la UJC

- ✓ Promociones al PCC
- ✓ Evaluaciones
- ✓ Condecoraciones

Como parte de la caracterización de la UJC se realiza en cada organización de base un análisis para conocer la situación real de la misma, haciendo uso de la herramienta estratégica DAFO (Debilidades, Amenazas, Fortalezas, Oportunidades). El objetivo de la utilización de esta herramienta consiste en concretar, en un gráfico o una tabla resumen, la evaluación de los puntos fuertes y débiles de la organización y oportunidades y amenazas externas, en coherencia con la lógica de que la estrategia debe lograr un adecuado ajuste entre sus capacidades internas y su posición externa.(3)

Tabla 1. Matriz DAFO (Elaboración propia)

Matriz DAFO	Oportunidades	Amenazas
Fortalezas	Estrategias ofensivas	Estrategias defensivas
Debilidades	Estrategias de reorientación	Estrategias de supervivencias

3. Caracterización de los jóvenes, la misma se realiza en dos momentos:

- I. Valoración colectiva de los jóvenes hasta 35 años de edad (militantes del comité de base), definiendo sus necesidades, intereses, preocupaciones y expectativas.
- II. Caracterización individual de cada joven no militante (menor de 30 años) definiendo los que muestran potencialidades para el proceso de crecimiento y los más destacados.(3)

En este subproceso se recoge la caracterización individual de todos los jóvenes que pertenezcan a una organización de base donde se expresan las necesidades, intereses, preocupaciones individuales y/o expectativas de los jóvenes como parte de una completa valoración de los mismos.

El resultado del diagnóstico sociopolítico quedará plasmado en el plan de trabajo que es la herramienta

fundamental con que cuenta el CB para planificar sus actividades. Este plan tiene un carácter trimestral y se garantiza su aprobación en los meses de diciembre, marzo, junio y septiembre antecediendo al trimestre que se va a planificar.

En la UCI el diagnóstico sociopolítico es realizado por una comisión de trabajo integrada por tres militantes. Actualmente toda la información que genera este proceso se gestiona de forma manual, haciéndose lenta la recopilación de información proveniente de diferentes fuentes. Toda la información se almacena haciendo uso de herramientas ofimáticas, que son utilizadas para la elaboración de los documentos, lo que provoca una aglomeración innecesaria de archivos en formato digital. Esta manera de proceder provoca demora a la hora de hacer simples consultas o modificar la información, y en muchos casos, esta aparece en más de un documento o no aparece. No existe una centralización de los documentos, lo que dificulta la disponibilidad y el acceso a estos desde cualquier parte debido a que el traslado se realiza de un lugar a otro en el mejor de los casos vía correo electrónico.

La pérdida de la información relevante que genera el diagnóstico dificulta la realización del análisis interno y externo de la organización, que no sean tomados en cuenta los criterios de los jóvenes y hace difícil aprovechar las potencialidades y mitigar los riesgos que inciden en el área. Los resultados del proceso en años anteriores en ocasiones se pierden e impiden realizar una comparación de los resultados del trabajo de las organizaciones y se dificulta la realización de la matriz DAFO para trazar una estrategia que logre un adecuado ajuste entre las capacidades internas y externas de la organización.

A partir de la situación problemática descrita anteriormente se propone como **problema a resolver**: ¿Cómo agilizar la gestión de la información que se genera en el proceso del diagnóstico sociopolítico en la Unión de Jóvenes Comunistas en la UCI?

Con el fin de solucionar el problema, se define como **objeto de estudio** de la presente investigación: la gestión de la información de los procesos sustantivos de la UJC. Enfocando el **campo de acción** a la gestión de la información del proceso del diagnóstico sociopolítico de la UJC en la UCI.

Para darle solución al problema planteado se define como **objetivo general**: Desarrollar un módulo para agilizar la gestión de la información que se genera en el proceso del diagnóstico sociopolítico de la Unión de Jóvenes Comunistas en la UCI.

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

1. Obtener los fundamentos teóricos asociados a la gestión de la información que permitan sustentar una posible propuesta de solución.
2. Diseñar una propuesta de solución que dé respuesta al problema planteado.
3. Implementar la solución diseñada.
4. Realizar pruebas que validen el correcto funcionamiento del sistema implementado.

Para guiar la investigación se define como **Hipótesis** el desarrollo de un módulo para la gestión de la información que genera el proceso del diagnóstico sociopolítico, permitirá agilizar la obtención de los indicadores que genera dicho proceso.

Para dar apoyo se utilizaron los siguientes **métodos investigativos**:

Métodos Teóricos:

- ✓ **Analítico-Sintético:** mediante este método se analizan los documentos y las teorías referentes a la gestión del diagnóstico sociopolítico y de los procesos existentes relacionados con este, logrando así una descomposición del proceso en cuestión, proporcionando la extracción de los datos más importantes.
- ✓ **Hipotético-deductivo:** se usó para la elaboración de la hipótesis de la investigación y en la obtención de nuevas líneas de trabajo a partir de los resultados obtenidos.
- ✓ **Análisis documental:** se utilizó para la consulta de la bibliografía especializada en los tópicos correspondientes a la investigación.

Métodos empíricos:

- ✓ **Entrevista:** se utilizó para la recopilación de información acerca de la estructura y funcionamiento de la organización.
- ✓ **Observación:** facilitó el estudio del proceso del diagnóstico sociopolítico de la UJC en la UCI y los principales problemas asociados.

Estructura Capitular

Capítulo 1: Fundamentación teórica

Se exponen los elementos teóricos que sustentan el problema científico y los objetivos del trabajo. Se realiza un análisis de las metodologías que se pueden utilizar, además se justifica la selección de cada una de las herramientas de desarrollo, lenguaje de programación a utilizar para la solución así como el framework de desarrollo.

Capítulo 2: Propuesta de solución

En este capítulo se exponen los elementos que permiten describir la propuesta de solución, tales como: modelo de dominio, requerimientos funcionales y no funcionales, diagramas de casos de uso y la descripción textual de cada uno de ellos, incluyendo los prototipos de interfaz de usuario para lograr un entendimiento claro del sistema a desarrollar.

Capítulo 3: Desarrollo de la propuesta de solución

Este capítulo abarca todo lo relacionado con la implementación del módulo, reflejando prácticas de programación y los estándares de codificación. Además, describe la estrategia trazada para desarrollar las pruebas al sistema.

Capítulo 1: Fundamentación Teórica

1 Capítulo 1: Fundamentación Teórica

1.1 Introducción

En este capítulo se exponen los elementos teóricos y conceptuales asociados al objeto de estudio, campo de acción y problema a resolver. Se realiza un estudio de los sistemas desarrollados anteriormente con fines de gestionar información acerca de las organizaciones, enfocado específicamente en la UJC a través del estudio del estado del arte para analizar sus características, funcionalidades y utilidades actuales.

Teniendo en cuenta que esta propuesta de solución es un sistema heredado del sistema SIGPS_UCI, se describen las diferentes herramientas de software, metodologías y lenguaje de programación que se emplearán en el proceso de diseño y desarrollo de la solución planteada.

1.2 Sistema de información

Los Sistemas de Información (SI) ofertan, regulan y gestionan todo tipo de recursos de información. Son los encargados de los procesos de almacenamiento, identificación, transformación, organización, tratamiento y recuperación de la información. En estos pasos o fases interviene la tecnología, que facilita el cumplimiento de los usos y funciones de la información. Como resultado se generan cambios en el estado del conocimiento que poseen las personas, la solución de problemas informativos o la toma de decisiones operacionales.(11)

Los sistemas de información:

- ✓ Gestionan los datos que almacenan
- ✓ Gestionan los usuarios que utilizan la información, que además pueden agregarla
- ✓ Poseen una interfaz en correspondencia con la información que contienen
- ✓ Integran no sólo datos e información, sino también programas y otros sistemas de información

1.3 Software de gestión

Un software de gestión es un programa para ordenador que facilita el desarrollo de actividades administrativas, así como el tratamiento, conservación y publicación de información (11). Un sistema de

Capítulo 1: Fundamentación Teórica

gestión ayuda a lograr los objetivos de la organización mediante una serie de estrategias, que incluyen la optimización de procesos, el enfoque centrado en la gestión y el pensamiento disciplinado.

Las empresas que operan en el siglo XXI se enfrentan a muchos retos significativos, entre ellos:

- ✓ Rentabilidad
- ✓ Velocidad de los cambios
- ✓ Capacidad de adaptación
- ✓ Crecimiento
- ✓ Tecnología

Equilibrar estos y otros requisitos empresariales puede constituir un proceso difícil y desalentador. Es aquí donde entran en juego los sistemas de gestión, al permitir aprovechar y desarrollar el potencial existente en la organización.

La implementación de un sistema de gestión eficaz puede ayudar a:

- ✓ Gestionar los riesgos sociales, medioambientales y financieros
- ✓ Mejorar la efectividad operativa
- ✓ Reducir costos
- ✓ Proteger la marca y la reputación
- ✓ Lograr mejoras continuas
- ✓ Potenciar la innovación
- ✓ Eliminar las barreras al comercio

Capítulo 1: Fundamentación Teórica

1.4 Análisis de sistemas similares

A nivel internacional existen varios sistemas con propósitos similares y que de alguna manera permiten caracterizar a empleados, gestionar planes de trabajo y exportar documentos en diferentes formatos con información relevante para la organización. Estos sistemas forman una interfaz entre la gestión de recursos humanos y las tecnologías de la información. Además combinan los RRHH y en particular sus actividades administrativas con los medios puestos a su disposición por la informática, y se refieren en particular a las actividades de planificación y tratamiento de datos para integrarlos en un único sistema de gestión. A continuación, se realiza un estudio de estos sistemas.

GestPeople: herramienta de gestión de Recursos Humanos (RRHH) para todo tipo de empresas. Esta permite acceder de forma rápida a la información relacionada con la organización, estructura organizativa, perfiles profesionales y evaluación del desempeño de los empleados(6). En esta herramienta cada empleado tendrá su propia ficha con toda la información pertinente y estará situado en un organigrama gráfico que servirá de referencia constante. Además, en cualquier momento se podrá actualizar cualquiera de estos parámetros en caso de promociones o bajas. Todas las acciones que se hacen con esta herramienta pueden ser guardadas en documentos en formato Excel o PDF.(7)

WorkMeter: herramienta que permite, mediante datos objetivos y de forma automática, medir y gestionar la actividad productiva de los empleados en una organización. Toda la información que genera este software incide en la toma de decisiones objetivas en la organización, identificar oportunidades de mejora, interpretar modelos de comportamiento, comprobar cargas de trabajo y redefinir la asignación de las tareas.(8)

A partir del estudio de estas soluciones a nivel internacional, se puede constatar que las mismas persiguen el objetivo común de gestionar la información que generan los RRHH. Estas cuentan con algunas funcionalidades semejantes a las del sistema a implementar que sirven de apoyo a la investigación y el desarrollo de la solución, tales como: la evaluación y caracterización de los empleados, la gestión de los planes de trabajo, la gestión de las fichas con información de la vida laboral de cada empleado y la generación de documentos en diferentes formatos.

Capítulo 1: Fundamentación Teórica

Los sistemas cercanos al campo de acción que nos compete y que se analizan a continuación fueron creados en la UCI. En su mayoría son sistemas de gestión que informatizan un proceso sustantivo de la organización o parte de este. A continuación, se realiza un estudio de estos sistemas.

Sistema de Gestión de la Unión de Jóvenes Comunistas de la Facultad 4: herramienta que fue creada principalmente para gestionar el registro, control y recuperación de la información que generan los procesos que se llevan a cabo en la Unión de Jóvenes Comunistas de la Facultad 4(10). Este software no gestiona los reportes de actas, lo que trae consigo que con su uso no se pueda realizar un resumen de los indicadores del cierre de funcionamiento en la Facultad 4, tampoco hace referencia a ningún proceso que intervenga en el diagnóstico sociopolítico en la UCI.

Subsistema para la gestión del ID2 de la UJC de la UCI: subsistema que informatiza toda la gestión referente al proceso del Control del estado de la militancia (ID2) en la UCI, es decir, todo el proceso que rige los pasos por los que transita un militante, las altas y bajas producidas en los comités de base. Este sistema tiene como objetivo tener una mejor organización de los datos que genera el proceso de ID2 y minimizar el tiempo al realizar consultas a dicha información.(10)

Después del estudio de los sistemas anteriores se decide no utilizar ninguno de estos porque aunque están vinculados a la organización no se ajustan a sus necesidades en cuanto al desarrollo del diagnóstico sociopolítico. Se decide realizar un módulo para el SIGPS_UCI que permita llevar el control de la de la información que genera el diagnóstico y agilizar la gestión de sus indicadores.

1.5 Lenguajes de Programación

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo. Está formado por un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura, el significado de sus elementos y expresiones.(12)

1.5.1 Lenguajes de programación del lado del cliente

Los lenguajes de programación del lado del cliente son aquellos que pueden ser directamente interpretados por el navegador y que no necesitan un pre tratamiento. La programación del lado del cliente tiene como principal ventaja que la ejecución de la aplicación se delega al cliente, con lo cual se evita

Capítulo 1: Fundamentación Teórica

recargar al servidor de trabajo. El servidor solo envía el código y es tarea del navegador interpretarlo.(13)

HTML v5

Lenguaje de Marcado para Hipertextos (*HyperText Markup Language*) es el bloque de construcción más básico de una página web que se usa para crear y representarla visualmente. Determina el contenido de la misma, pero no su funcionalidad. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web, como imágenes, listas, vídeos.(14)

En el desarrollo del sistema se utilizó HTML v5.0 fundamentalmente para describir la forma básica que tendrán las páginas. Para organizar el contenido que esta muestra, proporcionando un mayor entendimiento a la hora de corregir errores o hacer cambios en las páginas.

JavaScript

JavaScript es un lenguaje que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente y los navegadores son los encargados de interpretar estos códigos (15). Este lenguaje se utiliza principalmente en el sistema para la validación de los valores de entrada de un formulario web para verificar que son aceptables antes de ser enviado al servidor.

jQuery 1.10.2

jQuery es considerado un Framework de JavaScript, o ambiente de desarrollo. Lo que no es más que un conjunto de utilidades las cuales no necesitan ser programadas, de hecho ya fueron programadas, probadas y se pueden utilizar de una manera muy simplificada. Brinda la posibilidad de realizar tareas complejas en JavaScript de una manera muy sencilla. A través de esta librería de JavaScript se puede mostrar, arrastrar y hacer prácticamente lo que deseemos con los elementos DOM¹(Document Object Model) de nuestra página web (15). Este framework JavaScript se utiliza en la propuesta de solución para cambiar el contenido de una página sin necesidad de recargar la vista.

¹ Permite a los programadores web acceder y manipular las páginas XHTML como si fueran documentos XML.

Capítulo 1: Fundamentación Teórica

CSS v3

Las Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.(16)

En el desarrollo del sistema se utiliza para dar estilo a los documentos HTML, separando el contenido de la presentación y así tener el control centralizado del sitio para agilizar de forma considerable la actualización del mismo.

1.5.2 Lenguajes de programación del lado del servidor

Se denominan lenguajes del lado del servidor a aquellos que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Es una tecnología que consiste en el procesamiento de una petición de un usuario mediante la interpretación de un script² en el servidor web para generar páginas HTML dinámicamente como respuesta.(17)

PHP v5.4

PHP, acrónimo de "PHP: *HyperText Preprocessor*", es un lenguaje de scripting de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser embebido en páginas HTML. Su sintaxis recurre a los lenguajes de programación C, Java y Perl, siendo así sencillo de aprender. El objetivo principal de este lenguaje es permitir a los desarrolladores web escribir dinámica y rápidamente páginas web generadas.(18)

Se decide utilizar este lenguaje para el desarrollo del sistema debido a la gran cantidad de facilidades y utilidades que brinda ya que permite el desarrollo de aplicaciones web dinámicas con acceso a la información almacenada en la base de datos, además, permite la conexión con la mayoría de los motores de bases de datos que se utilizan en la actualidad. Posee una amplia documentación en su sitio web oficial y la colaboración de cientos de expertos lo cual permite la corrección de errores con mayor facilidad.

² Archivo de órdenes o programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.

Capítulo 1: Fundamentación Teórica

1.6 Marcos de trabajo

El marco de trabajo para el desarrollo de aplicaciones web es el responsable de proveer un entorno de ejecución para los proyectos con funcionalidades y características de los conocidos marcos de trabajo. Brinda prestaciones que aseguran un alto grado de calidad e integración de las aplicaciones realizadas en dicha plataforma tecnológica. Aspectos como la seguridad, el monitoreo de trazas, la gestión del personal y los reportes son contenidos entre las funcionalidades de la solución. Permite la obtención de soluciones con un alto grado de robustez, dadas por el número de funcionalidades brindadas durante el desarrollo de soluciones web.(19)

Symfony v2.5

Symfony es un proyecto PHP de software libre que permite crear aplicaciones y sitios web rápidos y seguros de forma profesional. Está desarrollado completamente en PHP 5.3. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.(20)

Para la realización del sistema se utiliza este framework lo cual posibilita:

- ✓ Implementar el sistema de acuerdo a las funcionalidades requeridas.
- ✓ La utilización de la Programación Orientada a Objetos (POO).
- ✓ La utilización de la arquitectura MVC (Modelo Vista Controlador), aunque tiene su propia manera de implementarlo.
- ✓ Utilizar formularios que soportan la validación automática, lo cual asegura mejor calidad de los datos en las bases de datos y una mejor experiencia para el usuario.

Bootstrap v2.0.3

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro

Capítulo 1: Fundamentación Teórica

dispositivo. Esta técnica de diseño y desarrollo se conoce como “*responsive design*” o diseño adaptativo.(21)

Se utiliza en el sistema en todo lo referente al diseño y maquetación del mismo ya que contiene plantillas con tipografías, botones, formularios, menús de navegación, cuadros, entre otros elementos de diseño basados en HTML y CSS, así como extensiones de JavaScript.

Doctrine

Doctrine es un Mapeador de Objetos-Relacional (ORM) escrito en PHP que proporciona una capa de persistencia para objetos PHP. Es una capa de abstracción que se sitúa justo encima de un Sistema Gestor de Bases de Datos (SGBD).(22)

Se escogió Doctrine para la realización del sistema propuesto pues permite llamar a los métodos de un objeto de datos desde distintas partes de la aplicación e incluso desde diferentes aplicaciones. La capa ORM encapsula la lógica de los datos pudiendo hacer cambios que afectan a toda la aplicación únicamente modificando una función. Los ORM suelen implementar mecanismos de seguridad que protegen la aplicación de los ataques más comunes como Inyecciones SQL.

1.7 Servidor Web

Un servidor web es el encargado de aceptar las peticiones de páginas (o recursos en general) que provienen de los visitantes que acceden a nuestro sitio web y gestionar su entrega o denegación, de acuerdo con las políticas de seguridad establecidas.

Algunas de las funciones fundamentales de un servidor web:

- ✓ Atender de manera eficiente, ya que puede recibir un gran número de peticiones HTTP, incluyendo una ejecución multitarea ya que pueden darse peticiones simultáneas. Cualquier petición compleja (por ejemplo, con acceso a base de datos) dejaría colapsado el servicio.
- ✓ Almacenar las peticiones recibidas, errores que se han producido y en general toda aquella información que puede ser registrada y analizada posteriormente para obtener las estadísticas de acceso al sitio web.(23)

Capítulo 1: Fundamentación Teórica

Apache v2.2.6

Apache es un proyecto de código abierto y uso gratuito, multiplataforma, muy robusto y que destaca por su seguridad y rendimiento(24). Además, Apache permite configurar un Hosting Virtual basado en IPs o en nombres, es decir, tener varios sitios web en un mismo equipo (por ejemplo: *nombreweb1.com*, *nombreweb2.com*.) o establecer distintos niveles de control de acceso a la información incluyendo el soporte a cifrado SSL utilizando protocolo seguro HTTPS.(25)

Para el montaje del sistema propuesto se decide utilizar apache dado que corre en una multitud de sistemas operativos, lo que lo hace prácticamente universal, es un servidor altamente configurable que permite personalizar la respuesta ante los posibles errores que puedan ocurrir en el servidor.

1.8 Entorno de desarrollo

Un Entorno de Desarrollo Integrado, traducido del inglés Integrated Development Environment (IDE), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios, así Dev C++, es un IDE para el lenguaje de programación C++; WebDevStudio, lo es en línea para el lenguaje de programación C/C++.(26)

Un IDE puede denominarse como un entorno de programación que ha sido tratado como un programa aplicación. Esto significa que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

NetBeans IDE v8.0.2

NetBeans es un entorno de desarrollo integrado libre, desarrollado principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios y con una comunidad en constante crecimiento.

Todas las funciones de este IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente.(27)

Capítulo 1: Fundamentación Teórica

Se seleccionó este IDE para la programación en PHP, CSS y HTML dado que facilita la realización del sistema informático a desarrollar. Permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Además, es multiplataforma, debido a que fue desarrollado en Java puede ser instalado y utilizado en cualquier sistema operativo (Windows, Mac OS, Linux).

1.9 Sistema gestor de base de datos (SGBD)

Un sistema de gestión de bases de datos es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto. Los SGBD también proporcionan métodos para mantener la integridad de los datos, administrar el acceso de usuarios a los datos y recuperar la información si el sistema se corrompe.

Permite presentar la información de la base de datos en variados formatos. La mayoría de los SGBD incluyen un generador de informes. También puede incluir un módulo gráfico que permita presentar la información con gráficos y tablas.(28)

PostgreSQL

PostgreSQL es un SGBD relacional orientado a objetos y libre. Se ejecuta en todos los principales sistemas operativos³. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una sola empresa, sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. En este servidor los bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos da, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientada a objetos o la programación funcional.(29)

1.10 Lenguaje de modelado

Es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar un diseño de software. Utiliza el paradigma de la orientación a objetos para el desarrollo de software. Este enfoque

³ Linux, Unix, BSDs, Mac OS, BeOS, Windows.

Capítulo 1: Fundamentación Teórica

realiza la construcción de modelos de un sistema por medio de la identificación y la especificación de un conjunto de objetos relacionados que colaboran entre sí de acuerdo a los requerimientos establecidos para el sistema de objetos(30). Se usa generalmente en combinación con una metodología de desarrollo de software para avanzar de una especificación inicial a un plan de implementación y para comunicar dicho plan a todo un equipo de desarrolladores.

Lenguaje de Modelado Unificado

UML (Unified Modeling Language) es un lenguaje gráfico para la especificación, visualización, construcción y documentación de piezas de información usadas o producidas durante el proceso de desarrollo de software. Provee un marco arquitectónico de diagramas para trabajar sobre análisis y diseños orientado a objetos. Es un lenguaje simbólico para expresar modelos y no una estrategia para desarrollarlos.(31)

Se decide usar UML para el desarrollo de la solución propuesta ya que este cubre la documentación de la arquitectura del sistema y todos sus detalles. Proporciona un lenguaje para expresar requisitos y pruebas. Comprende los casos de uso que describe el comportamiento del sistema tal y cómo es percibido por los usuarios. Comprende los componentes y archivos que se utilizan para ensamblar y hacer disponible el sistema físico.

1.11 Metodología de desarrollo de software

Metodología de desarrollo de software hace referencia a un marco de trabajo para estructurar, organizar, planificar y controlar el ciclo de vida de un sistema informático. Es un enfoque estructurado para el desarrollo de software que incluye modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos.(32)

Las metodologías de desarrollo son las encargadas de guiar a los desarrolladores a la hora de crear un software, brindando una serie de pasos definidos y estructurados desde el comienzo hasta el lanzamiento del producto. Esto ha dado lugar a que existan muchas metodologías que se clasifican en pesadas o tradicionales y ligeras o ágiles.

Capítulo 1: Fundamentación Teórica

1.11.1 Metodologías Tradicionales

Aquellas metodologías con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de metodologías tradicionales (o también denominadas metodologías pesadas).(33)

Estas metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto de software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar(34). Por estos motivos quedan descartadas estos tipos de metodologías para utilizarlas en la presente investigación.

1.11.2 Metodologías Ágiles

Las metodologías ágiles están basadas en el desarrollo iterativo e incremental. Existen muchos métodos de desarrollo ágil; la mayoría minimiza riesgos desarrollando software en lapsos cortos. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas.

Una iteración no debe agregar demasiada funcionalidad, sino que la meta es tener una solución parcial sin errores al final de cada iteración. Al final de cada iteración el equipo vuelve a evaluar las prioridades del proyecto.(35)

1.11.2.1 Programación Extrema (XP)

La programación extrema XP es posiblemente el método ágil más conocido y ampliamente utilizado. En la metodología extrema, todos los requerimientos se expresan como escenarios (llamados historias de usuario), los cuales se implementan directamente como una serie de tareas. Los programadores trabajan en parejas y desarrollan pruebas para cada tarea antes de escribir el código. Todas las pruebas se deben ejecutar satisfactoriamente cuando el código nuevo se integra al sistema.(36)

Capítulo 1: Fundamentación Teórica

Las características fundamentales de esta metodología son:

- ✓ **Desarrollo iterativo e incremental:** pequeñas mejoras, unas tras otras.
- ✓ **Programación en parejas:** se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera - el código es revisado y discutido mientras se escribe- es más importante que la posible pérdida de productividad inmediata.
- ✓ **Propiedad del código compartida:** en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.(37)

Proceso Unificado Ágil (AUP)

El Proceso Unificado Ágil (AUP) es una versión simplificada del proceso unificado de desarrollo de software (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas, Modelado Ágil, Gestión de Cambios Ágil, y Refactorización de Base de Datos para mejorar la productividad.(38)

AUP se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo. Para ello, se crean y mantienen listas identificando los riesgos desde etapas iniciales del proyecto. Especialmente relevante en este sentido es el desarrollo de prototipos ejecutables durante la base de elaboración del producto, donde se demuestre la validez de la arquitectura para los requisitos clave del producto y que determinan los riesgos técnicos.

El proceso AUP establece un modelo más simple que el que aparece en RUP por lo que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP.(39)

Capítulo 1: Fundamentación Teórica

Esta metodología establece cuatro fases que transcurren de manera consecutiva y que acaban con hitos claros alcanzados:

- ✓ **Concepción:** El objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
- ✓ **Elaboración:** El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
- ✓ **Construcción:** Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.
- ✓ **Transición:** el sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.(40)

SCRUM

SCRUM es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

Esta metodología realiza entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, SCRUM está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, los requisitos son cambiantes o poco definidos, la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.(41)

Fundamentación de la metodología a utilizar

Como resultado del estudio de las metodologías, se determinó guiar el proceso de desarrollo de la aplicación web con la metodología AUP-UCI, la cual es una variante de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. La misma está compuesta por tres fases: inicio, ejecución y cierre que en su conjunto ayudan a mejorar el proceso de desarrollo de software y dan cumplimiento además a las buenas prácticas que define CMMI-DEV v1.3.

Capítulo 1: Fundamentación Teórica

AUP es una metodología que encuentra un equilibrio entre las metodologías ágiles y las tradicionales. Se adapta, en gran medida, tanto al tipo de proyecto a desarrollar como a las condiciones para la elaboración de este trabajo. Tiene como ventaja que prioriza la gestión de riesgos lo que permite la detección y mitigación temprana de los mismos, se adecua rápidamente a los cambios en el entorno del sistema y proporciona un alto nivel de independencia a la hora de seleccionar las herramientas para el desarrollo del sistema.

1.12 Herramienta CASE

Las Herramientas CASE⁴ son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el diseño de proyectos, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras.(42)

Visual Paradigm v8.0

Es una herramienta CASE que propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Dentro de las características fundamentales se encuentran:

- ✓ Disponibilidad en múltiples plataformas (Windows, Linux)
- ✓ Diseño centrado en casos de uso⁵ y enfocado al negocio que generan un software de mayor calidad
- ✓ Licencia: gratuita y comercial
- ✓ Generación de código para Java y exportación como HTML (43)

⁴ CASE (Ingeniería de Software Asistida por Computación)

⁵ Técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software.

Capítulo 1: *Fundamentación Teórica*

Se decide utilizar esta herramienta en la realización del software por ser muy fácil de usar, tener una interfaz agradable para el usuario y ser multiplataforma. Además, permite aumentar la calidad del software, a través de la mejora de la productividad en el desarrollo y mantenimiento del software. Se utilizó para toda la modelación de los procesos de negocio y el modelado de los datos.

Conclusiones Parciales

En el capítulo se ha realizado un estudio sobre diferentes sistemas informáticos existentes. De los mismos se han tomado ideas a ser utilizadas en el sistema a desarrollar. También se ha realizado un estudio de herramientas y tecnologías informáticas. Basándose en el problema del trabajo y teniendo en cuenta que en la UJC UCI existen los requerimientos de hardware necesarios para instalar los servidores de aplicación y de bases de datos, se decidió desarrollar un módulo para el SIGPS_UCI que permita el desarrollo del diagnóstico sociopolítico de la organización. Esta propuesta de solución facilitaría el trabajo de los funcionarios de la institución, disminuirían los tiempos a la hora de realizar búsquedas de información y se tendría una mejor organización de toda la documentación existente.

Una vez decidido desarrollar este tipo de aplicación, para darle solución a las dificultades existentes en la UJC UCI, se decide desarrollarla utilizando el lenguaje PHP v5.4, con la integración de los frameworks Symfony v2.5 y Bootstrap v2.0.3. Para el modelado del sistema se hace uso del lenguaje de modelado UML y se emplearán las herramientas NetBeans v8.0 y Visual Paradigm en su v8.0 debido a las facilidades que estas brindan. Todo este proceso estará regido y orientado bajo la metodología de desarrollo AUP-UCI siguiendo el escenario dos, que propone la descripción de casos de uso, con el objetivo de mantener una línea base con respecto a la investigación realizada en el desarrollo del SIGPS_UCI.

2 Capítulo 2: Propuesta de solución

2.1 Introducción

Para el desarrollo de una herramienta informática es muy importante conocer cómo funcionan cada uno de los procesos que van a ser automatizados, así como mantener una comunicación importante con el cliente en aras de lograr el producto que se desea y el objetivo por el cual se decidió informatizar dichos procesos. Con vistas a lograr esto se debe realizar una correcta modelación del negocio para comprender la estructura y el funcionamiento de la organización en la cual se va a hacer uso de la herramienta, además se debe realizar un estudio de los problemas actuales e identificar las posibles mejoras que puede traer consigo el producto desarrollado. Otro objetivo esencial es lograr que el usuario final y los desarrolladores tengan un entendimiento común del funcionamiento de la organización y a partir de entonces analizar los requerimientos del sistema que se necesita implantar.

En este capítulo se presentarán los siguientes artefactos: Modelo de dominio, la especificación de requisitos funcionales y no funcionales que tendrá la propuesta de solución, se modelarán los casos de uso y la interacción de los actores con el sistema, así como una descripción detallada de los mismos.

2.2 Modelo de dominio

Un modelo de dominio es una representación de conceptos del dominio de un problema determinado. Permite representar elementos del mundo real y no componentes del software. Provee conocimiento de la nomenclatura del dominio al comunicarle a las partes interesadas los términos importantes y cómo se relacionan entre sí, siendo un artefacto fundamental para realizar el análisis orientado a objetos.(45)

2.2.1 Conceptos del dominio

- ✓ **Aval político:** es una valoración integral de la vida social, combatiente y revolucionaria del militante. El aval se fundamenta en el análisis integral de cada compañero teniendo en cuenta:
 - I. Su actitud como trabajador, estudiante o combatiente.
 - II. El cumplimiento de sus deberes con la organización.
 - III. Los méritos alcanzados.

Capítulo 2: Propuesta de solución

- IV. Los resultados de las evaluaciones anuales (si ha sido objeto de sanciones disciplinarias, si erradicó las faltas o errores cometidos, así como su comportamiento posterior).(3)
- ✓ **Organizaciones de base:** el término organizaciones de base sirve para identificar a las organizaciones de carácter social o político más cercanas a la comunidad a la que sirven. A su vez, las organizaciones de base son asistidas por organizaciones mayores, que pueden ser desde un partido político o secciones de organizaciones nacionales o regionales. En teoría, la organización de base es la célula más pequeña de la UJC en este caso comités de base.
 - ✓ **Evaluación:** Se elabora sobre la base de la actitud mantenida ante el cumplimiento de los deberes del militante como trabajador, estudiante, ante la defensa, las tareas de su organización de base y en las organizaciones de masa. Las formas para evaluar a los militantes son:
 - I. Evaluación para los militantes que son estudiantes.
 - II. Rendición de cuentas para los militantes que son trabajadores o combatientes.
 - ✓ **Plan de trabajo:** Es elaborado por la dirección del comité de base con tareas de su propia iniciativa, estimulando la participación de todos en las actividades planificadas. Debe garantizar que los miembros de la organización de base tengan tareas concretas.

2.2.2 Diagrama del modelo de dominio

Un diagrama de modelo de dominio es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción, en la tarea construcción del modelo de dominio, presentado como uno o más diagramas de clases y que contiene, no conceptos propios de un sistema de software sino de la propia realidad física.

Los modelos de dominio pueden utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir. (45)

Capítulo 2: Propuesta de solución

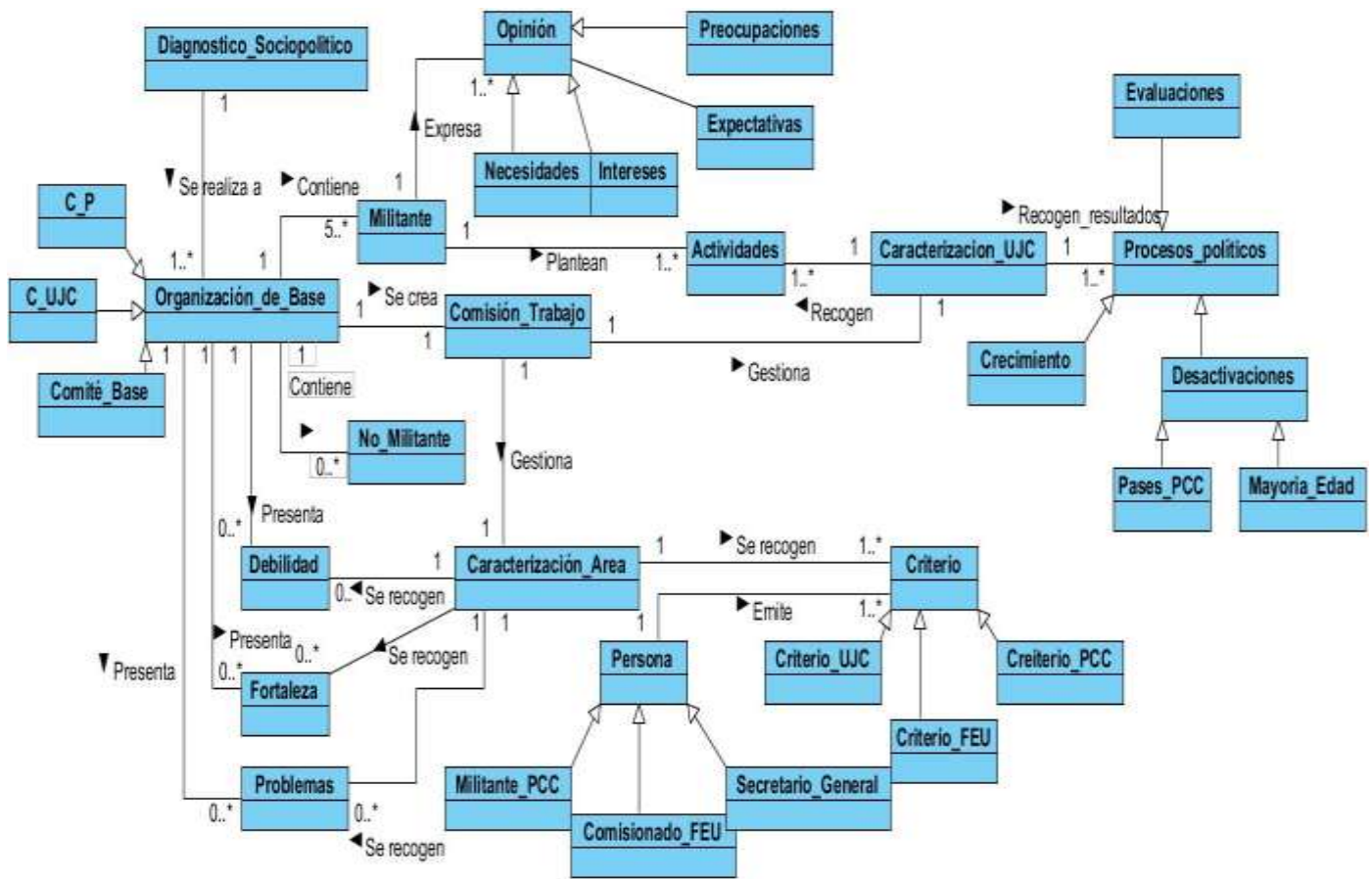


Figura 1. Diagrama de modelo del dominio (Elaboración propia)

2.3 Descripción del sistema propuesto

El sistema propuesto se corresponde con varios procesos sustantivos que desarrolla la UJC en su funcionamiento interno, el diagnóstico sociopolítico, el proceso de evaluación, condecoraciones y el otorgamiento del aval para integrar las filas del PCC. La solución se integrará al Sistema Integral para la Gestión de los Procesos Sustantivos de la UJC en la UCI (SIGPS_UCI). El sistema tiene como principal objetivo permitir a las estructuras de dirección de la UJC, la navegación por cada uno de estos procesos a través del índice.

El sistema proporcionará a los usuarios, funcionalidades que harán del contenido de este un ambiente cargado de facilidades, dentro de las cuales se encuentran:

Capítulo 2: Propuesta de solución

- ✓ Gestionar las estructuras de la UJC brindando la oportunidad de acceder, obtener y modificar información de los comités de la UCI.
- ✓ Gestionar las comisiones de trabajo para el desarrollo del diagnóstico sociopolítico.
- ✓ Generar informes de caracterización de los jóvenes, áreas de trabajo y de la organización en general.
- ✓ Generar matriz DAFO para apoyar la toma de decisiones en la organización.
- ✓ Gestionar las evidencias para el otorgamiento del aval político y científico para integrar las filas del PCC.
- ✓ Gestionar el plan de trabajo de la organización y el cumplimiento del mismo.

2.4 Requerimientos funcionales

Los requisitos del software expresan las necesidades y los apremios colocados en un producto de software que contribuye a la solución de un cierto problema del mundo real.

La especificación de los requisitos de software describe las funcionalidades, servicios y restricciones que debe cumplir el sistema a desarrollar y permiten un mayor entendimiento entre el cliente y el equipo de desarrollo.

El sistema debe permitir realizar:

RF1 Autenticar usuario

RF2 Exportar a pdf

RF2.1 Listado del universo juvenil de un C/B

RF2.2 Listado de militantes de un C/B

RF2.3 Informe de caracterización del área

RF2.4 Informe de caracterización de los jóvenes

RF2.5 Informe del proceso de pases al PCC

RF2.6 Informe del proceso de evaluación de los militantes

RF2.7 Informe de caracterización de los jóvenes no militantes

RF2.8 Plan de trabajo

Capítulo 2: Propuesta de solución

RF3 Gestionar comisión

RF3.1 Insertar comisión

RF3.2 Modificar comisión

RF3.3 Eliminar comisión

RF3.4 Mostrar comisión

RF4 Gestionar característica del área

RF4.1 Insertar característica del área

RF4.2 Insertar criterio de la FEU

RF4.3 Insertar criterio del PCC

RF4.4 Insertar criterio de profesores

RF4.5 Insertar problemas asociados al área

RF4.6 Mostrar característica del área

RF4.7 Eliminar característica del área

RF5 Gestionar indicadores de la matriz DAFO

RF5.1 Insertar debilidades de la UJC

RF5.2 Insertar amenazas para la UJC

RF5.3 Insertar oportunidades en la UJC

RF5.4 Insertar fortalezas de la UJC

RF5.5 Visualizar matriz DAFO

RF7 Gestionar caracterización de los jóvenes

RF7.1 Crear caracterización de los jóvenes

RF7.2 Insertar necesidad individual de cada joven

RF7.3 Insertar intereses de cada joven

RF7.4 Insertar preocupaciones de los jóvenes

RF7.5 Insertar expectativa

RF7.6 Mostrar caracterización de los jóvenes

RF7.7 Eliminar caracterización de los jóvenes

RF8 Gestionar plan de trabajo

Capítulo 2: Propuesta de solución

RF8.1 Crear plan de trabajo

RF8.2 Insertar tarea

RF8.3 Insertar tarea general

RF8.4 Insertar actividades

RF8.5 Mostrar plan de trabajo

RF8.6 Eliminar plan de trabajo

RF9 Gestionar evidencias

RF9.1 Crear evidencias

RF9.2 Modificar evidencias

RF9.3 Mostrar evidencias

RF9.4 Eliminar evidencias

RF10 Gestionar expediente de las BTJ

RF10.1 Crear expediente de las BTJ

RF10.2 Modificar expediente de las BTJ

RF10.3 Mostrar expediente de las BTJ

RF10.4 Eliminar expediente de las BTJ

RF11 Gestionar ficha técnica

RF11.1 Crear ficha técnica

RF11.2 Modificar ficha técnica

RF11.3 Mostrar ficha técnica

RF11.4 Eliminar ficha técnica

RF12 Gestionar aval científico

RF12.1 Crear aval científico

RF12.2 Modificar aval científico

RF12.3 Mostrar aval científico

RF12.4 Eliminar aval científico

RF13 Gestionar líneas de investigación

Capítulo 2: Propuesta de solución

- RF13.1** Crear líneas de investigación
- RF13.2** Modificar líneas de investigación
- RF13.3** Mostrar líneas de investigación
- RF13.4** Eliminar líneas de investigación

RF14 Gestionar tutorías

- RF14.1** Crear tutorías
- RF14.2** Modificar tutorías
- RF14.3** Mostrar tutorías
- RF14.4** Eliminar tutorías

RF15 Gestionar publicaciones

- RF15.1** Crear publicaciones
- RF15.2** Modificar publicaciones
- RF15.3** Mostrar publicaciones
- RF15.4** Eliminar publicaciones

RF16 Gestionar méritos

- RF16.1** Crear mérito
- RF16.2** Modificar mérito
- RF16.3** Mostrar mérito
- RF16.4** Eliminar mérito

RF17 Gestionar evaluación

- RF17.1** Crear evaluación
- RF17.2** Mostrar evaluación
- RF17.3** Eliminar evaluación

RF18 Gestionar aval del partido

- RF18.1** Crear aval del partido
- RF18.2** Mostrar aval del partido
- RF18.3** Eliminar aval del partido

RF19 Gestionar caracterización de la UJC

RF19.1 Crear caracterización de la UJC

RF19.2 Insertar señalamiento

RF19.3 Insertar estímulo

RF19.4 Insertar reconocimiento

RF19.5 Mostrar caracterización de la UJC

RF19.6 Eliminar caracterización de la UJC

2.5 Requerimientos no funcionales

Requerimientos de Interfaz o apariencia:

1. Organizar los datos en paneles.
2. La entrada de datos incorrecta será detectada claramente e informada al usuario.
3. Los iconos utilizados deberán corresponderse con la función que realizan.

Requerimientos de usabilidad.

1. Debe ser práctico y fácil de usar.

Requerimientos de rendimiento:

1. La velocidad en el tiempo de respuesta y procesamiento de la información en el momento de generar información en el diagnóstico sociopolítico debe ser mínima para lograr un producto eficiente, cumpliendo con la gestión de la información que se necesita.
2. Se necesita un mayor rendimiento en el instante de exportar alguna información del proceso del diagnóstico sociopolítico debido al volumen de información.

Requerimientos de soporte:

1. Se necesita un servidor de bases de datos que soporte grandes volúmenes de datos.
2. Para garantizar el soporte de esta aplicación, se requiere PostgreSQL como servidor de bases de datos, PHP en su versión 5 o superior, un servidor Web Apache 2.0 o superior, por parte del cliente se requiere un navegador capaz de interpretar JavaScript y CSS, el sistema debe dar la posibilidad de ser mejorado, así como de incorporarle nuevos servicios en caso de ser necesarios.

Capítulo 2: Propuesta de solución

Requerimientos de seguridad:

1. Confidencialidad: La información manejada por el sistema deberá estar protegida de acceso no autorizado y divulgación.
2. Integridad: La información manejada por el sistema será objeto de cuidadosa protección contra estados inconsistentes y corrupción.
3. Disponibilidad: A los usuarios autorizados se les deberá garantizar el acceso a la información solicitada en todo momento.

Requerimientos de diseño:

1. El sistema se desarrollará siguiendo la metodología AUP (Proceso Unificado Ágil). Se utilizará Visual Paradigm v8.0 como herramienta CASE para el modelado UML. Se utilizará el lenguaje de programación PHP 5.4 con el framework Symfony v2.5. El servidor de base de datos será PostgreSQL v8.3.8.

Requerimientos de portabilidad:

1. El software es multiplataforma.

Requerimientos de confiabilidad:

1. El sistema de BD debe contar con soporte para la recuperación en el caso de fallos y errores.
2. El software debe ser capaz de mantener la integridad de los datos.

Requerimientos de Software:

1. Se requiere para el funcionamiento del sistema disponer de un servidor que cuente con un Sistema Operativo Linux o un Sistema Operativo Windows 2000 o superior, Apache 2.4 y PostgreSQL v9.3 o versiones superiores. Los usuarios del sistema deberán contar con un navegador Internet Explorer 8 o Mozilla Firefox v33.0 o superior, para poder acceder a las opciones que brinda el sistema.

Requerimientos de Hardware:

1. Microprocesador Pentium IV a 3.0 GHz o superior.
2. 512 MB de RAM o superior.

Capítulo 2: Propuesta de solución

2.6 Modelo de casos de uso del sistema

El modelo de casos de uso permite que los desarrolladores del software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. El modelo de casos de uso sirve como acuerdo entre clientes y desarrolladores, y proporciona la entrada fundamental para el análisis, el diseño y las pruebas. Un modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones y además describe lo que hace el sistema para cada tipo de usuario.(46)

2.6.1 Descripción de los actores del sistema

Un actor es un usuario del sistema. Incluye usuarios humanos y otros sistemas computarizados. Un actor usa un caso de uso para desempeñar alguna porción de trabajo que es de valor para el negocio. (47)

Tabla 2. Actores del sistema y su descripción

Actor	Descripción
Secretario del C_UJC	Actor que tendrá acceso total a todas las funcionalidades del sistema, para realizar operaciones de gestión sobre la información que se maneja en el mismo.
Organizador	Es el encargado de la gestión, organización, funcionamiento de todos los procesos que se realizan en la organización.
Político	Es el encargado de la gestión de las preparaciones políticas de la organización a todos los niveles.
Secretario del C/P	Es el encargado de visualizar y modificar todos los datos referentes a los CB que se le subordinan.
Secretario del C/B	Es el encargado de visualizar y modificar todos los datos de los procesos realizados en su CB.
Usuario	Es el encargado de autenticarse en el sistema.

2.6.2 Diagrama de actores del Sistema

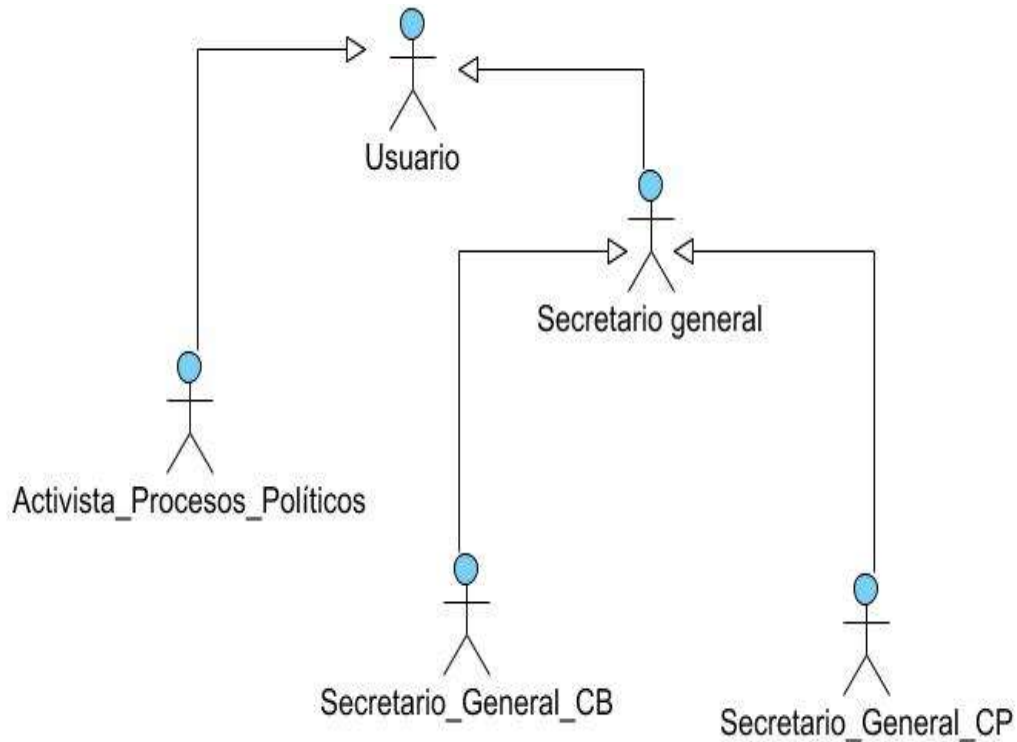


Figura 2. Diagrama de actores del sistema (Elaboración propia)

2.6.3 Diagrama de casos de uso del sistema

El modelo de casos de uso describe la funcionalidad propuesta del nuevo sistema. Un caso de uso representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema. Un caso de uso es una unidad simple de trabajo significativo. Cada caso de uso tiene una descripción que describe la funcionalidad que se construirá en el sistema propuesto. Un caso de uso puede "incluir" la funcionalidad de otro caso de uso o "extender" a otro caso de uso con su propio comportamiento.(47)

Capítulo 2: Propuesta de solución

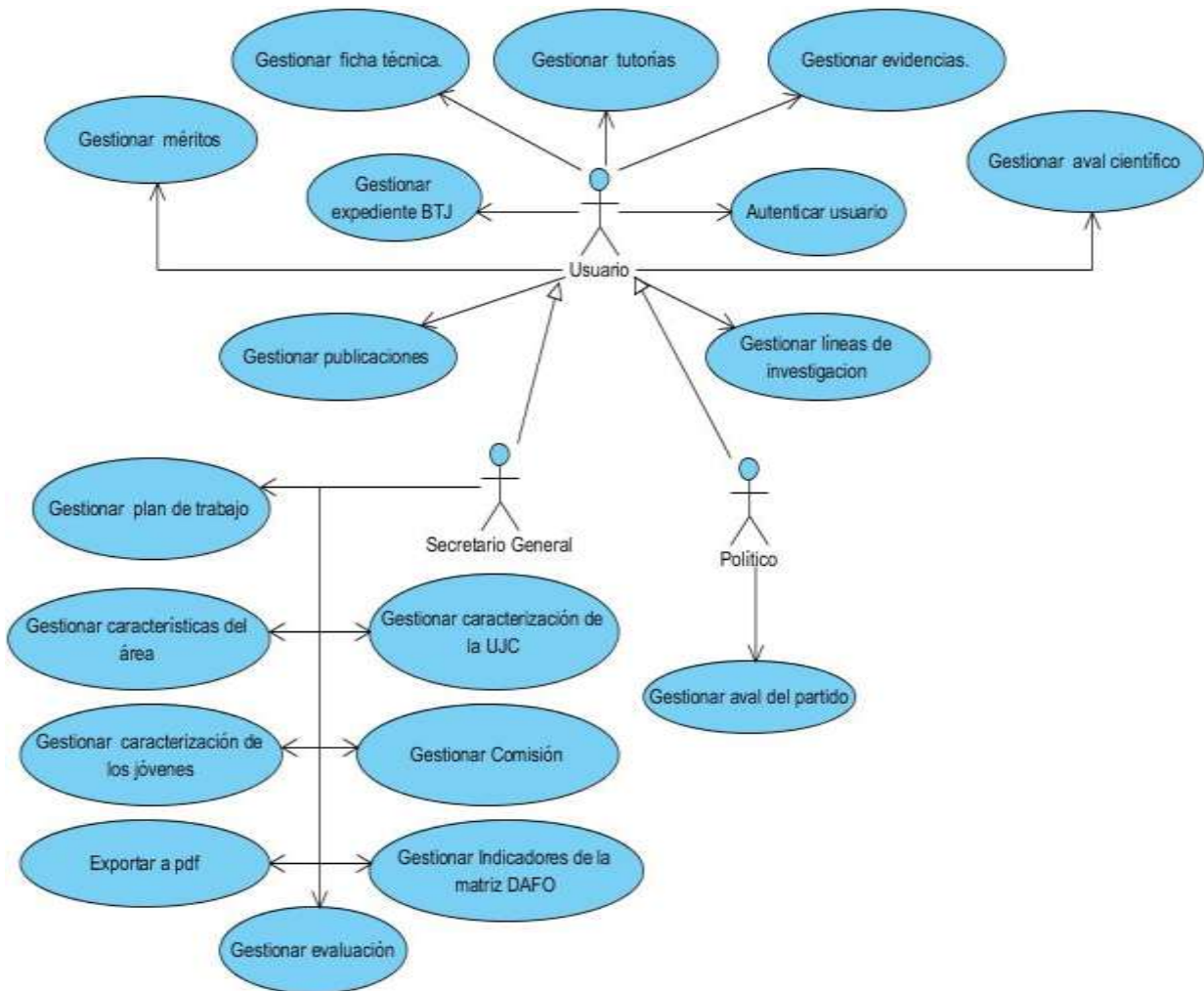


Figura 3. Diagrama de casos de uso del sistema (Elaboración propia)

2.6.4 Patrones de casos de uso del sistema

Son patrones que capturan mejores prácticas para modelar CU, permiten representar los comportamientos que deben existir en el sistema. Estos patrones ayudan a un mejor entendimiento de la funcionalidad de un sistema y la interacción del usuario con el mismo.

CRUD: Es el acrónimo de Crear, Obtener, Actualizar y Borrar (del original en inglés: **Create**, **Read**, **Update** and **Delete**). Este patrón es usado en los casos donde se realizan tareas de gestión, ejemplo de ellos los casos gestionar comisión, gestionar militante, gestionar universo juvenil y otros. En los casos donde se

Capítulo 2: Propuesta de solución

realizan parte de estas tareas se denomina CRUD parcial. Este patrón se utilizó en casos como: gestionar comisión, gestionar méritos y gestionar publicaciones.(48)

Múltiples Actores: Este patrón se refiere al ingreso de dos o más actores del sistema y todos tienen un rol común. Este patrón se utilizó en algunos casos de uso como: gestionar expediente de las BTJ , gestionar línea de investigación, gestionar tutorías y autenticar usuario.(48)

2.6.5 Descripción textual de los casos de uso del sistema

A continuación se muestran las descripciones textuales de los casos de usos Gestionar comisión y Generar informe de cumplimiento de las actividades. Las demás descripciones se encuentran en el Anexo No 1.

Tabla 3. Descripción textual del caso de uso Gestionar comisión (Elaboración propia)

Caso de uso	Gestionar comisión
Actores	Usuario
Resumen	El caso de uso inicia cuando el actor accede a la ventana inicial que el sistema muestra según su rol y accede a la opción gestionar comisión. Muestra un listado con las comisiones registradas y la opción de crear una nueva comisión. Además por cada una permite: modificar y eliminar comisión. El actor crea, modifica o elimina una comisión finalizando así el caso de uso.
Precondiciones	El usuario debe estar autenticado con el rol de Secretario C/P.
Postcondición	Se creó, se modificó o se eliminó una comisión.
Referencias	RF2, RF2.1, RF2.2, RF2.3
Flujo de eventos	
Flujo básico	

Capítulo 2: Propuesta de solución

Acciones del actor	Respuesta del sistema
1. El caso de uso inicia cuando el actor accede al sistema y selecciona la opción Comisión en el menú Diagnóstico Sociopolítico.	
	2. El sistema muestra los datos de cada una de las comisiones almacenadas en el sistema. Y permite: <ul style="list-style-type: none">• Crear nueva comisión• Modificar comisión. Ver sección 1: “Modificar comisión”• Eliminar comisión. Ver sección 2: “Eliminar comisión”<ul style="list-style-type: none">• Visualizar comisión. Ver sección 3: “Visualizar comisión”
3. Selecciona la opción Crear comisión	
	4. El sistema brinda la posibilidad de insertar los siguientes datos: <ul style="list-style-type: none">• Nombre de la comisión• Nombre del jefe de la comisión• Nombre de los miembros de la comisión• Comité de base asociado a esa comisión Y las opciones: <ul style="list-style-type: none">• Aceptar• Cancelar
5. Llena los campos requeridos en el formulario y	

Capítulo 2: Propuesta de solución

selecciona la opción Aceptar.	
	6. Valida los datos introducidos.
	7. Registra una nueva comisión en el sistema.
	8. Muestra un mensaje: "Comisión creada con éxito".
	9. El caso de uso termina
Flujos alternos	
*.a Selecciona la opción Cancelar	
Acciones del actor	Respuesta del sistema
	4. a.1 El sistema regresa a la vista anterior.
	4. a.2 El caso de uso termina.
4.a Existen campos vacíos	
Acciones del actor	Respuesta del sistema
	4. a.1 El sistema muestra un mensaje "Por favor, rellene este campo".
	4. a.2 Regresa al paso 4.
Sección 1: Modificar comisión.	
Flujo básico	
Acción del actor	Respuesta del sistema.
1. Selecciona la opción modificar solicitud.	
	2. El sistema muestra una ventana auxiliar con los datos a modificar de la comisión. Y las opciones: <ul style="list-style-type: none"> • Aceptar • Cancelar

Capítulo 2: Propuesta de solución

3. Actualiza los campos requeridos en el formulario y selecciona la opción aceptar.	
	4. Valida los datos introducidos.
	2. Actualiza los datos de la comisión.
	3. El caso de uso termina.
Flujo alternativo	
Sección 1: “Modificar comisión”.	
*.a Selecciona la opción cancelar.	
Acciones del actor	Respuesta del sistema
	4. a.1 El sistema regresa a la vista anterior.
	4. a.2 El caso de uso termina.
4.a Existen campos vacíos	
Acciones del actor	Respuesta del sistema
	4. a.1 El sistema muestra un mensaje “Verifique que no existan campos vacíos”.
	4. a.2 Muestra el indicador de campos que no pueden estar vacíos.
Sección 2: “Eliminar comisión”.	
Flujo básico	
Acciones del actor	Respuesta del sistema
1. Selecciona la opción Eliminar comisión.	
	2. Elimina la comisión del sistema.
	3. Termina el caso de uso
Flujo alternativo	
Sección 2: “Eliminar comisión”.	
*.a Selecciona la opción cancelar.	

Capítulo 2: Propuesta de solución

Acciones del actor	Respuesta del sistema
	*.a.1 Cierra el mensaje de confirmación.
	*.a.2 Retorna a la vista anterior.

Sección 3: “Visualizar comisión”

Flujo básico

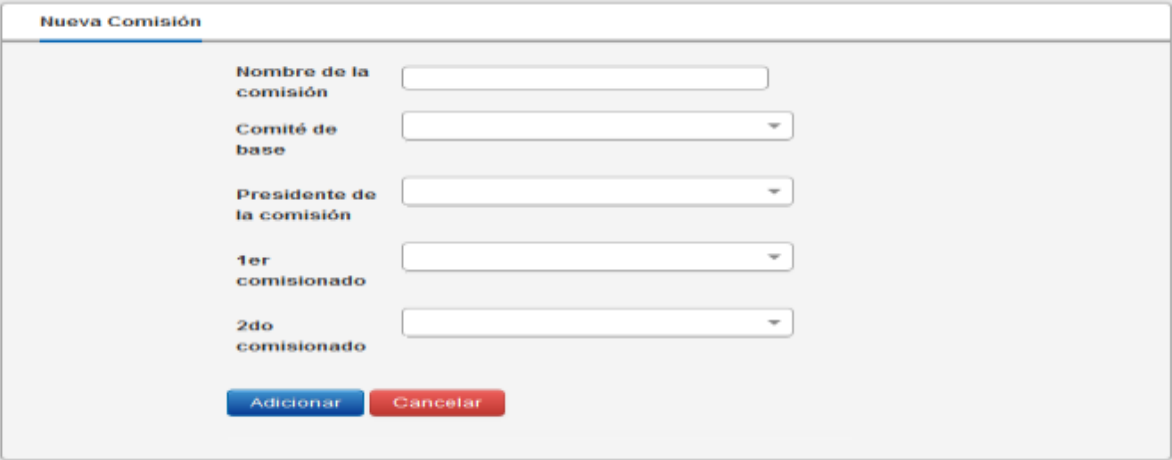
Acciones del actor	Respuesta del sistema
1. Selecciona la opción visualizar comisión.	
	2. El sistema muestra una ventana auxiliar con todos los datos almacenados de la comisión seleccionada. Y la opción: ✓ Aceptar
3. Selecciona la opción aceptar	
	4. Cierra el diálogo actual.

Prototipos de interfaz

The screenshot displays a web application interface for managing commissions. At the top left, there is a blue button labeled 'Nuevo'. Below it, the page title 'Comisión' is shown. A search bar with the placeholder text 'Buscar:' is present. The main content area features a table with the following data:

Nombre de la Comisión	Comité de base asociado	Opciones
C4501	4501	

Below the table, it indicates 'Mostrando 1 a 1 de registros 1'. At the bottom left, a status bar shows 'Total de comisiones: 1'.



Formulario "Nueva Comisión" con los siguientes campos:

- Nombre de la comisión:
- Comité de base:
- Presidente de la comisión:
- 1er comisionado:
- 2do comisionado:

Botones:

2.7 Modelo de análisis

El modelo de análisis es la primera representación técnica de un sistema. Utiliza una mezcla de formatos en texto y diagramas para representar los requisitos del software, las funciones y el comportamiento. De esta manera se hace mucho más fácil comprender dicha representación, ya que es posible examinar los requisitos desde diferentes puntos de vista aumentando la probabilidad de encontrar errores, de que surjan debilidades y de que se descubran descuidos.(49)

2.7.1 Diagrama de clases del análisis

Uno de los elementos del modelo de análisis son las clases del análisis o algunas veces llamados objetos de análisis. Las clases de análisis son clases estereotipadas que representan un modelo conceptual inicial para elementos en el sistema que tienen responsabilidad y comportamiento. (49) Existen tres tipos de clases de análisis y ellas son usadas en todo el modelo de análisis:

- ✓ Interfaz
- ✓ Control
- ✓ Entidad

Capítulo 2: Propuesta de solución

A continuación se muestra el diagrama de clases del análisis del caso de uso Gestionar comisión.

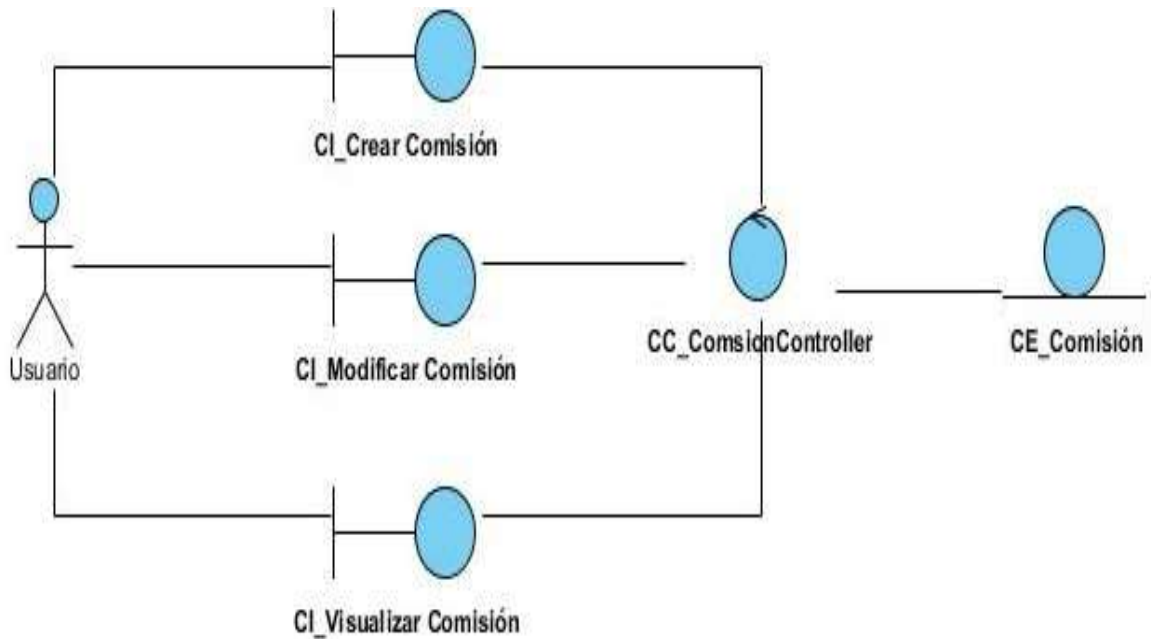


Figura 4. Diagrama de clases del análisis del caso de uso Gestionar comisión (Elaboración propia)

2.7.2 Diagrama de colaboración del análisis

Los diagramas de colaboración, por su parte, son una representación más concreta y detallada que los diagramas de clases del análisis, aunque también representan la parte estática del sistema conteniendo las clases y sus relaciones.(49)

Capítulo 2: Propuesta de solución

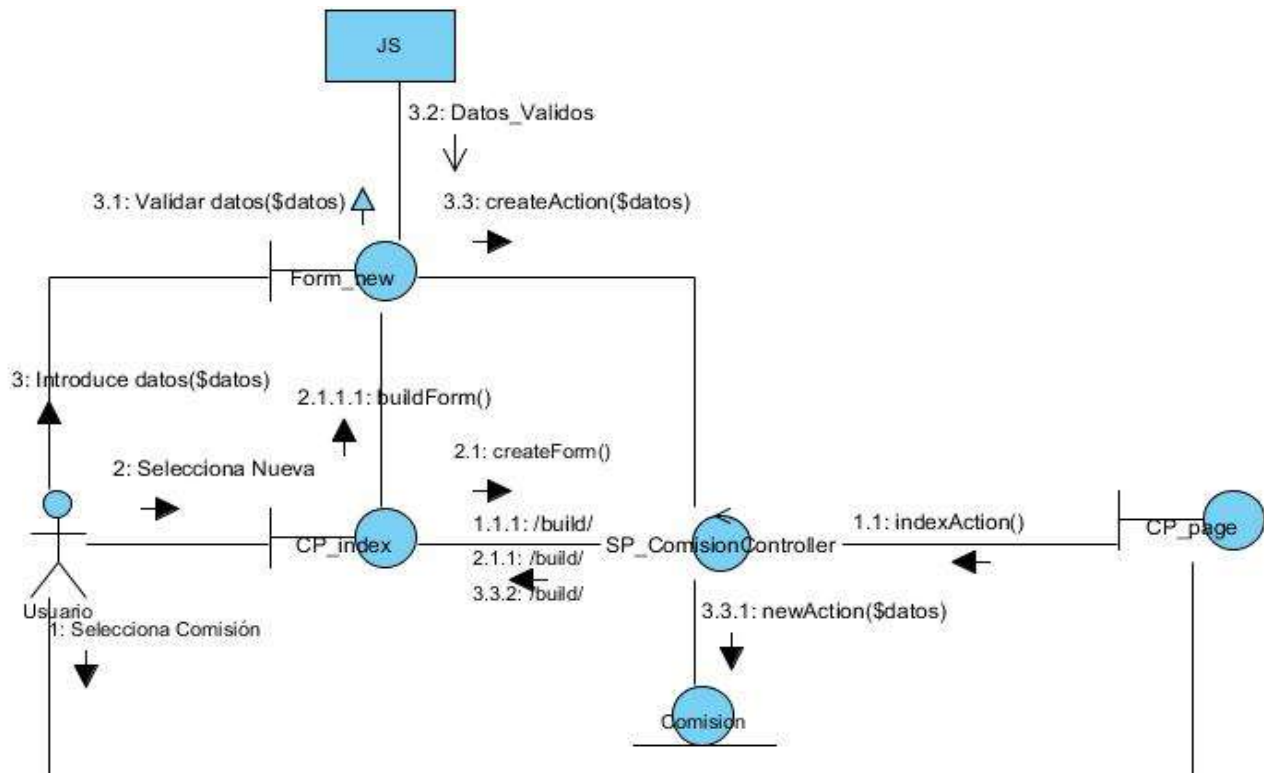


Figura 5. Diagrama de clases de colaboración del caso de uso Gestionar comisión (Elaboración propia)

Conclusiones Parciales

En este capítulo se hizo una descripción de la solución propuesta, obteniéndose a partir del análisis del proceso plasmado en las descripciones textuales, se confeccionó un listado con las funciones que debe tener el sistema, los cuales se representaron mediante un conjunto de diagramas de casos de uso. Partiendo de este punto se puede empezar a construir el sistema, tratando de que se cumplan todos los requerimientos y las funciones que se han considerado necesarias en este capítulo.

Capítulo 3: Desarrollo de la propuesta de solución

3 Capítulo 3: Desarrollo de la propuesta de solución

3.1 Introducción

La metodología de desarrollo AUP-UCI comprende varias fases: inicio, ejecución y cierre. En esta se generan varios artefactos necesarios para la comprensión del sistema por parte de los usuarios y los desarrolladores y una vez generado los mismos se procede a la implementación de la propuesta de solución y la realización de pruebas al sistema obtenido.

En este capítulo se abordan los patrones utilizados para el diseño de la propuesta de solución, se elaborará el modelado del diseño del sistema, debido a que el equipo de trabajo y el cliente poseen una total claridad de los requerimientos, además se generan los diagramas de clases, diagramas de secuencia y diagrama de despliegue. Para la implementación y las pruebas se realizan los diagramas de componentes y el diseño de los casos de prueba.

3.2 Patrones de diseño

Los patrones de diseño son soluciones para problemas típicos y recurrentes que se pueden encontrar a la hora de desarrollar una aplicación. Estos básicamente son modelos de muestra que sirven como guía para que los programadores trabajen sobre ellos. Los patrones pretenden proporcionar catálogos de elementos reusables en el diseño del sistema de software, evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente, formalizar un vocabulario común entre diseñadores, estandarizar el modo en el que se realiza el diseño y facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento existente. (50)

3.3 Patrón arquitectónico Modelo – Vista – Controlador en Symfony

El framework Symfony está basado en el patrón arquitectónico Modelo Vista Controlador (MVC). Es un patrón de arquitectura de las aplicaciones software que separa la lógica de negocio de la interfaz de usuario. Facilita la evolución por separado de ambos aspectos. Incrementa reutilización y flexibilidad. Este patrón se especifica bajo la proposición de dividir la aplicación en tres tipos de elementos, el modelo, las vistas y controladores. Estos elementos están separados por límites abstractos lo que convierte a MVC más paradigma que patrón, ya que la comunicación entre sí a través de esos límites no se especifica más. (50)

Capítulo 3: Desarrollo de la propuesta de solución

La manera en que los elementos dentro de MVC se comunican difieren y no solo lo diferencia el tipo de aplicación que se está describiendo (Desktop, WEB), sino también por la parte de la aplicación que actualmente está mirando (frontend, backend). (50)

Modelo: Maneja el comportamiento y los datos del dominio de la aplicación, responde a los requerimientos de información acerca de su estado (usualmente desde la vista) y responde a las instrucciones para cambiar de estado (usualmente desde el controlador). Elementos (objetos) que contienen los datos y definen la lógica para manipular dichos datos. A menudo los objetos tienen una naturaleza reutilizable, distribuida, persistente y portátil para una variedad de plataformas.(50)

Vista: Maneja el despliegue de la información. Hace referencia a los elementos que representan algo visible en la interfaz de usuario, por ejemplo, un panel o botones. Con el fin de mostrar datos de los objetos de modelo es posible que desee crear sus propios objetos personalizados.(50)

Controlador: Interpreta las acciones del usuario de teclado y ratón, informando al modelo y/o a la vista para cambiar apropiadamente sus estados. Actúa como un mediador entre los objetos del modelo y la vista. Un objeto Controlador comunica datos de ida y vuelta entre los objetos del modelo y de la vista. Un controlador también realiza todas las tareas específicas de la aplicación, tales como la entrada del usuario o la carga de procesamiento de datos de configuración.(50)

3.4 Patrones GOF

Los patrones de diseño se centran en una solución a un problema concreto en el ámbito de la programación orientada a objetos, proponiendo una solución genérica de clases y relaciones para resolver dicho problema. Favorecen la reutilización de código y ayudan a construir software basado en la reutilización. (51)

Los patrones GOF (Gang of Four) se dividen en tres categorías diferentes:

- ✓ **De creación:** Forma de crear instancias. Abstraer y ocultar cómo son creados e inicializados los objetos (Factoría (Factory), Singleton).
- ✓ **Estructurales:** Cómo se combinan clases y objetos para formar nuevas estructuras y proporcionar

Capítulo 3: Desarrollo de la propuesta de solución

nuevas funciones (Adaptador (Adapter)).

- ✓ **De comportamiento:** Ayudan a definir la comunicación e interacción de los objetos. Reducir acoplamiento (Observer).

Para el desarrollo de la solución se utilizaron los patrones:

Decorador (Decorator): el patrón decorador permite añadir responsabilidades a objetos concretos de forma dinámica. Los decoradores ofrecen una alternativa más flexible que la herencia para extender las funcionalidades. En Symfony el archivo layout.php contiene al <<SP_index.php>> de la página (plantilla global), esta almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada una.(52)

Observador (Observer): este patrón define una dependencia “uno-a muchos” entre objetos, para que, cuando uno de ellos cambie su estado, todos los que dependan de él sean avisados y puedan actualizarse convenientemente.(52) Este patrón es utilizado al realizar algún cambio en los datos de alguna organización de base.

3.5 Patrones GRASP

Según el patrón de arquitectura antes seleccionado en la fundamentación teórica y las tecnologías y herramientas elegidas para el desarrollo del software se decide emplear en el diseño del sistema patrones **GRASP** (Patrones Generales de Software para Asignar Responsabilidades, según sus siglas en inglés) para apoyar la implementación del sistema, facilitar la reutilización de código y lograr que el mismo tenga alto grado de mantenibilidad.

Los patrones **GRASP** describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente el software orientado a objetos.(53)

Para la propuesta de solución se usaron los siguientes:

Capítulo 3: Desarrollo de la propuesta de solución

Controlador: Asigna la responsabilidad del manejo de mensajes de los eventos de un sistema a una clase que represente el sistema global, organización, elemento activo del mundo real o manejador artificial de los eventos del sistema. El empleo de este patrón será muy útil para aquellas clases encargadas de registrar datos de entidades que manejará el sistema, y este se aplica en las clases: **ComisionController**, **EvaluacionController**, **AvalPCCController** y demás controladoras.

Creador: Se emplea cuando se le aplica la responsabilidad a una clase determinada de crear una o más instancias de otra. Esto sucede en caso de que la clase creadora contenga, agregue, registre, utilice o posea datos de inicialización de objetos de alguna clase determinada. En la clase <<SystemConfigController.php>> se encuentran las acciones definidas para el sistema y se ejecutan en cada una de ellas. En dichas acciones se crean los objetos de las clases que representan las entidades, lo que evidencia que la clase <<SystemConfigController.php>> es “creador” de dichas entidades.

Experto: Se basa en asignar la responsabilidad a una clase que se encargará de ser experto de la información, la misma cuenta con la información necesaria para cumplir con la responsabilidad. (43) Al aplicarlo al sistema, este patrón permitirá identificar las clases que poseen la información requerida, para que luego estas puedan ser manipuladas por otras. Se evidencia en las clases **Entidad**, las cuales son expertas en su propia información, tales como: **Comision**, **AvalPCC**, **Evaluacion** y demás entidades.

Alta cohesión: Se basa en asignar una responsabilidad de modo que la cohesión siga siendo alta, resolviendo el problema de mantener la complejidad en las clases dentro de los límites manejables, permitiendo definir en las clases las funcionalidades relacionadas para que estas no realicen un trabajo enorme. El controlador << SystemConfigController.php >> asigna responsabilidades de controlar el flujo de eventos del sistema a clases específicas. Este delega en otras clases (ejemplo <<SP_index.php>>) las actividades con las que mantiene un modelo de alta cohesión.

3.6 Modelado de diseño

El modelo de diseño crea un modelo de software enfocado en la representación de los datos, las funciones y el comportamiento requerido. Permite al ingeniero de software modelar el sistema o producto que se va a construir, posibilitando evaluar su calidad y efectuar mejoras antes de generar el código. Con este se obtiene una representación arquitectónica, de interfaz y despliegue del sistema.(54)

Capítulo 3: Desarrollo de la propuesta de solución

3.6.1 Diagrama de clases del diseño

Un diagrama de clases de diseño muestra la especificación para las clases software de una aplicación. Incluye la siguiente información:

- ✓ Clases, asociaciones y atributos
- ✓ Interfaces, con sus operaciones y constantes
- ✓ Métodos
- ✓ Navegabilidad
- ✓ Dependencias

A diferencia del Modelo Conceptual, un Diagrama de Clases de Diseño muestra definiciones de entidades software más que conceptos del mundo real(55). El Diagrama de Clase es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones.(56)

Se muestra a continuación el diagrama de clases del diseño del caso de uso Gestionar comisión, donde se observan los paquetes que agrupan las clases participantes y sus respectivas relaciones. El resto de los diagramas se encuentran en el Anexo, se han diseñado los diagramas más representativos según la necesidad del equipo de desarrollo.

Capítulo 3. Desarrollo de la propuesta de solución

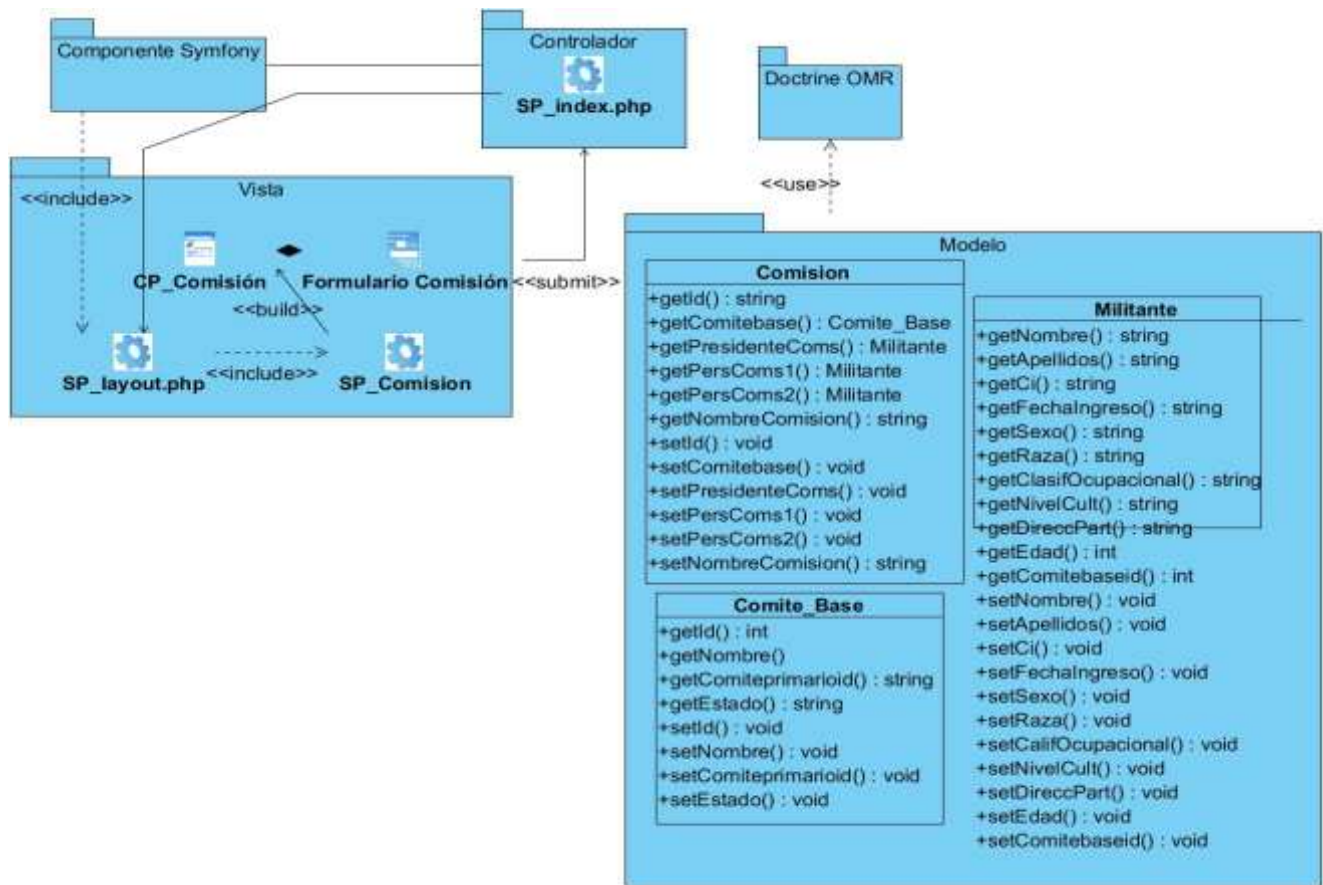


Figura 6. Diagrama de clases del diseño del caso de uso Gestionar comisión (Elaboración propia)

3.6.2 Diagrama de secuencia

El diagrama de secuencia en UML muestra la forma en que los objetos se comunican entre sí al transcurrir el tiempo. El diagrama muestra los objetos participando en la interacción y la secuencia de mensajes intercambiados. Este contiene objetos con sus líneas de vida y mensajes de intercambio entre objetos en una misma secuencia ordenada.(57)

A continuación, se presenta el diagrama de secuencia correspondiente al caso de uso Gestionar comisión, el resto de los diagramas se encuentran en el Anexo No 4.

Capítulo 3: Desarrollo de la propuesta de solución

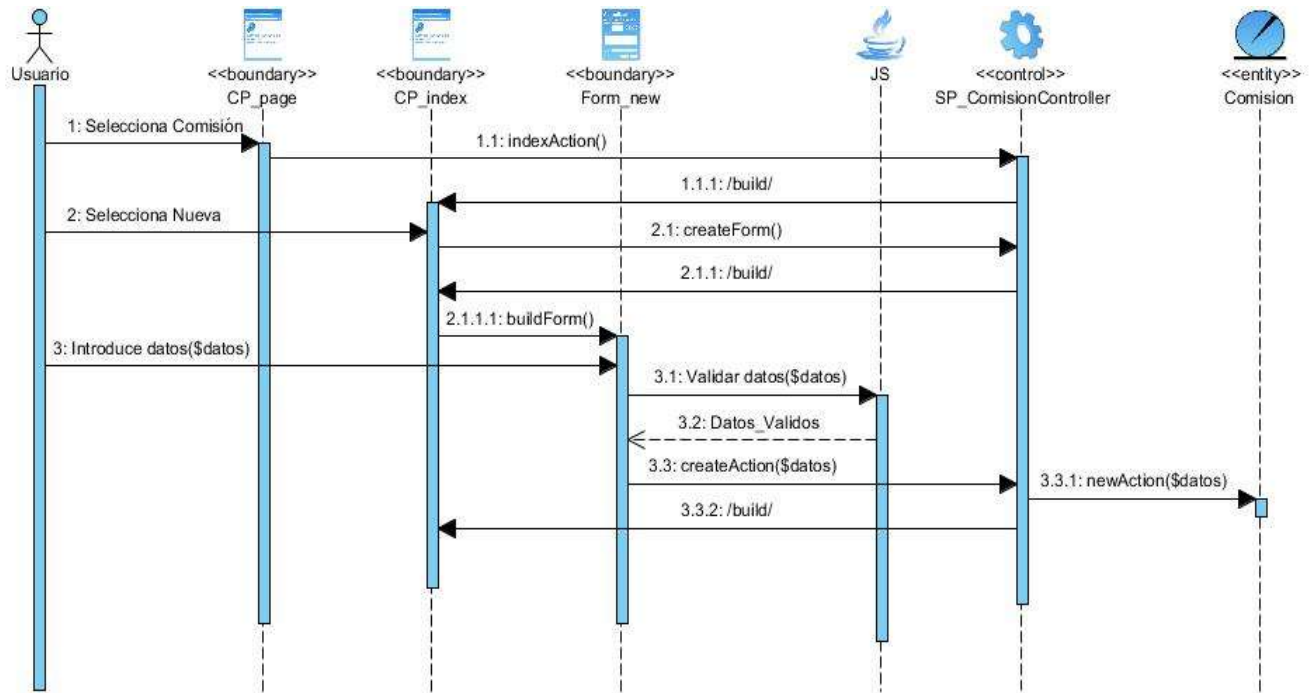


Figura 7. Diagrama de secuencia de la sección Crear comisión del caso de uso Gestionar comisión (Elaboración propia)

3.6.3 Diagrama de despliegue

El diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema, además es un Lenguaje Unificado de Modelado que se utiliza para modelar el hardware utilizado en las implementaciones del sistema y las relaciones entre sus componentes. (58)

A continuación se muestra el diagrama de despliegue del sistema.

Capítulo 3. Desarrollo de la propuesta de solución

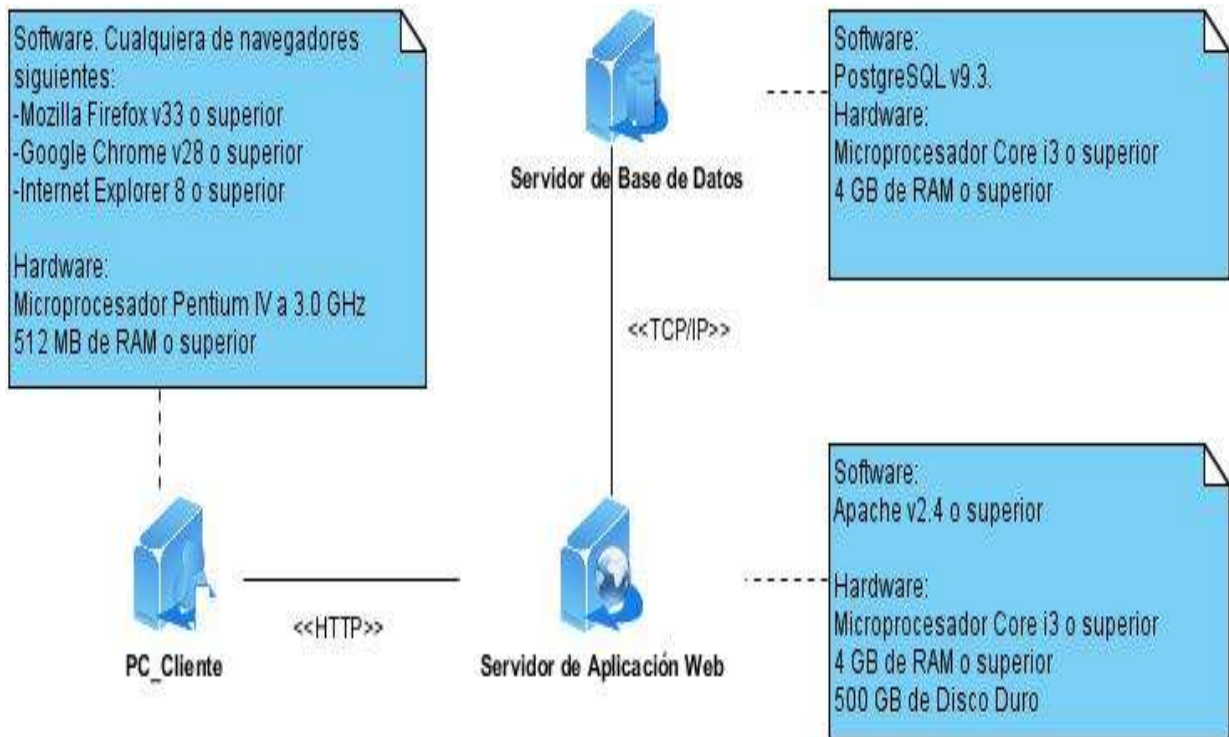


Figura 8. Diagrama de despliegue del sistema (Elaboración propia)

3.6.4 Diagrama de componentes

El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes. Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean estos componentes de código fuente, binarios o ejecutables. Además, tiene un nivel más alto de abstracción que un diagrama de clase, usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución. Estos son bloques de construcción, como eventualmente un componente puede comprender una gran porción de un sistema.(59)

A continuación, se muestra el diagrama de componentes correspondiente al caso de uso Gestionar comisión, el resto de los diagramas de este tipo se encuentran en el Anexo No 5.

Capítulo 3. Desarrollo de la propuesta de solución

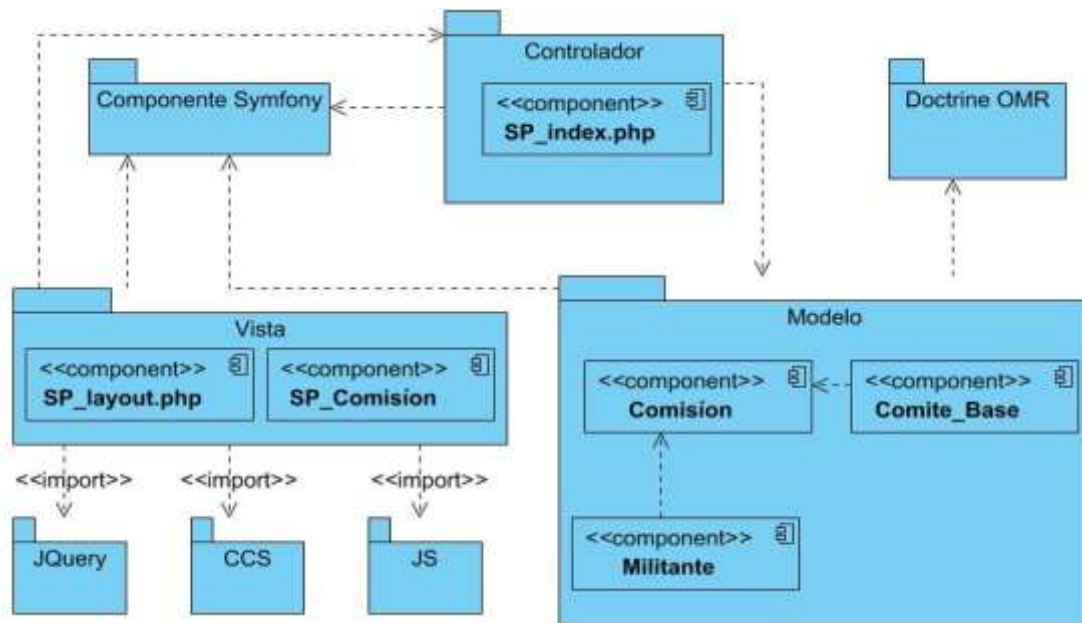


Figura 9. Diagrama de componentes del caso de uso Gestionar comisión (Elaboración propia)

3.7 Pruebas de software

Las pruebas de software son las investigaciones empíricas y técnicas cuyo fin es proporcionar información objetiva e independiente sobre la calidad del producto. Esta actividad forma parte del proceso de control de calidad global. Las pruebas son básicamente un conjunto de actividades dentro del desarrollo de software y dependiendo del tipo de pruebas, estas actividades podrán ser implementadas en cualquier momento del proceso de desarrollo.

Con las pruebas de software se persigue descubrir errores en el sistema y verificar que cumple con los requerimientos del cliente, con el fin de lograr un sistema con la calidad necesaria para su uso. Con esto no se obtiene un sistema totalmente libre de errores, pero si apto para ser usado por el usuario final.(60)

3.7.1 Niveles de prueba

En un proceso de pruebas formal, suelen confundirse con mucha facilidad, los niveles de pruebas con los tipos de prueba, y a pesar de que se encuentren íntimamente relacionadas, tienen connotaciones diferentes en el proceso. Las pruebas pueden ejecutarse en cualquier punto del proceso de desarrollo de software, y es aquí donde los niveles de prueba permiten entender con claridad los diferentes puntos o

Capítulo 3: Desarrollo de la propuesta de solución

etapas en donde pueden ejecutarse ciertos tipos de prueba. Por lo anterior, es común que algunas personas se refieran a los niveles de pruebas o intenten clasificarlos como: pruebas de desarrollador, pruebas funcionales y pruebas de usuario final. Sin embargo, la terminología apropiada para referirse a los diferentes niveles corresponde a la siguientes cuatro clasificaciones que son: pruebas unitarias, pruebas de integración, pruebas de sistema y pruebas de aceptación. En cada uno de estos niveles de prueba, se podrán ejecutar diferentes tipos de prueba tales como: pruebas funcionales, no funcionales, de arquitectura y asociadas al cambio de los productos.(61)

Se muestran a continuación algunos niveles de pruebas para valorar cuales serán utilizados:

Pruebas Unitarias o de Componente:

Este tipo de pruebas son ejecutadas normalmente por el equipo de desarrollo, básicamente consisten en la ejecución de actividades que le permitan verificar al desarrollador que los componentes unitarios están codificados bajo condiciones de robustez, esto es, soportando el ingreso de datos erróneos o inesperados y demostrando así la capacidad de tratar errores de manera controlada. Adicionalmente, las pruebas sobre componentes unitarios suelen denominarse pruebas de módulos o pruebas de clases, siendo la convención definida por el lenguaje de programación la que influye en el término a utilizar. (61)

Pruebas de Integración:

Este tipo de pruebas son ejecutadas por el equipo de desarrollo y consisten en la comprobación de que elementos del software que interactúan entre sí, funcionan de manera correcta. En este nivel se verifican por separados las piezas de software en un funcionamiento aislado, estas piezas pueden ser módulos individuales, subprogramas y componentes. Para realizarlas pueden llevarse a cabo con herramientas de depuración, acceso al código fuente y pueden participar en esta de forma opcional los programadores.(62) Se aplicarán las pruebas unitarias para poder validar las funcionalidades con las que cuenta el sistema sin la utilización de un software informático ni documentación, estas serán ejecutadas por el propio equipo de desarrollo. Se ejecutarán las pruebas de sistema una vez que esté concluido, para verificar que cumple con los requerimientos funcionales y no funcionales, dejándolo listo para su uso final. Se aplicarán pruebas de integración debido a que el sistema es un módulo que se integrará al SIGPS_UCI. Se utilizó la estrategia matriz de trazabilidad, donde se asocian o definen dependencias en las salidas de las funciones programadas y las ya existentes, verificando que no se generen errores o salidas no deseadas.

Capítulo 3: Desarrollo de la propuesta de solución

3.7.2 Métodos de prueba

Cualquier producto de ingeniería puede probarse de una de estas dos formas: (1) conociendo la función específica para la que fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa y buscando errores de cada función; (2) conociendo el funcionamiento del producto, se pueden desarrollar pruebas que aseguren que todas las piezas encajan, o sea, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada. El primer enfoque de prueba se denomina prueba de caja negra y el segundo, prueba de caja blanca. (54)

Pruebas de caja negra:

Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que produce un resultado correcto, así como la integridad de la información externa (por ejemplo, archivos de datos) se mantiene. Una prueba de caja negra examina algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software.(54)

Dentro del método de Caja Negra la técnica de la Partición de Equivalencia es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. Serán aplicadas las pruebas de caja negra sobre la interfaz del programa a desarrollar con el objetivo de medir la calidad del sistema desarrollado, estas pruebas no requieren del conocimiento de la lógica del programa y solamente se debe conocer la funcionalidad que este debe realizar.

3.7.3 Diseño de casos de prueba

El diseño de casos de prueba es una parte de las pruebas de componentes y sistemas en las que se diseñan las entradas y las salidas esperadas para probar el sistema, el objetivo que persigue es crear un conjunto de casos de prueba que sean efectivos descubriendo defectos en los programas y muestren que el sistema satisface los requerimientos. Se deben diseñar pruebas que tengan la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y tiempo posible.

Capítulo 3. Desarrollo de la propuesta de solución

A continuación se muestran los diseños de casos de prueba del caso de uso Gestionar comisión de materiales, el resto de los artefactos de este tipo se encuentran en el Anexo.

Tabla 4. Caso de prueba del caso de uso Gestionar comisión (Elaboración propia)

Escenario	Descripción	Nombre de la comisión	C/B	Pdte	Com 1	Com 2	Respuesta del sistema	Flujo central
EC1.1	Selecciona la opción comisión en el menú Diagnóstico Sociopolítico.						<p>El sistema muestra los datos de cada una de las comisiones almacenadas en el sistema.</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Crear nueva comisión. • Modificar comisión. (Ver SC 2: Modificar comisión). • Eliminar comisión. (Ver SC 3: Eliminar comisión). • Visualizar comisión. (Ver SC 4: Visualizar comisión). 	Diagnóstico Sociopolítico/ Comisión

Capítulo 3: Desarrollo de la propuesta de solución

EC 1.2	Selecciona la opción Crear Comisión						El sistema muestra un formulario con los datos requeridos para la creación de una comisión. Brinda la opción: <ul style="list-style-type: none"> • Crear • Cancelar 	Diagnóstico Sociopolítico/ Comisión/Crear comisión.
EC 1.3	Introduce datos y selecciona la opción Crear	V	V	V	V	V	Valida los datos introducidos. Registra una nueva comisión en el sistema. El sistema muestra un mensaje de éxito.	Diagnóstico Sociopolítico/ Comisión/Crear comisión/Crear
EC 1.4	Datos incompletos	I	N/A	N/A	N/A	N/A	El sistema muestra un mensaje: "Existen campos vacíos".	Diagnóstico Sociopolítico/ Comisión/Crear comisión/Crear
		V	N/A	N/A	N/A	N/A		
EC 1.5	No se ha elegido un presidente de comisión.	V	N/A	I	V	V	El sistema muestra un mensaje: "Debe elegir un Presidente de comisión"	Diagnóstico Sociopolítico/ Comisión/Crear comisión/Crear

Capítulo 3: Desarrollo de la propuesta de solución

EC 1.6	No se ha elegido el primer miembro de la comisión.	V	N/A	V	I	V	El sistema muestra un mensaje: "Debe elegir todos los miembros de la comisión."	Diagnóstico Sociopolítico/Comisión/Crear comisión/Crear
EC 1.7	No se ha elegido el segundo miembro de la comisión.	V	N/A	V	V	I	El sistema muestra un mensaje: "Debe elegir todos los miembros de la comisión."	Diagnóstico Sociopolítico/Comisión/Crear comisión/Crear
EC 1.8	Selecciona la opción Cancelar.						Cierra la ventana actual. Retorna al 1.1.	Diagnóstico Sociopolítico/Comisión/Crear comisión/Cancelar
SC 2 Modificar comisión.								
EC 2.1	Selecciona la opción Modificar comisión.						El sistema muestra una nueva página con los campos a modificar en la comisión seleccionada. Muestra las opciones: <ul style="list-style-type: none"> • Actualizar • Cancelar 	Diagnóstico Sociopolítico/Comisión/Modificar comisión

Capítulo 3. Desarrollo de la propuesta de solución

EC 2.2	Selecciona la opción Actualizar						El sistema actualiza la información de la comisión seleccionada en la lista de comisiones	Diagnóstico Sociopolítico/ Comisión/Modificar comisión/Actualizar
EC 2.3	Selecciona la opción Cancelar						Cierra el dialogo actual.	Diagnóstico Sociopolítico/ Comisión/Modificar comisión/Cancelar
SC 3 Eliminar comisión								
	Selecciona la opción Eliminar comisión						El sistema elimina la comisión seleccionada.	Diagnóstico Sociopolítico/ Comisión/Eliminar
SC 4 Mostrar comisión.								
EC 4.1	Selecciona la opción Mostrar comisión						El sistema muestra una nueva página con todos los datos de la comisión seleccionada y la opción	Diagnóstico Sociopolítico/ Comisión/Mostrar

3.7.4 Resultados obtenidos de las pruebas de caja negra

Al aplicar el método de caja negra sobre las interfaces gráficas del sistema se pudo constatar el funcionamiento de los requisitos funcionales y no funcionales del mismo.

Capítulo 3: Desarrollo de la propuesta de solución

La realización de las pruebas permitió establecer un conjunto de No Conformidades (NC) en la 1ra iteración, estas fueron agrupadas en la siguiente tabla siendo evaluadas en un rango comprendido entre: Alta, Media y Baja. A continuación se presentan los resultados arrojado por estas pruebas:

Tabla 5. Resultados de las pruebas de caja negra (Elaboración propia)

Iteraciones	Cantidad de casos de prueba	No conformidades			
		Alta	Media	Baja	Total
1	19	5	12	19	36
2	19	3	8	11	22
3	19	1	2	2	5

La tabla anterior muestra la ejecución de tres iteraciones, donde encada iteración se le da solución a las no conformidades detectadas, mejorando la calidad del software y preparándolo progresivamente para su uso final.

3.7.5 Resultados de las pruebas de integración

Al utilizar la estrategia matriz de trazabilidad para verificar si se generen errores no deseados entre las salidas de las funcionalidades programadas y las ya existentes, se obtuvo el siguiente resultado:

Tabla 6. Resultados de las pruebas de integración (Elaboración propia)

Funcionalidades	Militante	Miembro del universo juvenil	Comité de base
Evaluación del militante	RS		
Caracterización de los jóvenes no militantes		RS	
Aval del PCC	RS		
Matriz DAFO			RS

Leyenda: RS (Resultado Satisfactorio), RI (Resultado Insatisfactorio).

Capítulo 3: Desarrollo de la propuesta de solución

3.8 Validación de los resultados

Para validar los resultados obtenidos teniendo en cuenta la Hipótesis: el desarrollo de un módulo para la gestión de la información que genera el proceso del diagnóstico sociopolítico, permitirá agilizar la obtención de los indicadores que genera dicho proceso, se realizó un estudio comparando los pasos en los que se ejecutaba el proceso antes y luego de ser informatizado.

El diagnóstico sociopolítico contaba con los siguientes pasos:

1. Se creaban las comisiones de trabajo en una reunión ordinaria, posteriormente se realizaban manualmente los tres componentes del diagnóstico y luego se confeccionaban los tres informes finales.
2. Se llenaba manualmente todos los aspectos de la evaluación del militante.
3. Se hacía manualmente la confección del aval, apoyado en herramientas ofimáticas.
4. Una vez concluidos los tres componentes del diagnóstico, la comisión debía recopilar toda la información en tres plantillas que debían ser entregadas al secretario general del comité primario al cual se subordina su comité de base.
5. Para la confección del expediente de las BTJ, los jóvenes recopilaban una a una las evidencias de sus trabajos. No contaban con un modelo oficial para confeccionar el expediente. La entrega del expediente se realizaba al secretario del comité primario personalmente.
6. La confección del análisis DAFO como parte de la caracterización de la UJC se hacía manualmente, se recogían los indicadores y luego se procedía a la confección de la matriz.

Una vez realizada la propuesta de solución el proceso se realiza de la siguiente manera:

1. El secretario del comité primario crea todas las comisiones de trabajo de las organizaciones de base que se le subordinan y los militantes tienen conocimiento de la integración de la comisión.
2. Los integrantes de la comisión crean la caracterización del área, de los jóvenes y del UJC, los

Capítulo 3: Desarrollo de la propuesta de solución

cuales se añaden directamente a cada uno de los informes, con la posibilidad de exportarlos a pdf.

3. Al seleccionar la opción de evaluar a un militante, los datos de la plantilla oficial se llenan automáticamente, insertando solo los señalamientos y la propuesta de evaluación.
4. Al seleccionar la opción de avalar a un militante, los datos de la plantilla oficial se llenan automáticamente.
5. Los jóvenes insertar directamente las evidencias de los trabajos realizados e insertan los datos directamente en una plantilla oficial del expediente.
6. El militante autorizado a realizar el análisis DAFO inserta los indicadores y la matriz se genera automáticamente, calculando los pesos entre el análisis interno y externo, para de acuerdo al resultado, trazar un tipo de estrategia.

Una vez hecho este análisis se llega a la conclusión que la propuesta de solución agiliza la obtención de los indicadores que genera el proceso, tanto en tiempo como en nivel de esfuerzo.

Conclusiones Parciales

En este capítulo se desarrollaron los flujos de trabajo del diseño, implementación y las pruebas. El proceso de implementación estuvo guiado por la elaboración de los diagramas de componentes de los casos de uso más representativos, con lo que se representó la vista de implementación del sistema y se obtuvo al finalizar dicha fase el producto final con las funcionalidades requeridas.

El uso de patrones de diseño permitió aplicar buenas prácticas en el proceso de modelado e implementación del software, logrando una mayor reutilización y mantenibilidad del código. La realización del modelado del sistema permitió obtener los diagramas de clases del diseño, secuencia y despliegue, los cuales le indicaron al equipo de desarrollo los objetos que se deben construir, los atributos y comportamientos de las clases, su relación y la forma en la que interactúan las mismas desde que el usuario realiza una acción.

Conclusiones generales

La realización del módulo para el SIGPS_UCI que informatiza el proceso del diagnóstico sociopolítico constituye un aporte importante para el funcionamiento más eficiente de la UJC y sobre todo en aras de agilizar la gestión de dicho proceso, además de contribuir a que el SIGPS_UCI sea una herramienta más completa y abarcadora para el desarrollo de los procesos sustantivos de la UJC en la UCI.

La utilización de la herramienta posibilita la seguridad de los datos con los que trabajan y facilita las búsquedas de los datos en un corto tiempo y con un mínimo de errores.

Teniendo en cuenta los objetivos específicos, se puede concluir que:

- Tras el análisis de sistemas de gestión existentes con propósitos similares en el ámbito nacional e internacional se concluye no utilizar ninguno de ellos pues no cumplen con las características y necesidades de la UJC en el país, por lo que se lleva a cabo la implementación del Módulo del diagnóstico sociopolítico.
- La identificación de los procesos que conformaban el diagnóstico sociopolítico, los que dependían de él y requerían automatización arrojó resultados importantes para la determinación de los requisitos funcionales del sistema.
- La especificación de los requisitos funcionales posibilitó la definición de las funcionalidades de la aplicación informática para la gestión de información del diagnóstico sociopolítico y los procesos que dependen de él.

Recomendaciones

- ✓ Incorporar al sistema la informatización de los expedientes para optar por las medallas Jose Antonio Hecheverría y Abel Santamaría de forma automática una vez sea acordado por el comité de base.
- ✓ Desarrollar un sistema que permita la realización del proceso informatizado de forma offline, con la posibilidad de sincronizarse con el servidor central, una vez que la estación cliente sea conectada a la red.
- ✓ Implementar funcionalidades que realicen gráficos de comportamiento para una mejor visión del estado de la organización, la estabilidad de su dirección, el cumplimiento de las actividades y comparaciones anuales del comportamiento de las organizaciones de base.
- ✓ Crear un manual de usuario para la administración del sistema.

Referencias Bibliográficas

1. Partido Comunista de Cuba. [online]. [Accessed 9 February 2016]. Available from: http://www.pcc.cu/opm_ujc.php
2. INTRODUCCIÓN » Jóvenes del Comercio. [online]. [Accessed 10 February 2016]. Available from: <http://ujcmincin.cubava.cu/deberes-y-derechos/introduccion/>
3. UNIÓN DE JÓVENES COMUNISTAS. *Reglamento Unión de Jóvenes Comunistas*. 2014.
4. CAPITULO VI: ESTRUCTURA Y FUNCIONAMIENTO DE LA UJC » Jóvenes del Comercio. [online]. [Accessed 10 February 2016]. Available from: <http://ujcmincin.cubava.cu/deberes-y-derechos/capitulo-vi-estructura-y-funcionamiento-de-la-ujc/>
5. *REGLAMENTO UNIÓN DE JÓVENES COMUNISTAS*. 2014.
6. SL, Media Ingea. GestPeople (Windows). *Uptodown.com* [online]. [Accessed 7 April 2016]. Available from: <http://gestpeople.uptodown.com> Descargar GestPeople v2.0. Completo sistema de gestión de recursos humanos. GestPeople es una herramienta de gestión de recursos humanos para todo tipo de empresas. Podemos definir la estructura organizativa de nuestra empresa, definir los puestos profesionales de cada empleado, delimitar las competencias
7. GestPeople. [online]. [Accessed 7 April 2016]. Available from: <http://www.abcdatos.com/programa/recursos-humanos-gestion.html>
8. WORKMETER. WorkMeter optimiza la productividad empresarial. [online]. [Accessed 7 April 2016]. Available from: <http://es.workmeter.com/workmeter>
9. Simple HRM: Software gratuito para la administración de Recursos Humanos. [online]. [Accessed 7 April 2016]. Available from: <https://www.emprendices.co/simple-hrm-software-gratuito-para-la-administracion-de-recursos-humanos/>
10. AMADOR, Carlos Montenegro and TORRES, Yasmani Martínez. *Sistema integral para la gestión de la información de los procesos sustantivos de la UJC en la UCI*. Universidad de las Ciencias Informáticas, [no date].
11. *nota técnica erp - Los Sistemas de Informacion en la Empresa - Nota técnica - Carlos Suarez Rey - 23-03-2012.pdf* [online]. [Accessed 9 February 2016]. Available from: http://www.gcd.udc.es/subido/catedra/presentaciones/economia_competencia_ii/Los%20Sistemas%20de

- %20Informacion%20en%20la%20Empresa%20-%20Nota%20 técnica%20-%20Carlos%20Suarez%20Rey%20-%202023-03-2012.pdf
12. Lenguajes de programación, Lenguajes de programación. [online]. [Accessed 9 February 2016]. Available from: <http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>
 13. 2.2 Lenguajes de programación del lado del cliente. *Programacion Web* [online]. 14 November 2013. [Accessed 17 May 2016]. Available from: <https://programacionwebisc.wordpress.com/2-2-lenguajes-de-programacion-del-lado-del-cliente/>
 14. the basics of html · guides · WPD · WebPlatform.org. [online]. [Accessed 9 February 2016]. Available from: https://docs.webplatform.org/wiki/guides/the_basics_of_html
 15. JavaScript. *Mozilla Developer Network* [online]. [Accessed 9 February 2016]. Available from: <https://developer.mozilla.org/es/docs/Web/JavaScript>
 16. Guía Breve de CSS. [online]. [Accessed 9 February 2016]. Available from: <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>
 17. 2.3 Lenguajes de programación del lado del servidor. *Programacion Web* [online]. 14 November 2013. [Accessed 17 May 2016]. Available from: <https://programacionwebisc.wordpress.com/2-3-lenguajes-de-programacion-del-lado-del-servidor/>
 18. PHP 5 Introduction. [online]. [Accessed 9 February 2016]. Available from: http://www.w3schools.com/php/php_intro.asp
 19. ANGLADA MARTÍNEZ, Ramón Alexander and GARÓFALO HERNÁNDEZ, Alain Abel. Marco de trabajo para el desarrollo de herramientas orientadas a la gestión e integración de servicios telemáticos de infraestructura en GNU/Linux. *Revista Cubana de Ciencias Informáticas*. June 2013. Vol. 7, no. 2, p. 157–168.
 20. Qué es Symfony. [online]. [Accessed 9 February 2016]. Available from: <http://symfony.com/es/what-is-symfony>
 21. Bootstrap Get Started. [online]. [Accessed 9 February 2016]. Available from: http://www.w3schools.com/bootstrap/bootstrap_get_started.asp
 22. Home — Doctrine Project. [online]. [Accessed 9 February 2016]. Available from: <http://www.doctrine-project.org/>
 23. Qué es un servidor y cuáles son los principales tipos de servidores (proxy,dns, web,ftp,pop3 y smtp, dhcp...). [online]. [Accessed 17 May 2016]. Available from:

- http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftppop3-y-smtp-dhcp&catid=57:herramientas-informaticas&Itemid=179
24. Apache - openSUSE. [online]. [Accessed 9 February 2016]. Available from: <https://es.opensuse.org/Apache>
25. 11.2. Servidor web (HTTP). [online]. [Accessed 9 February 2016]. Available from: <https://debian-handbook.info/browse/es-ES/stable/sect.http-web-server.html>
26. Qué es un entorno de desarrollo integrado, IDE. *Programacion Desarrollo* [online]. 15 February 2011. [Accessed 9 February 2016]. Available from: <http://programaciondesarrollo.es/que-es-un-entorno-de-desarrollo-integrado-ide/>
27. NetBeans IDE - Overview. [online]. [Accessed 9 February 2016]. Available from: <https://netbeans.org/features/index.html>
28. Sistemas gestores de bases de datos. [online]. [Accessed 9 February 2016]. Available from: <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>
29. Sobre PostgreSQL | www.postgresql.org.es. [online]. [Accessed 9 February 2016]. Available from: http://www.postgresql.org.es/sobre_postgresql
30. OMG UML. [online]. [Accessed 9 February 2016]. Available from: http://www.omg.org/gettingstarted/what_is_uml.htm
31. OMG UML. [online]. [Accessed 9 February 2016]. Available from: http://www.omg.org/gettingstarted/what_is_uml.htm
32. Modelos Y Metodologías Para El Desarrollo De Software. [online]. [Accessed 9 February 2016]. Available from: <http://www.eumed.net/tesis-doctorales/2014/jlcv/software.htm>
33. MEZQUITAL, Universidad Tecnológica del Valle del. Las metodologías tradicionales y el desarrollo ágil de software. *Milenio* [online]. [Accessed 9 February 2016]. Available from: http://www.milenio.com/firmas/universidad_tecnologica_del_valle_del_mezquital/metodologias-tradicionales-desarrollo-agil-software_18_420138039.html
34. Las metodologías tradicionales y el desarrollo ágil de software - Grupo Milenio. [online]. [Accessed 9 February 2016]. Available from: http://www.milenio.com/firmas/universidad_tecnologica_del_valle_del_mezquital/metodologias-tradicionales-desarrollo-agil-software_18_420138039.html

35. Metodologías Ágiles (Introducción) | Business Intelligence, Data Warehouse, Monterrey, México : Gravitar. [online]. [Accessed 9 February 2016]. Available from: <http://gravitar.biz/bi/metodologias-agiles-intro/>
36. XP - Extreme Programming Ingeniería de Software. [online]. [Accessed 9 February 2016]. Available from: http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html XP, Programacion Extrema
37. LETELIER, Patricio and PENADÉS, M^a Carmen. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). *www.cyta.com.ar/ta0502/v5n2a1.htm* [online]. 15 April 2006. [Accessed 9 February 2016]. Available from: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>
38. AUP Ingeniería de Software. [online]. [Accessed 9 February 2016]. Available from: http://ingenieriadesoftware.mex.tl/63758_AUP.html AUP
39. METODOLOGIAS AGILES AUP. [online]. [Accessed 9 February 2016]. Available from: http://www.academia.edu/7894130/METODOLOGIAS_AGILES_AUP METODOLOGIAS AGILES AUP
40. ROGER S. PRESSMAN. *Software Engineering, a practitioners approach*. 2010. Seventh Edition.
41. Qué es SCRUM. *Proyectos Ágiles* [online]. 4 August 2008. [Accessed 10 February 2016]. Available from: <http://proyectosagiles.org/que-es-scrum/> Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto.
42. CASE. [online]. [Accessed 9 February 2016]. Available from: <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>
43. Visual Paradigm Essential. *Udemy* [online]. [Accessed 9 February 2016]. Available from: <https://www.udemy.com/visual-paradigm-essential/> Learn all essential skills of software design and modeling including, UML, BPMN and SysML with Visual Paradigm. - Curso gratuito
44. Unified Modeling Language (UML). [online]. [Accessed 26 January 2016]. Available from: <http://uml.org/>
45. Modelo de Dominio. *Tecnología y Synergix* [online]. 10 July 2008. [Accessed 17 May 2016]. Available from: <https://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>
46. VARGAS. expo tecnomas: Modelo de caso de uso del sistema. *expo tecnomas* [online]. 5 February 2010. [Accessed 17 May 2016]. Available from: <http://facilte.blogspot.com/2010/02/modelo-de-caso-de-uso-del-sistema.html>

47. Enterprise Architect - Modelo de Caso de Uso. [online]. [Accessed 17 May 2016]. Available from: http://www.sparxsystems.com.ar/resources/tutorial/use_case_model.html
48. ANÓNIMO. Patrones de Casos de Uso. *Software Guru* [online]. [Accessed 17 May 2016]. Available from: <http://sg.com.mx/content/view/510>
49. DICE, INDICE « Mundo Kramer's Blog. MODELO DE ANÁLISIS. *Mundo Kramer's Blog* [online]. 20 May 2011. [Accessed 17 May 2016]. Available from: <https://mundokramer.wordpress.com/2011/05/20/modelo-de-analisis-software/>
50. RUBENFA. Patrones de diseño: qué son y por qué debes usarlos. *Genbeta Dev* [online]. 14 July 2014. [Accessed 17 May 2016]. Available from: <http://www.genbetadev.com/metodologias-de-programacion/patrones-de-diseno-que-son-y-por-que-debes-usarlos>
51. GoF – Patrones de diseño (II): ¿Qué son? *Mundo Informático* [online]. 30 January 2013. [Accessed 17 May 2016]. Available from: <https://infow.wordpress.com/2013/01/30/gof-patrones-de-diseno-ii-que-son/>
52. JUAREZ, Maximiliano. Mi granito de java: Decorator. *Mi granito de java* [online]. 4 June 2011. [Accessed 17 May 2016]. Available from: <http://migranitodejava.blogspot.com/2011/06/decorator.html>
53. ANDRÉS GROSSO. Patrones GRASP | Prácticas de Software. [online]. [Accessed 17 May 2016]. Available from: <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>
54. PRESSMAN, Roger S. *Ingeniería de software: Un enfoque práctico 5ta edición*. 2002.
55. Lección 39. Diagrama de Clases de Diseño. [online]. [Accessed 17 May 2016]. Available from: http://datateca.unad.edu.co/contenidos/200609/exeuml/leccin_39_diagrama_de_clases_de_diseo.html
56. Análisis y Diseño con el Diagrama de Clase. [online]. [Accessed 17 May 2016]. Available from: <https://www.ibiblio.org/pub/linux/docs/LuCaS/Tutoriales/doc-modelado-sistemas-UML/multiple-html/x219.html>
57. Sparx Systems - Tutorial UML 2 - Diagrama de Secuencia. [online]. [Accessed 20 May 2016]. Available from: http://www.sparxsystems.com.ar/resources/tutorial/uml2_sequencediagram.html
58. Sparx Systems - Tutorial UML 2 - Diagrama de Despliegue. [online]. [Accessed 25 May 2016]. Available from: http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html
59. Sparx Systems - Tutorial UML 2 - Diagrama de Componentes. [online]. [Accessed 17 May 2016]. Available from: http://www.sparxsystems.com.ar/resources/tutorial/uml2_componentdiagram.html

Referencias Bibliográficas

60. PATHS, Publicado por Eleven. ElevenPaths Blog: QA: Pruebas para asegurar la calidad del producto software (I). [online]. [Accessed 17 May 2016]. Available from: <http://blog.elevenpaths.com/2014/09/qa-pruebas-para-asegurar-la-calidad-del.html>
61. PRUEBAS DE SOFTWARE. *PRUEBAS DE SOFTWARE* [online]. [Accessed 17 May 2016]. Available from: [https://pruebasdelsoftware.wordpress.com/Ingenieria de Software con énfasis en pruebas](https://pruebasdelsoftware.wordpress.com/Ingenieria%20de%20Software%20con%20%C3%A9nfasis%20en%20pruebas).
62. BUENO, Carlos Blanco. *Ingeniería de Software II Construcción y Pruebas de Software*.

Bibliografía

1. **Pressman, Roger S.** *Ingeniería del software. Un enfoque práctico.* 2002. Vol 2.
2. **Nuez, A.d.I.,** (2003) *La medición de usabilidad y experiencia de usuario en la Web.*
3. **Castro, E.,** *HTML con XHTML y CSS.* 2003.
4. **Ambyssoft Inc.** *La disciplina de pruebas - Agile UP. Ambyssoft Inc.'s Agile UP Product.* [En línea] 2006. [Citado el: 23 de 4 de 2015.] <http://www.cc.una.ac.cr/AUP/html/test.html>.
5. **Aranda, P.J.V.,** (2004) *Especificación de Requisitos en el Desarrollo de Aplicaciones Web.*
6. **Zeldman, J.,** *Diseño con estándares Web.* 2004.
7. **Laboreo, D.C.,** (2005) *Cómo hacer buenas páginas Web.*
8. **ALVAREZ, M. A.,** *Índice del manual de lenguajes del web.*
9. **Gil Martín, M.A.,** *Sistema de Control de Accesos.* Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Politécnico “José Antonio Echeverría”, Universidad de las Ciencias Informáticas 2005, 106 p.
10. **World Wide Web Consortium.** *Guía Breve de XHTML. W3C España.* [En línea] [Citado el: 2 de 06 de 2015.] <http://www.w3c.es/Divulgacion/GuiasBreves/XHTML>.
11. **Montserrat, Grupos de Carlos.** *BDA (Bases de Datos).* 2º Curso. 2º cuatrimestre. Escuela Universitaria de Informática. Facultad de Informática. : s.n., curso 2005/2006.
12. **Paré, Rafael Camps.** *Software Libre, Base de Datos.* Fundación de la Universidad Oberta de Catalunya : s.n., mayo 2005. 84-9788-269-5.
13. **Pérez, Javier Eguíluz.** *Introducción a CSS .* s.l. : Programadores web y diseñadores web sin experiencia con CSS .
14. CDT Internet. [Online] [Cited: 06 01, 2008.] <http://cdtinternet.org/libro-sobre-drupal>.
15. **Gallego, Juan Pablo Gómez.** *Fundamentos de la Metodología RUP.* Universidad Tecnológica de Pereira : s.n., 2007.
16. *SISTEMAS DE GESTIÓN DE CONTENIDOS (CMS).* s.l. : <http://www.catamarca.gov.ar/>.
17. **Naramore, Elizabet, Glass, Michael K y Scouarnec, Yann Le.** *Desarrollo Web con PHP, APACHE y MySQL.* s.l. : Ed. Anaya Multimedia.