



Facultad 2

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



**Título: Sistema de Información para la UJC de la UCI.**  
**Subsistema para la gestión del ID2 de la UJC de la UCI.**

**Autor:** Yilen Yamilé Reyes Montejo.

**Tutor:** Ing. Antonio Hernández Domínguez.

**Co-tutor (es):**

Msc. Asnioby Hernández López.

Ing. Alejandro Morales Torres.

“Año 54 de la Revolución”.

Ciudad de la Habana, Cuba. Junio del 2012.

# *Agradecimientos*



## **AGRADECIMIENTOS**

A mis padres Leticia Montejo Virelles y Ramón Reyes Montano, por todo el apoyo, cariño, comprensión y confianza depositada en mí.

A mi papá: Por ser tan importante y especial en mi vida, por enseñarme lo importante que es amar a alguien.

A mi mamá: Quien es más que una amiga para mí, es mi confidente, compañera, mi ejemplo de mujer luchadora y fuerte.

A mi hermano Yudisnel: Al cual quiero y respeto mucho, por sus consejos, cariño y apoyo.

A mi esposo José Andrés: Por enseñarme lo importante que es perseverar en la vida y luchar por las cosas que queremos, además de todo el apoyo, cariño y ternura de todos estos años.

A toda mi familia, mis abuelos: Clara y Raúl, mis tíos y tías, primos en general a todos ellos Por confiar en mí.

A mi abuela Cuca, la cual es la persona que siempre está conmigo a ella quiero agradecerle el estar hoy aquí, y ser la persona que me ha iluminado cada paso que he dado.

A la familia de mi esposo: su abuela y abuelo, sus tías y primas, mi suegra, mi cuñado, mi suegro, a todos ellos gracias por aceptarme como un miembro más de esa maravillosa familia, por creer en mí y apoyarme en todo.

A mi tutor Antonio Hernández Dominguez, al cual quiero agradécele el haber tenido la oportunidad de haber aprendido a su lado, y que más que mi tutor ha sido un compañero y amigo. Por todo su apoyo incondicional y toda la confianza depositada en mí.

A mis Co-Tutores: Alejandro Morales Torres y Asnioby Hernández López: A ellos por estar siempre disponibles para cualquier duda, pregunta, por todo lo que me enseñaron.

A Alejandro por todas esas tardes las cuales me dedico, por todo lo aprendido a su lado y principalmente por haber sido tan incondicional.



## *Agradecimientos*

A Asnioby Por haber confiado en mí para el desarrollo de este trabajo diploma.

A mis compañeras de cuarto por haber compartido tantas cosas y tantos años.

A mis amigos: Osmany Ramos, a Yenier Vega, a Marvis Delabat y Sucel Martí.

A mis profesores, en general a todos aquellos de una forma u otra contribuyeron a que hoy pudiera estar aquí.

*Resumen.*

*Subsistema para la gestión del ID2 de la UJC de la UCI.*

## **Resumen**

En la Universidad de las Ciencias Informáticas (UCI) la información relacionada con la Unión de Jóvenes Comunistas (UJC) se controla y actualiza en gran medida de forma manual, y en algunos casos se utilizan algunas herramientas informáticas que no cubren todas las necesidades de la organización. El Departamento de Documentación y Control es el encargado de manejar toda la información referente a los comités primarios, organizaciones de base, expedientes, generación de las plantillas de control de la militancia (ID2), la cotización, entre otras actividades referentes a los miembros de la UJC.

La universidad actualmente cuenta con 9 comités primarios, 364 comités de base y un total de 5637 militantes entre trabajadores y estudiantes. En dicho departamento el control de las incorporaciones y traslados realizados diariamente por los especialistas que allí laboran, es registrado y actualizado través del procesador de hojas de cálculo Microsoft Excel; por lo que mantener actualizada esta información resulta engorroso debido a la gran cantidad de militantes y los continuos movimientos que ocurren en dicha organización.

Por tal motivo el objetivo del siguiente trabajo diploma, se centra en el desarrollo de un subsistema, como parte del Sistema de Información para la UJC de la UCI (SIUJC), que provea un mecanismo para la gestión de la información del ID2 en la universidad.

En el documento se recoge la descripción del problema, se explican los conceptos relacionados, se detallan la tecnología, lenguajes y herramientas utilizadas, así como se muestra la propuesta de solución, y la validación del subsistema mediante las pruebas realizadas.

## **Palabras claves:**

Control del Estado de la Militancia (ID2), comité primario, comité de base, militantes.

## Índice

|  |           |
|--|-----------|
| <b>Introducción</b> .....  | <b>1</b>  |
| <b>Capítulo 1</b> .....  | <b>6</b>  |
| <b>Fundamentación teórica</b> .....  | <b>6</b>  |
| 1.1    Introducción .....  | 6         |
| 1.2    Gestión de la información .....   | 6         |
| 1.3    Conceptos asociados al dominio del problema .....                               | 8         |
| 1.3.1    ID2 .....   | 8         |
| 1.4    Sistemas automatizados similares existentes vinculados al campo de acción. .... | 8         |
| 1.5    Metodología, lenguaje y herramientas de desarrollo .....                        | 10        |
| 1.5.1    Metodología de desarrollo .....   | 11        |
| 1.5.2    Notación de Modelado de Procesos de Negocio BPMN .....                        | 12        |
| 1.5.3    Herramienta CASE .....  | 13        |
| 1.5.4    Gestor de Base de Datos .....   | 13        |
| 1.5.5    Herramienta de Desarrollo (IDE) .....   | 14        |
| 1.5.6    Servidor Web .....  | 15        |
| 1.5.7    Lenguajes de programación .....   | 16        |
| 1.5.8    Entorno de Desarrollo Integrado.....  | 17        |
| 1.6    Conclusiones .....  | 18        |
| <b>Capítulo 2</b> .....  | <b>19</b> |
| <b>Características del sistema</b> .....   | <b>19</b> |
| 2.1    Introducción .....  | 19        |
| 2.2    Modelo de negocio .....   | 20        |
| 2.2.1    Modelo de procesos de negocio.....  | 20        |
| 2.2.2    Descripción textual de los procesos de negocio .....                          | 21        |
| 2.2.3    Actividades presentes en el negocio .....                                     | 24        |

*Subsistema para la gestión del ID2 de la UJC de la UCI.*

|                                |  |           |
|--------------------------------|--|-----------|
| 2.3                            | Objeto de Automatización.....                              | 26        |
| 2.4                            | Propuesta del sistema .....                                | 26        |
| 2.5                            | Personas Relacionadas con el sistema .....                 | 27        |
| 2.6                            | Requisitos no funcionales del sistema .....                | 28        |
| 2.7                            | Fase de Exploración .....                                  | 30        |
| 2.7.1                          | Historia de Usuario.....                                   | 30        |
| 2.8                            | Planificación .....  | 32        |
| 2.8.1                          | Estimación de esfuerzos por historias de usuario.....      | 33        |
| 2.9                            | Iteraciones.....   | 33        |
| 2.9.1                          | Iteración 1 .....  | 34        |
| 2.9.2                          | Iteración 2 .....  | 34        |
| 2.9.3                          | Iteración 3 .....  | 34        |
| 2.10                           | Plan de entrega .....                                      | 35        |
| 2.11                           | Conclusiones .....   | 36        |
| <b>Capítulo 3</b>              | .....  | <b>37</b> |
| <b>Diseño del sistema</b>      | .....  | <b>37</b> |
| 3.1                            | Introducción.....  | 37        |
| 3.2                            | Arquitectura .....   | 37        |
| 3.2.1                          | Patrón de Arquitectura .....                               | 37        |
| 3.3                            | Patrones de diseño.....                                    | 39        |
| 3.3.1                          | Patrones GOF (Gang of Four, Banda de Cuatro) .....         | 39        |
| 3.3.2                          | Patrones para Asignar Responsabilidades (GRASP) .....      | 39        |
| 3.4                            | Tarjetas Clase – Responsabilidad – Colaborador (CRC) ..... | 40        |
| 3.5                            | Modelo de Datos.....                                       | 41        |
| 3.6                            | Conclusiones .....   | 42        |
| <b>Capítulo 4</b>              | .....  | <b>43</b> |
| <b>Implementación y Prueba</b> | .....  | <b>43</b> |
| 4.1                            | Introducción.....  | 43        |

## Índice.

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

|       |  |           |
|-------|--|-----------|
| 4.2   | Estándares .....                                     | 43        |
| 4.2.1 | Estándar de codificación para la base de datos ..... | 43        |
| 4.2.2 | Estándar de codificación para PHP .....              | 44        |
| 4.3   | Fase de Implementación.....                          | 50        |
| 4.3.1 | Iteración 1. ....                                    | 54        |
| 4.3.2 | Iteración 2 .....                                    | 56        |
| 4.3.3 | Iteración 3 .....                                    | 58        |
| 4.4   | Pruebas .....  | 59        |
| 4.4.1 | Pruebas unitarias .....                              | 60        |
| 4.4.2 | Pruebas de integración .....                         | 66        |
| 4.5   | Conclusiones .....                                   | 69        |
|       | <b>Conclusiones Generales .....</b>                  | <b>70</b> |
|       | <b>Recomendaciones .....</b>                         | <b>71</b> |
|       | <b>Bibliografía .....</b>                            | <b>72</b> |

## *Introducción*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

#### **Introducción**

Desde la antigüedad la gestión de la información es de gran importancia para el conocimiento y la administración de las diferentes empresas y organizaciones. Con el auge de la informatización de las tecnologías, surgen los sistemas de gestión de la información utilizados en diferentes áreas, organismos y empresas; convirtiéndose así en un elemento fundamental para el control y la toma de decisiones sobre las actividades que se desarrollan en estas entidades.

En Cuba existen numerosas empresas y organizaciones, las cuales generan producto de sus actividades un conjunto de información, que van representando el actuar diario e histórico de las mismas. Una de las organizaciones presentes dentro del país es la Unión de Jóvenes Comunistas (UJC), la cual constituye la organización juvenil del Partido Comunista de Cuba (PCC), forjada en las concepciones marxista-leninista, heredera y continuadora de las tradiciones de lucha de la juventud cubana, que a lo largo de la historia ha estado en el destacamento de vanguardia revolucionaria (1).

En el año 2002 se crea en el país la Universidad de las Ciencias Informáticas (UCI), con el objetivo de formar profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática. Producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo, como modelo de formación. Servir de soporte a la industria cubana de la software (2).

La universidad como comunidad existente dentro del país está constituida por diferentes organizaciones políticas y de masas. Una de ellas, la UJC. En la actualidad, gran parte del flujo de trabajo de la UJC en la UCI es realizado de forma manual y algunos procesos son gestionados con herramientas informáticas que no cubren todas las necesidades de dicha organización, lo que ha generado trabas en el correcto funcionamiento de la misma.

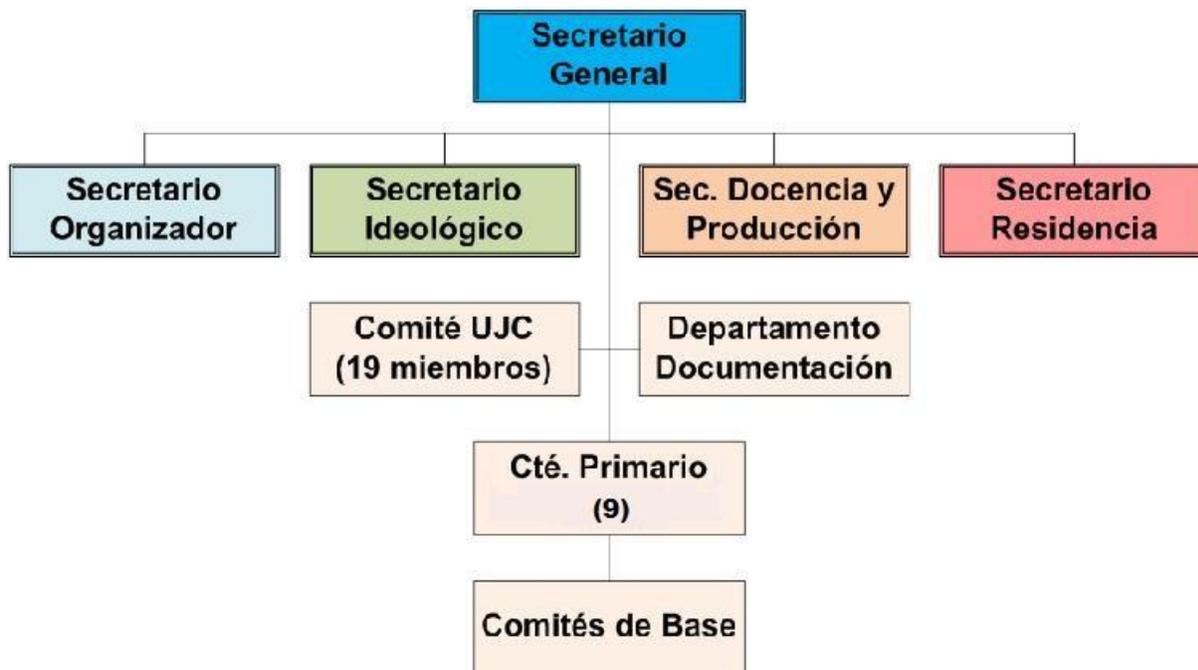
En aras de resolver los problemas existentes, la universidad en conjunto con la dirección de esta organización, realizó un acuerdo para desarrollar el Sistema de Información de la UJC (SIUJC). Este proporciona mejoras en cuanto a la gestión de los datos y el empleo de un menor tiempo a la hora de realizar consultas sobre la información. Dentro de las funcionalidades con

## Introducción

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

las que el sistema debe cumplir, se encuentra un módulo que informatice el proceso de elaboración de actas y opiniones para las reuniones que se efectúan en las organizaciones de base. Estas reuniones se efectúan mensualmente por los militantes, donde las opiniones realizadas por estos son recogidas en un acta. El SIUJC carece de un mecanismo para gestionar la información referente a los militantes, así como las incorporaciones y traslados realizados por los mismos, que son registrados en la Plantilla de Control del Estado de la Militancia (ID2).

La UJC en la UCI se encuentra dividida en diferentes departamentos como se aprecia en la figura (Ver Figura. 1).



**Figura 1: Estructura de la UJC.**

Uno de los departamentos de la UJC es el Departamento de Documentación y Control. Los documentadores tienen la misión de mantener actualizada toda la información referente al estado de la militancia, la actualización de las plantillas, los expedientes de los miembros de la organización, los movimientos realizados por estos y la generación del ID2. Debido a esto, en el departamento se genera gran cúmulo de información por el desarrollo continuo de las tareas, el control de las actividades y registros internos; por lo que la gestión de la información es de gran importancia para su funcionamiento. Los procesos de gestión de la información en el

## *Introducción*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

departamento, se realizan de manera manual y en el mejor de los casos usando procesadores de texto como Microsoft Word y hojas de cálculo como Microsoft Excel. Constituyendo sin duda alguna, una limitación en varios aspectos como es el caso de la disponibilidad de la información, el no aprovechamiento de otras fuentes de datos y la agilización de los procesos.

Otra actividad que es llevada a cabo por los documentadores, es la actualización del ID2. Esta es ejecutada de manera mensual, centralizada y tiene vital importancia en el funcionamiento interno de la organización.

Lo anteriormente mencionado, implica que, a corto plazo exista una aglomeración de archivos en formato digital, debido al conjunto de información que se genera en dicho departamento. Además se suman los problemas que implica el almacenamiento de la información, la cual es guardada en una computadora destinada para ello y organizada mediante el uso de diferentes carpetas; provocando demoras a la hora de realizar simples consultas de la información gestionada.

A partir de la situación problémica descrita anteriormente se propone como **problema a resolver** de la siguiente investigación: ¿Cómo proveer un mecanismo al SIUJC para la gestión de la información del ID2 de la UJC en la UCI?

Con el fin de solucionar el problema, se define como **objeto de estudio** de la presente investigación: Los procesos de gestión de información de la UJC. Enfocando el **campo de acción en**: El proceso de gestión de información del ID2 en la UCI.

Teniendo en cuenta el problema a resolver se define como el **Objetivo General**: Desarrollar un subsistema para la gestión de información del ID2 para la UJC de la UCI, que se integre con el SIUJC de la UCI.

Complementando el objetivo general se definen los siguientes **Objetivos Específicos**:

- ✓ Diseñar el Subsistema de Gestión de Información del ID2 como parte del sistema SIGUJC.
- ✓ Implementar la solución diseñada.
- ✓ Realizar pruebas al subsistema implementado.

## *Introducción*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

Una vez definido el objetivo general y en aras de darle cumplimiento, se plantea la siguiente **idea a defender**: Con el desarrollo del Subsistema para la gestión del ID2 de la UJC de la UCI se mantendrá una mejor organización de los datos y precisión a la hora de su búsqueda, así como el empleo de un menor tiempo al realizar consultas a la información y las tareas de la organización como brindar un mecanismo que permita la gestión de las actas y opiniones al SIUJC.

Para lograr el cumplimiento de los objetivos se proponen las siguientes **tareas a realizar**:

- ✓ Caracterización de los sistemas informáticos que existen en Cuba para la gestión de la información de la UJC
- ✓ Realización de un estudio de los procesos de dar altas y bajas a un militante de la UJC.
- ✓ Realización de un estudio de cómo se genera el modelo del cómputo o cierre del ID2.
- ✓ Realización de un estudio y caracterizar las metodologías, herramientas y tecnologías a utilizar en el desarrollo del subsistema.
- ✓ Realización de un estudio sobre la integración de reportadores dinámicos.
- ✓ Definición de la arquitectura y prototipos de usuario.
- ✓ Realización de un análisis de cómo acceder a los servicios web existentes en la universidad.

Para apoyar el desarrollo de la investigación se emplea el siguiente método científico: **Métodos Teóricos**:

- ✓ **Analítico Sintético**: Mediante este método se analizan los documentos y las teorías referentes a la gestión del ID2 y de los procesos existentes relacionados con este, proporcionando así la extracción de los datos más importantes.
- ✓ **Modelación**: Se utiliza para la modelación del sistema propuesto. Para convertir los procesos en diagramas, modelos y artefactos que se puedan utilizar en el diseño.
- ✓ **Histórico-Lógico**: Empleando este método se realiza un estudio relacionado con los sistemas de gestión del ID2, para así obtener un conocimiento histórico de su desarrollo y su comportamiento nivel nacional.

El trabajo de diploma se encuentra estructurado en 4 capítulos donde se describe todo el proceso de investigación y desarrollo:

## *Introducción*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

**Capítulo1. Fundamentación Teórica:** Abarca los conceptos asociados al tema, necesarios para la comprensión de la solución del problema planteado. Se realiza un estudio del estado del arte acerca de los sistemas de gestión de la información en la UJC a nivel nacional y en la universidad. Se define la metodología y las herramientas que se utilizan para el desarrollo del sistema.

**Capítulo2. Características del sistema:** Se hace énfasis en una propuesta de solución a partir de un análisis crítico de la situación actual y de los flujos de trabajo involucrados en el campo de acción y se describe una propuesta de solución.

**Capítulo3. Diseño del sistema:** Se define la arquitectura del sistema y con ello los patrones arquitectónicos empleados. Contiene también el modelo de datos y las tarjetas Clase – Responsabilidad – Colaborador.

**Capítulo4. Implementación y prueba:** Se detallan los elementos que se tuvieron en cuenta para la implementación del sistema. Incluye además el Plan de Release con la planificación y duración de desarrollo de cada Historia de Usuario. Así como de las Tareas de Ingeniería y la estrategia de pruebas a seguir para comprobar las funcionalidades del sistema.

## *Capítulo 1: Fundamentación Teórica.*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

## **Capítulo 1**

### **Fundamentación teórica**

#### **1.1 Introducción**

En el capítulo se realiza un estudio de diferentes sistemas de gestión de la información para la UJC existentes en el país y en la Universidad de las Ciencias Informáticas. Además, se realiza la fundamentación del uso de las herramientas y lenguajes de programación que serán utilizados para el desarrollo del sistema.

Para un mayor entendimiento del problema planteado se hace necesario explicar algunos términos relacionados con la gestión de la información y con el ID2 de la UJC.

#### **1.2 Gestión de la información**

La gestión de la información es el proceso que incluye operaciones como extracción, manipulación, tratamiento, depuración, conservación, acceso y/o colaboración de la información adquirida por una organización a través de diferentes fuentes y que gestiona el acceso y los derechos de los usuarios sobre la misma (3).

**Extracción:** Es el proceso de extraer los datos de diferentes fuentes, con fines o no estadísticos.

**Manipulación:** Son aquellas operaciones que se realizan con los datos como: insertar, modificar, buscar, eliminar, entre otras operaciones.

**Tratamiento:** El tratamiento de los datos es el proceso que incluye operaciones y técnicas automatizadas o no como: recogida, grabación, conservación, elaboración, modificación, bloqueo y cancelación, consultas, interconexiones y transferencias (4).

**Depuración:** Es el proceso que realiza la limpieza de los datos eliminando aquellos que no le son útiles.

## *Capítulo 1: Fundamentación Teórica.*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

**Conservación:** Es el proceso que se encarga de mantener, cuidar y almacenar la información para que no sea alterada.

**Acceso:** Se encarga de permitir a los usuarios realizar acciones sobre la información según los niveles de accesibilidad definidos para estos.

**Colaboración:** Es el proceso que involucra el trabajo de varias personas en conjunto para conseguir una solución.

Como objetivos esenciales de un proceso de gestión se pueden destacar los siguientes aspectos:

**Planificar:** Es el proceso de establecer objetivos con el fin de obtener determinados resultados, así como, identificar las acciones necesarias para alcanzarlos. Dentro de este concepto se contemplan un conjunto de decisiones o una selección de alternativas para el logro de tales resultados. Esta función se registra en diversos tipos de documentos: planes, programas, pronósticos y políticas.

**Organizar:** Es el proceso de dividir el trabajo a realizar y de coordinar el logro de resultados que tienen un propósito común. Organizar es la química de la organización donde se mezclan todos los elementos que interactúan entre sí a fin de obtener los resultados esperados. Es el acto de combinar habilidades, posibilidades técnicas, experiencias, recursos y todos los elementos que podrían convertirse en resultados.

**Dirigir:** Es el proceso de conducir y coordinar los esfuerzos laborales de las personas que integran una organización, ayudándolos a desarrollar tareas relevantes dentro de ella. La dirección es la función mediante la cual se ponen en marcha las actividades programadas. Comprende el compromiso de alcanzar un objetivo mediante el liderazgo de un grupo. La dirección ejerce una influencia notable en las personas para que trabajen voluntaria y entusiastamente, para el logro de las metas colectivas de equipos y de la organización en su conjunto.

**Controlar:** Es el proceso de supervisar las actividades y resultados, comparándolos con los objetivos y tomando las acciones correctivas, si son necesarias. Para ello se compara el desempeño con metas y planes, se muestran las desviaciones y al emprender medidas para corregir las desviaciones, se ayuda a

## *Capítulo 1: Fundamentación Teórica.*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

asegurar el logro de los planes. Esta función comprende el establecimiento de normas de desempeño como base para la medida de los resultados, investigación, análisis, diseño, implantación y operación de los sistemas de información, registros contables y estadísticos, auditorías, inspecciones, controles y otros métodos de verificación directa.

Entre los principales beneficios que proporcionan los sistemas de gestión de información se encuentran:

- ✓ Mayor velocidad de procesamiento de la información y obtención de resultados.
- ✓ Consultas de la información.
- ✓ Productividad del trabajo.
- ✓ Exactitud y consistencia de los datos.
- ✓ Interrelación de las áreas con el sistema.
- ✓ Seguridad de la información (5).

### **1.3 Conceptos asociados al dominio del problema**

#### **1.3.1 ID2**

Registro de los militantes de la UJC. Proceso que rige los pasos por lo que ha transitado un militante, lleva el control de las altas y bajas producidas en los comités de base.

Plantilla utilizada en la UJC que recoge los movimientos realizados en un comité de base o comité primario debido a las continuas incorporaciones o bajas, por concepto de traslados que se producen en los mismos (Ver anexo 1).

#### **1.4 Sistemas automatizados similares existentes vinculados al campo de acción.**

La UJC es una organización creada en Cuba, por lo que no existen precedentes de sistemas automatizados relacionados con esta organización a nivel internacional. A nivel nacional existen diferentes sistemas que pudieran constituir antecedentes de esta investigación, como son mayormente sitios web de algunas universidades y joven club.

## *Capítulo 1: Fundamentación Teórica.*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

#### **En Cuba.**

La organización de la UJC a nivel nacional cuenta con un sistema llamado Sistema Integral de la UJC Nacional, este cuenta con un módulo para la gestión de Cuadros, el mismo permite gestionar toda la información referente a los directivos de la organización. El sistema cuenta también con un módulo para la gestión de los militantes el cual gestiona toda la información referente a los mismos y su universo juvenil. Ambos módulos, no gestionan la información de los movimientos realizados por los militantes de la organización, ni realiza reportes de ID2 o movimientos internos o externos entre comités primarios o de base. Estos sistemas tienen como deficiencias que: No consumen de ningún servicio web presente en la universidad, no se integran al SIUJC, la entrada de los datos es totalmente manual en todo momento y ambos módulos son aplicaciones de escritorio desarrollados en Java.

El sistema de integral para la UJC Nacional cuenta además con un módulo para la realización de reportes, entre los reportes que realiza se encuentran el de control de la militancia a nivel de organización de base y referentes a los militantes, universo juvenil, entre otros. Este sistema no genera reportes relacionados con el ID2 a nivel de centro, comités primarios o sobre el expediente del militante, no consume servicios web para la obtención de datos y no se integra al SIUJC.

Otro sistema encontrado es el portal provincial de la UJC en Matanzas al cual se dedica a informar y tener noticias actualizadas referentes a la organización. Se encuentra también el sistema de información geográfica para la UJC Nacional este maneja la información referentes a los locales destinados a la recreación en la etapa veraniega, permitiendo representar los distintos locales de los organismos e instituciones del país, destinados para la recreación. Además de realizar análisis estadísticos sobre la información referente a dichos locales. Estos sistemas presenta las deficiencias de que: no generan reportes del ID2, no gestionan la información referente a los militantes, ni a los movimientos realizados por estos.

#### **En la universidad**

La UCI cuenta con un portal web llamado "Intranet" el cuál posee un espacio para las organizaciones, entre ellas para la UJC. En el sitio se encuentran publicados los reglamentos, estatutos, misión de la UJC, entre otras informaciones de interés, Este espacio se encuentra disponible para todos los estudiantes y

## *Capítulo 1: Fundamentación Teórica.*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

trabajadores de dicha universidad. Dicho sistema no brinda la posibilidad de gestionar información relativa a los procesos relacionados al ID2, ni referentes a los movimientos realizados por los militantes.

Se cuenta también con el Sistema Informático para la Gestión de la Información de la Unión de Jóvenes Comunistas de la Facultad 2, este sistema permite gestionar los militantes y profesores de la facultad 2, genera reporte de ID2 para los comités de base compuestos por estos militantes, no gestionando la información de los militantes que no son de la facultad, ni trabajadores que pertenezcan al comité primario de la facultad 2 y no permite la entrada de datos para otros comités de base que no sean miembros de este comité primario.

La organización cuenta además con un módulo de reportes para la UJC que permite generar reportes acerca de esta organización, ninguno de los mismos es referente al ID2 o a los movimientos realizados por los militantes. Otro sistema estudiado es el CAINOS el cual gestiona la información referente a los militantes y los movimientos realizados por estos. CAINOS presenta las siguientes características: Es una aplicación desktop desarrollada en java, las herramientas que utiliza deben de ser instaladas en las estaciones de trabajo de los documentadores, limitando a otros usuarios que no tengan acceso a estas computadoras, además de que no consume ningún servicio web presente en la universidad, por lo que la entrada de los datos es totalmente manual.

Teniendo en cuenta que los sistemas estudiados no se adaptan a las necesidades presentadas por el Departamento de Documentación y Control en el desarrollo de las actividades que allí se realizan, se propone el desarrollo de un subsistema, que permita la gestión del ID2, los comités de primarios, los militantes, y que brinde diversos reportes de interés para el departamento.

#### **1.5 Metodología, lenguaje y herramientas de desarrollo**

Con el objetivo de desarrollar un sistema que cumpla con los requerimientos necesarios es importante que se realice una adecuada selección de la metodología, lenguaje y herramientas de desarrollo a utilizar. En el presente epígrafe se hará referencia a las principales características y facilidades que brinda cada uno de los elementos escogidos. Es válido aclarar que estas herramientas fueron elegidas por el equipo de desarrollo que implementa el Sistema de Información para la UJC; al cual se le integrará el Subsistema para la gestión del ID2 de la UJC de la UCI.

## *Capítulo 1: Fundamentación Teórica.*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

#### **1.5.1 Metodología de desarrollo**

Uno de los principales objetivos del Proceso de Desarrollo de Software es lograr un producto eficiente y eficaz que cumpla con las necesidades del cliente. La presente investigación utiliza como metodología de desarrollo Extreme Programming (XP) que es la metodología ágil de desarrollo de software más exitosa en la actualidad. (6)

##### **1.5.1.1 Extreme Programming (XP)**

Define una serie de pautas para desarrollar software, colocando al cliente en el centro del proceso de desarrollo, en estrecha colaboración con el mismo. Se utiliza para proyectos de corto plazo, de equipos pequeños y poco tiempo de entrega, desarrollándose en iteraciones. Su ciclo de vida consta de seis fases:

- ✓ Exploración.
- ✓ Planificación de la entrega.
- ✓ Iteraciones.
- ✓ Producción
- ✓ Mantenimiento.
- ✓ Muerte del proyecto.

En el trabajo diploma solo se utilizarán 3 fases del ciclo de vida de XP.

**Exploración:** En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo.

**Planificación de la Entrega (Release):** En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta

## *Capítulo 1: Fundamentación Teórica.*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

fase dura unos pocos días. Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto.

**Iteraciones:** Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción. Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores (7).

A continuación se muestran las siguientes razones que conllevaron a su selección:

- ✓ Es un proyecto pequeño donde todo el trabajo se realiza por parejas.
- ✓ Los requisitos tienden a cambiar frecuentemente, según se avance en el trabajo, el cliente puede agregar nuevas Historias de Usuarios (HU), dividirlos o simplemente eliminarlos.
- ✓ El cliente forma parte del equipo de desarrollo.
- ✓ El programador y el cliente trabajan juntos a lo largo del proyecto.

#### **1.5.2 Notación de Modelado de Procesos de Negocio BPMN**

La notación de modelado se utiliza para modelar los procesos del negocio que están presentes en el Departamento de Documentación y Control mediante la herramienta case seleccionada (Ver epígrafe 1.4.3).

BPMN proporciona un lenguaje común para que las partes involucradas puedan comunicar los procesos de forma clara, completa y eficiente. Es sencilla de utilizar y fácil de aprender ya que los procesos tienen un bajo grado de complejidad dado a que está dirigida a modelar solo los procesos de negocio. Es un

## *Capítulo 1: Fundamentación Teórica.*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

estándar internacional de modelado de procesos independiente a cualquier metodología de modelado de procesos (8). Permite el modelamiento de los procesos de forma unificada y estandarizada alcanzando un mejor entendimiento de los procesos, crea además un puente estandarizado para minimizar el camino entre los procesos de negocio y la implementación de estos. Presenta extensiones para UML que ayudan a modelar la situación actual de los procesos de negocio del cliente.

#### **1.5.3 Herramienta CASE**

Herramientas CASE: Son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un Software. (9)

**Visual Paradigm 8.0** Se utilizada para el modelamiento de procesos de negocio del subsistema. ES una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue (10). Propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, la generación del código fuente y documentación. Permite modelar los diagramas de clases, realizar ingeniería inversa, generar código desde diagramas y documentación.

- ✓ Se utilizó en el proceso de desarrollo del software la herramienta Visual Paradigm, además de todas las características planteadas anteriormente, por ser fácil de usar, tiene un ambiente agradable para el usuario y es multiplataforma.
- ✓ Además se utilizó para la modelación de los procesos de negocio, el modelo de datos y la generación del código para la base de datos desde el modelo de datos.

#### **1.5.4 Gestor de Base de Datos**

Un Sistema Gestor de Base de Datos, (SGBD), es un software que permite crear y mantener una base de datos. Asegurando su integridad, confidencialidad, seguridad, privacidad y posteriormente acceder a los datos de forma rápida y estructurada. Un SGBD provee facilidades a la hora de manipular grandes volúmenes de datos, proporciona interfaces y lenguajes de consulta que simplifican la recuperación de los datos. Cualquier operación que un cliente haga hacia la base de datos está controlada por el gestor. (11)

## *Capítulo 1: Fundamentación Teórica.*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

Una base de datos es un almacén que permite guardar grandes cantidades de datos pertenecientes a un mismo contexto de forma organizada con el fin de consultar y utilizar fácilmente estos datos. (12)

#### **1.5.4.1 PostgreSQL 9.1**

PostgreSQL es un SGBD, su código fuente se encuentra disponible libremente. Proporciona una serie de características positivas con respecto a otros SGBD, tiene gran escalabilidad siendo capaz de soportar mayor cantidad de peticiones simultáneas de manera correcta.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (13).

Algunas de las características que conllevaron a su uso son:

- ✓ PostgreSQL funciona bien con grandes volúmenes de datos.
- ✓ Alta concurrencia de usuarios accediendo al sistema.
- ✓ Es un sistema multiplataforma.
- ✓ Incorpora funciones para el manejo de fechas.
- ✓ Incluye herencia entre tablas (aunque no entre objetos).
- ✓ Permite la declaración de funciones propias, así como la definición de disparadores.
- ✓ Presenta una amplia funcionalidad a través de su sistema de activación de disparadores.

#### **1.5.5 Herramienta de Desarrollo (IDE)**

Un IDE (Entorno de desarrollo integrado), es un paquete de software diseñado para la creación y ejecución de un programa. El mismo está compuesto por un conjunto de herramientas que han sido empaquetadas (editor de código, compilador, depurador y un constructor de interfaz gráfica); es un programa informático compuesto por un conjunto de herramientas de programación. (14)

## *Capítulo 1: Fundamentación Teórica.*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

#### **1.5.5.1 NetBeans**

Para la implementación del sistema se decidió el uso de NetBeans 7.0.1 debido a que ofrece un excelente entorno para programar en PHP.

El IDE NetBeans es un entorno de desarrollo integrado disponible para Windows, Linux, entre otros. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicaciones que permiten a los desarrolladores crear rápidamente web, escritorio y aplicaciones móviles utilizando la plataforma Java, así como PHP.

NetBeans es un producto de código abierto, con todos los beneficios del software disponible en forma gratuita (15).

Proporciona ventajas como:

- ✓ La facilidad de uso durante todo el ciclo de desarrollo.
- ✓ Soporte para crear interfaces gráficas de forma visual.
- ✓ Es un IDE multiplataforma.
- ✓ Desarrollo de aplicaciones web para lenguajes de modelado, lo que aumenta la productividad.
- ✓ La construcción del software es modular, es decir módulo sobre módulo.
- ✓ Permite el completamiento de código, facilitándole el trabajo a los programadores.
- ✓ Es una herramienta fácil de instalar y consume pocos recursos.
- ✓ El editor de código fuente que presenta es muy ágil y a la vez robusto.
- ✓ Permite crear aplicaciones Web con PHP 5.1
- ✓ Ofrece documentación y recursos de formación exhaustivos.

#### **1.5.6 Servidor Web**

Un servidor es una computadora remota que provee los datos solicitados por parte de los navegadores de otras computadoras. En redes locales se entiende como el software que configura una computadora como servidor para facilitar el acceso a la red y sus recursos. Los servidores almacenan información en forma de páginas web, FTP y a través del protocolo HTTP lo entregan a petición de los clientes (navegadores web) en formato HTML. Es un ordenador cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos (16).

## *Capítulo 1: Fundamentación Teórica.*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

#### **1.5.6.1 Apache**

Apache, es un servidor web http<sup>1</sup> de código abierto para plataformas Unix, Windows y otras, que implementa el protocolo http. Apache es un servidor flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos, es el servidor HTTP más usado. Posee características que propiciaron su uso:

- ✓ Multiplataforma.
- ✓ Modular: puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona.
- ✓ Se desarrolla de forma abierta.
- ✓ Extensible: Gracias a ser modular se han desarrollado diversas extensiones entre las que se destaca PHP (17).

#### **1.5.7 Lenguajes de programación**

Un lenguaje de programación es una notación para comunicarle a una computadora lo que deseamos que haga (18), siendo utilizado para controlar el comportamiento de un ordenador. Está constituido por un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura. El significado de sus elementos y expresiones se pone a disposición del programador para que éste pueda comunicarse con los dispositivos de hardware y software existentes (18).

##### **1.5.7.1 PHP 5.1**

PHP es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. PHP se escribe dentro del código HTML, lo que lo hace realmente fácil de utilizar. Existe un módulo de PHP para los servidores web más utilizados. Esto hace que cualquier sistema pueda ser compatible con el lenguaje significando a su vez una ventaja importante, ya que permite portar un sitio desarrollado en PHP de un sistema a otro sin prácticamente

---

<sup>1</sup> "Protocolo de Transferencia de Hipertexto"

## Capítulo 1: Fundamentación Teórica.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

ningún trabajo. Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones web gracias a la extensa librería de funciones con la que está dotado. Posee gran compatibilidad con las bases de datos más comunes (19). A continuación se mencionan algunas características que conllevaron a su uso:

- ✓ Está orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Permite aplicar técnicas de programación orientada a objetos.

#### 1.5.8 Entorno de Desarrollo Integrado

##### 1.5.8.1 Framework de Desarrollo

Framework es un concepto genérico, se refiere al ambiente de trabajo, y ejecución (20). En general los frameworks son soluciones que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución) (20).

##### 1.5.8.1.1 Yii

Yii es un framework PHP basado en componentes de alto rendimiento para desarrollar aplicaciones Web de gran escala. Permite la reutilización en la programación web y puede acelerar el proceso de desarrollo. Es totalmente orientado a la programación orientada a objetos (OOP). Implementa el patrón de diseño Modelo, Vista, Controlador (MVC) (21). Otras características que conllevaron a su uso son:

- ✓ Yii sólo carga las características que el cliente necesita.
- ✓ Posee un soporte de almacenamiento en caché de gran alcance.
- ✓ Trabaja de manera eficiente con AJAX.
- ✓ Yii incluye la validación de entrada, filtro de salida, la inyección de SQL y Cross-site, secuencias de comando de seguridad y prevención.
- ✓ Yii desarrolla un código limpio y reutilizable.
- ✓ Sigue el patrón MVC, lo que garantiza una clara separación de la lógica y la presentación.
- ✓ Presenta ayuda, ejemplos y tutorial en línea (21).

## *Capítulo 1: Fundamentación Teórica.*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

- ✓ Es uno de los framework más utilizado actualmente en el mundo, basado en la implementación MVC (22).

#### **1.6 Conclusiones**

En este capítulo se plantearon conceptos y definiciones relacionados con la investigación y el desarrollo del subsistema. Se estudiaron sistemas relacionados con el campo de acción a nivel nacional y dentro de la universidad. Además de realizar un análisis de las herramientas, metodología y lenguaje a utilizar para el desarrollo del subsistema.

## *Capítulo 2: Características del sistema.*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

## **Capítulo 2**

### **Características del sistema**

#### **2.1 Introducción**

En el presente capítulo se realiza un estudio del problema dado, una caracterización del sistema a desarrollar y se presenta una propuesta de solución. Contiene además un análisis del negocio funcional actual y de los requerimientos esperados para el desarrollo del sistema. Se confeccionan las HU, las cuales proporcionan un mayor entendimiento de los procesos. De igual forma se definen los requisitos no funcionales del sistema. Concluyendo con la estimación de las HU y la especificación de las iteraciones definiendo las etapas para el desarrollo de la aplicación.

Capítulo 2: Características del sistema.

Subsistema para la gestión del ID2 de la UJC de la UCI.

2.2 Modelo de negocio

2.2.1 Modelo de procesos de negocio

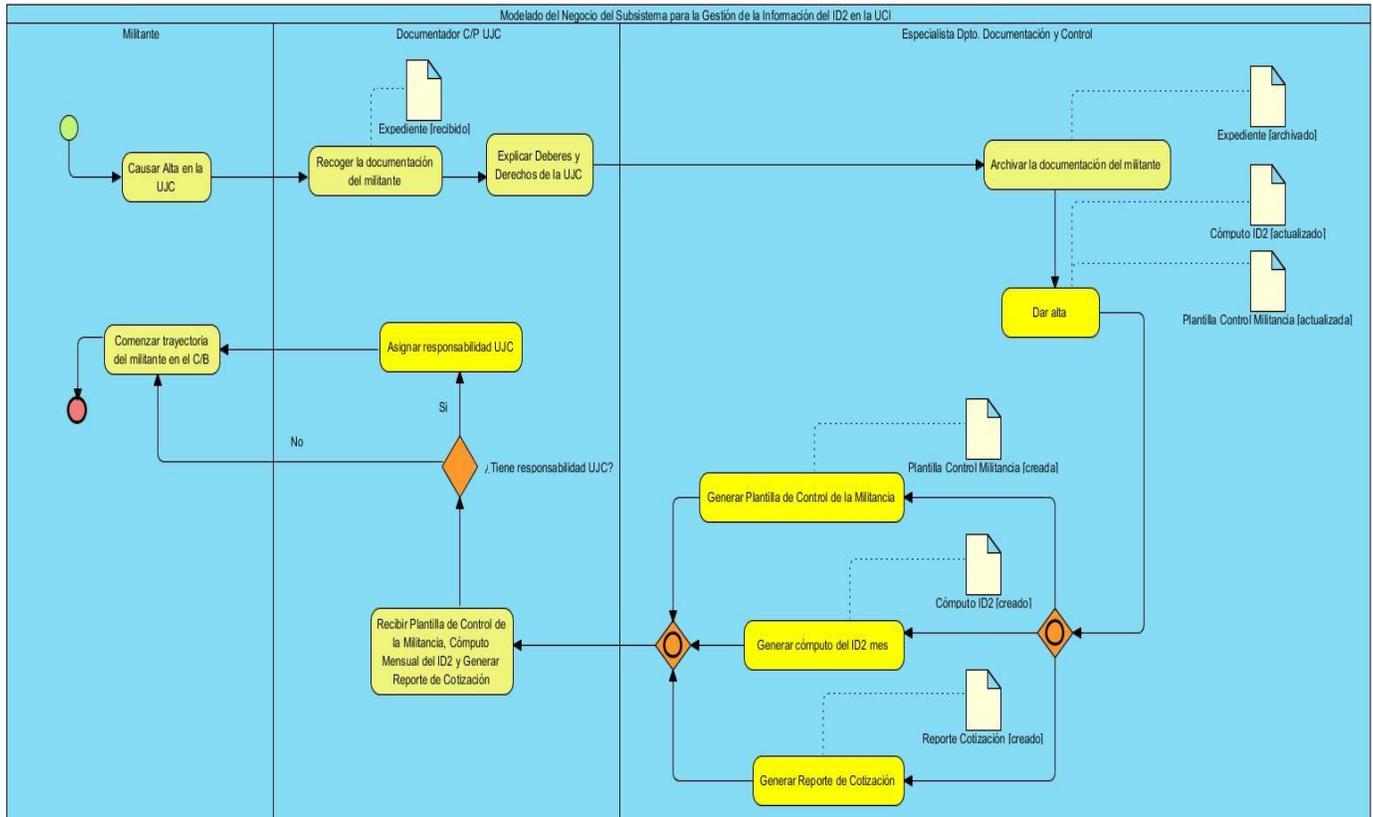


Figura 2: Modelo de Proceso de Negocio: Control de Altas de la UJC en la UCI.

Capítulo 2: Características del sistema.

Subsistema para la gestión del ID2 de la UJC de la UCI.

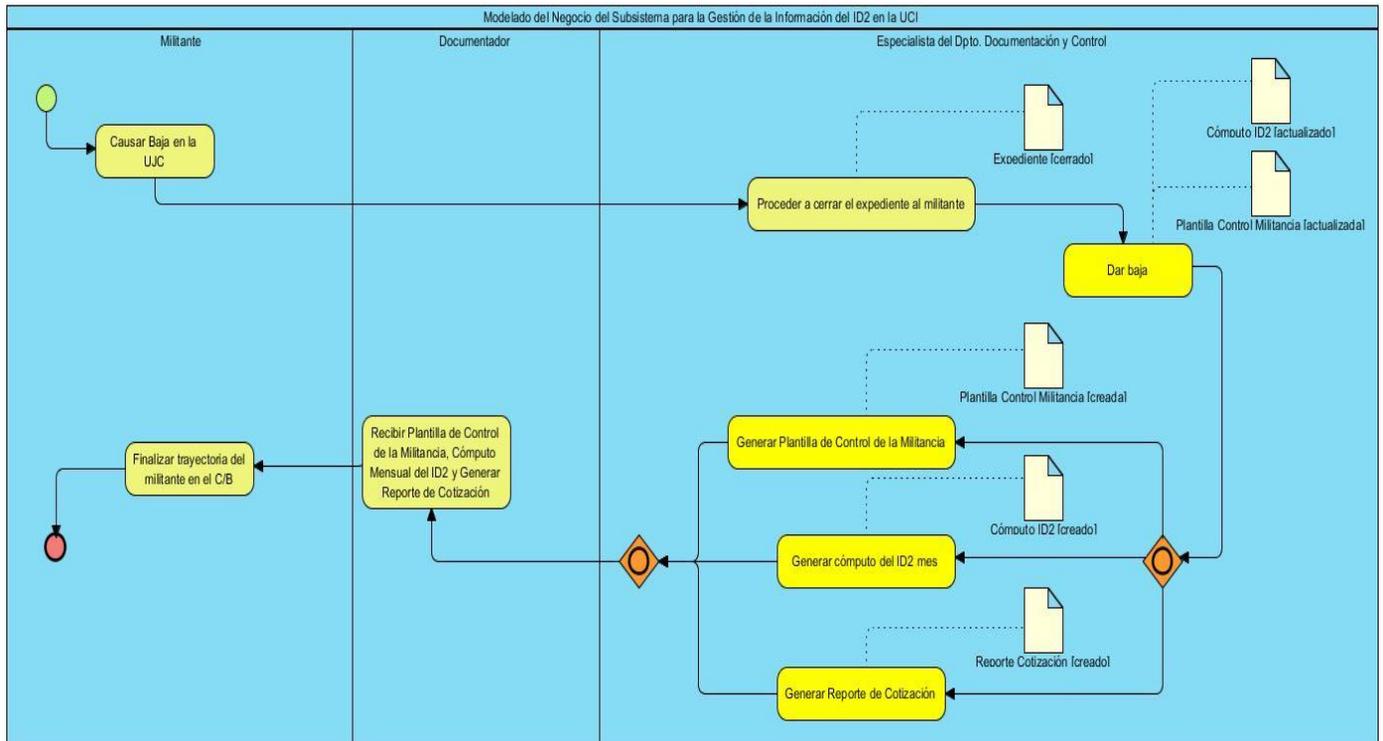


Figura 3: Modelo de Proceso de Negocio: Control de Bajas de la UJC en la UCI.

2.2.2 Descripción textual de los procesos de negocio

|                                  |  |
|----------------------------------|--|
| <b>Nombre:</b>                   | <b>Control de Altas de la UJC en la UCI.</b>   |
| <b>Objetivos:</b>                | Mantener actualizado el control del Estado de la Militancia de la UJC en la UCI, debido a los continuos movimientos que provocan las diversas altas en cada uno de los C/B.  |
| <b>Evento(s) que lo generan:</b> | <ul style="list-style-type: none"> <li>✓ Resultado de realizar algún proceso de crecimiento de la UJC.</li> <li>✓ Que un militante por una de las causas definidas dentro de sus estatutos, proceda a realizar algún traslado de C/B.</li> </ul> |
| <b>Precondiciones:</b>           | No aplica.   |
| <b>Postcondiciones:</b>          | No aplica.   |

## Capítulo 2: Características del sistema.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

|                           |  |
|---------------------------|--|
| <b>Reglas de Negocio:</b> | <p>Un militante solo puede ser incorporado en un único C/B en un período determinado.</p> <p>Un militante se puede incorporar a un C/B siempre y cuando no esté de alta en otro C/B, así como este posea su documentación.</p> <p>Un mismo militante solo puede efectuar un solo traslado en el mes.</p> <p>Cada vez que se produzca un traslado interno en un C/B, debe existir una incorporación interna en otro C/B de la universidad.</p> <p>El Cómputo de ID2 y la Plantilla de Control de la Militancia es actualizada durante todo el mes con cada uno de los movimientos realizado por parte de el/la especialista del Dpto. de Documentación.</p> <p>El mismo es generado finalmente los día 25 de cada mes. Este ID2 será válido durante el próximo mes.</p> |
| <b>Responsables:</b>      | Militante, Documentador de C/P UJC, Especialista del Departamento de Documentación y Control.  |
| <b>Clientes internos:</b> | No aplica.   |
| <b>Clientes externos:</b> | No aplica.   |
| <b>Entradas:</b>          | Expediente.  |
| <b>Salidas:</b>           | Plantilla de Control de la Militancia por C/B [Excel].<br>Cómputo de ID2 a nivel de C/P y Comité UJC UCI [Excel].  |
| <b>Actividades:</b>       | <ul style="list-style-type: none"> <li>✓ Causar Alta en la UJC.</li> <li>✓ Recoger la documentación del militante.</li> <li>✓ Explicar Deberes y Derechos de la UJC.</li> <li>✓ Archivar la documentación del militante.</li> <li>✓ Dar Alta.</li> <li>✓ Generar Plantilla de Control de la Militancia.</li> <li>✓ Generar cómputo del ID2 mes.</li> <li>✓ Generar Reporte de Cotización.</li> <li>✓ Recibir Plantilla de Control de la Militancia, Cómputo Mensual del ID2 y Generar Reporte de Cotización.</li> </ul>  |

## Capítulo 2: Características del sistema.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

|  |   |
|--|---|
|  | <ul style="list-style-type: none"> <li>✓ Asignar responsabilidad UJC.</li> <li>✓ Comenzar trayectoria del militante en el C/B.</li> </ul> |
|--|---|

Tabla 1: Control de altas de la UJC en la UCI.

| Nombre:                          | Control de Bajas de la UJC en la UCI.   |
|----------------------------------|---|
| <b>Objetivos:</b>                | Mantener actualizado el control del Estado de la Militancia de la UJC en la UCI, debido a los continuos movimientos que provocan las diversas bajas en cada uno de los C/B.   |
| <b>Evento(s) que lo generan:</b> | <ul style="list-style-type: none"> <li>✓ Que un militante por una de las causas definidas dentro de sus estatutos, proceda a realizar algún traslado de C/B.</li> <li>✓ Resultado de realizar algún proceso de crecimiento del PCC.</li> <li>✓ Aplicación de un tipo de sanción externa: Separación o Expulsión de las Filas.</li> <li>✓ Desactivación del militante.</li> </ul>  |
| <b>Precondiciones:</b>           | No aplica.  |
| <b>Postcondiciones:</b>          | No aplica.  |
| <b>Reglas de Negocio:</b>        | <p>Un mismo militante solo puede efectuar un solo traslado en el mes. Cada vez que se produzca un traslado interno en un C/B, debe existir una incorporación interna en otro C/B de la universidad.</p> <p>El Cómputo de ID2 y la Plantilla de Control de la Militancia es actualizada durante todo el mes con cada uno de los movimientos realizado por parte de el/la especialista del Dpto. de Documentación. El mismo es generado finalmente los día 25 de cada mes. Este ID2 será válido durante el próximo mes.</p> |
| <b>Responsables:</b>             | Militante, Documentador de C/P UJC, Especialista del Departamento de Documentación y Control.   |
| <b>Clientes internos:</b>        | No aplica.  |
| <b>Clientes externos:</b>        | No aplica.  |
| <b>Entradas:</b>                 | No aplica.  |

## Capítulo 2: Características del sistema.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

|                     |   |
|---------------------|---|
| <b>Salidas:</b>     | Plantilla de Control de la Militancia por C/B [Excel].<br>Cómputo de ID2 a nivel de C/P y Comité UJC UCI [Excel].   |
| <b>Actividades:</b> | <ul style="list-style-type: none"> <li>✓ Causar Baja en la UJC.</li> <li>✓ Proceder a cerrar el expediente al militante.</li> <li>✓ Dar baja.</li> <li>✓ Generar Plantilla de Control de la Militancia.</li> <li>✓ Generar cómputo del ID2 mes.</li> <li>✓ Generar Reporte de Cotización.</li> <li>✓ Recibir Plantilla de Control de la Militancia, Cómputo Mensual del ID2 y Generar Reporte de Cotización.</li> <li>✓ Finalizar trayectoria del militante en el C/B.</li> </ul> |

**Tabla 2: Control de bajas de la UJC en la UCI.**

#### 2.2.3 Actividades presentes en el negocio

1. Causar Alta o Baja en la UJC: Es provocado por varias causas:
  - ✓ Resultado de realizar algún proceso de crecimiento de la UJC.
  - ✓ Que el militante por una de las causas definidas dentro de sus estatutos, proceda a realizar algún traslado de C/B.
  - ✓ Resultado de realizar algún proceso de crecimiento del PCC.
  - ✓ Aplicación de un tipo de sanción externa: Separación o Expulsión de las Filas.
  - ✓ Desactivación del militante.
2. Recoger la documentación del militante: Comienza cuando el militante informa en su área que es miembro de la UJC, en ese momento el documentador procede a recogerle el expediente para trasladarlo al Departamento de Documentación y Control. También puede ocurrir que este entregue directamente su expediente en dicho departamento.
3. Explicar Deberes y Derechos de la UJC: El documentador procede a explicarle al militante sobre sus derechos y deberes que debe cumplir, actividad que se realiza con carácter formal.

## Capítulo 2: Características del sistema.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

4. Archivar la documentación del militante: Actividad que se produce en el Departamento de Documentación y Control lugar donde se archivan todos los expedientes en formato duro, los cuales son organizados mediante el uso de estantes por cada uno de los Comités Primarios.
5. Dar alta: Este proceso se inicia una vez conformados los comités de base para cada comité primario, el documentador procede a incorporar un militante dada la organización de base que le corresponde. Las mismas pueden ser producidas por diferentes causas: ajuste, incorporación del MININT, traslado de provincia, entre otras. Para esto el estudiante o trabajador al cual se le realizará el proceso. no puede pertenecer en ese momento a ningún comité de base.
6. Dar baja: Este proceso se inicia cuando el documentador procede a dar baja a un militante de un comité de base. Las mismas pueden ser: por ajuste, natural, por defunción, por traslado de área, entre otras. Para esto dicho miembro debe pertenecer a un comité de base.
7. Asignar responsabilidad: Esta actividad es iniciada una vez conformada la organización de base y que la misma tenga asignado por lo menos un miembro. En la primera reunión de un curso escolar, los militantes seleccionan a su secretario general. Se definen además otras responsabilidades dentro del mismo, como son: activista de actas (al menos uno), orientador político, activista de cotización, organizador. En el reglamento de la UJC está establecido que a una persona solo se le puede asignar una única responsabilidad al mismo tiempo.
8. Generar reporte de cotización: Este proceso se realiza mensualmente, el documentador para el caso de los trabajadores, necesita saber cuánto deben entregar ya que estos reciben ingresos por conceptos de salario. Los militantes que no reciben ingresos por salarios solo pagan 20 centavos. Para los trabajadores se le define una escala del pago de la cotización que se ajusta a la tarifa de los ingresos recibidos en el mes. (Ver anexo 2)
9. Generar Plantilla de Control de la Militancia: Los especialistas del Dpto. de Documentación y Control generan la plantilla del Control del Estado de la Militancia, donde se muestra por cada C/B diversos datos relacionados con sus militantes, como son: nombre, apellidos, dirección particular (compuesta por la provincia y el municipio), número de identidad, sexo, etc. La misma es generada los días 25 de cada mes (Ver anexo 1).
10. Generar cómputo del ID2: Comienza una vez generada todas las plantillas de Control del Estado de la Militancia por comités de base. Se genera el cómputo por comité primario para tener una idea de los movimientos ocurridos en este nivel por concepto de bajas y altas durante el mes y compararlo

## Capítulo 2: Características del sistema.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

con el mes anterior. Esta actividad también es realizada a nivel de Comité UJC UCI, la cual tendría una información global del comportamiento de los Comité Primarios de la universidad (Ver anexo 1).

11. Recibir Plantilla de Control de la Militancia, Cómputo Mensual del ID2 y Generar Reporte de Cotización: Después de cerrado el ID2 del mes son enviadas a los documentadores, vía email las Platillas de Control del Estado de la Militancia de su C/P, así como el reporte mensual del cómputo ID2 de toda la universidad y el Reporte de Cotización para proceder a la recogida de la misma el próximo mes.
12. Comenzar trayectoria del militante en el C/B: El militante comienza a formar parte del C/B asignado hasta que se produzca una de las causas que inicien el proceso de Dar Baja.
13. Proceder a cerrar el expediente al militante: Actividad que se produce en el Departamento de Documentación y Control, a través de la cual se procede a realizar el cierre y empaque del expediente en formato duro para su posterior traslado. Solo es iniciado debido alguna causa de traslado del militante.
14. Finalizar trayectoria del militante en el C/B: El militante culmina de formar parte del C/B que tenía asignado hasta el momento.

### 2.3 Objeto de Automatización

En la actualidad la gestión del ID2 es un proceso que resulta engorroso. El mismo es llevado a cabo manualmente con un período mensual, retrasando así el tiempo de las actividades. En la organización se manejan grandes volúmenes de información y se ejecutan numerosos cambios que obligan a los documentadores a realizar una tarea ardua para mantener actualizados los mismos. Por lo que se propone para la erradicación de dichos problemas el desarrollo de un subsistema, como parte del SIUJC que permita la gestión de la información; con el objetivo de minimizar el tiempo en el cumplimiento de las tareas, el manejo y organización de los datos.

### 2.4 Propuesta del sistema

Se propone el desarrollo de un subsistema, como parte del SIUJC que permita la gestión de la información, con el objetivo de minimizar el tiempo empleado en: la consulta de los datos y/o la organización de los mismos, la generación del cómputo del ID2, un control sobre los movimientos realizados por los militantes,

## Capítulo 2: Características del sistema.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

los secretarios de las organizaciones de base, entre otros reportes de importancia para la UJC. A esta aplicación web podrán acceder los documentadores de cada comité primario, los especialistas del Departamento de Documentación y Control así como el secretario del comité primario UCI, permitiéndole a estos realizar dichas actividades. En caso de que ocurra algún cambio de los mismos, el administrador del sistema podrá modificar la accesibilidad para que esta persona pueda realizar las tareas correspondientes correctamente.

Se plantea la realización de una aplicación web, permitiendo consultar la información contenida desde cualquier estación de trabajo conectada a la red de la institución. Las actualizaciones o mejoras que se le realicen a la aplicación resultan transparentes al usuario final, pues se realizan solamente en la estación que actúa como servidor de aplicaciones web, independiente completamente de la estación cliente. Para acceder a la aplicación sólo se necesita de un navegador web y la conexión puede ser desde cualquier computadora que esté conectada a la red.

#### 2.5 Personas Relacionadas con el sistema

Las personas relacionadas con el sistema son aquellas que obtienen un resultado del valor de uno o varios procesos que se ejecutan o realizan en el mismo. También son aquellas que no reciben ningún valor, pero que están estrechamente vinculadas con las actividades que efectúan.

| Personas Relacionadas con el Sistema    | Descripción   |
|---|---|
| <b>Administrador</b>                    | Puede realizar todas las funcionalidades definidas para el sistema, efectúa reportes y controla la información en cualquier momento. Administra elementos propios del sistema.  |
| <b>Documentadores de la UJC</b>         | Actualizan los movimientos ocurridos en sus organizaciones de base, la información de los militantes, así como la generación de reportes entre ellos el cómputo o cierre del ID2 tanto por comité primario como por comité de base. |
| <b>Especialista del departamento de</b> | Puede realizar todas las funcionalidades definidas para el  |

## Capítulo 2: Características del sistema.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

|                      |  |
|----------------------|--|
| <b>documentación</b> | sistema, efectúa reportes y controla la información de los comités Primarios en cualquier momento. |
|----------------------|--|

**Tabla 3: Relación de las personas con el sistema.**

#### 2.6 Requisitos no funcionales del sistema

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable (23). Están estrechamente vinculados a los requisitos funcionales, puesto que una vez que está definido lo que el sistema debe hacer, es necesario especificar cómo ha de hacerlo. Pueden llegar a marcar la diferencia entre un producto bien aceptado por los clientes o usuarios y uno de poca o ninguna calidad o aceptación. A continuación se encuentra la descripción de los requisitos no funcionales.

**Usabilidad:** Garantizar un acceso fácil y rápido a los usuarios: El sistema debe presentar una interfaz amigable que permita la fácil interacción con el mismo. Debe, además, ser una interfaz de manejo cómodo que posibilite a los usuarios sin experiencia una rápida adaptación.

**Apariencia o Interfaz interna:** La aplicación propuesta debe tener una interfaz sencilla dirigida a las personas que interactúen con el sistema; estableciendo una plantilla general para toda la aplicación, de forma tal que la información sólo varíe en la parte central de la misma. Cada página no debe exceder los 500 kb en imágenes.

#### Restricciones de software:

- ✓ En las computadoras de los clientes solo se requiere de un navegador web (Internet Explorer versión 4.5 o superior, Mozilla Firefox versión 3.0 o superior).
- ✓ Servidor Web Apache v2.2.2 o superior con módulo PHP v5.1 disponible, este debe estar configurado con las extensiones "ldap", "soap" y "pgsql".
- ✓ El SGBD deberá ser PostgreSQL preferiblemente v9.1.1 en adelante.

#### Hardware:

## *Capítulo 2: Características del sistema.*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

- ✓ Para las estaciones de trabajo: PC Pentium 3 o superior, CPU 133 MHZ o superior, 256 RAM mínimo, 512 RAM recomendada o superior. Se requiere que tengan tarjeta de red.
- ✓ Para el servidor: CPU Dual Core 2.0 GHZ o superior, memoria RAM de 2 GB, 250 GB HDD. Se requiere que tenga tarjeta de red.

#### **Rendimiento:**

- ✓ El subsistema deberá ser capaz de gestionar toda la información y dar respuesta a las solicitudes lo más rápido posible.
- ✓ Las páginas web deben cargar en el menor tiempo posible.

#### **Seguridad:**

- ✓ Se podrá acceder al sistema solamente después de autenticarse.
- ✓ Se garantizará que las funcionalidades del subsistema se muestren de acuerdo al cargo que ocupe dentro del C/P.
- ✓ Se garantizará que la información sensible solo pueda ser vista por los usuarios con el rol autorizado para ello.
- ✓ Tiene que ser capaz de anular acciones incorrectas que atenten contra la integridad de los datos.

#### **Políticos-culturales:**

- ✓ Se empleará en el subsistema el idioma español.
- ✓ Se contará con logotipos e imágenes que se encuentren en correspondencia con la UJC y refleje los ideales de la organización.
- ✓ Algún cambio que se quiera realizar debe ser previamente consultado con la dirección del Comité UJC-UCI.

#### **Confiabilidad:**

## Capítulo 2: Características del sistema.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

- ✓ Debe garantizarse el resguardo de la información (imágenes, documentos), así como la grabación periódica (backups) de la Base de Datos, de forma tal que se posibilite la reinstalación del sistema y los datos, en caso de fallos en el sistema o en el hardware.

## 2.7 Fase de Exploración

Como se menciona en epígrafes anteriores en esta fase se define el alcance real del sistema permitiendo una familiarización del equipo de desarrollo con las herramientas, tecnologías y procesos a utilizar. Esta fase comienza por la creación de Historias de Usuarios (HU) las cuales definen mediante su redacción, qué es lo que verdaderamente necesita el cliente y es aquí donde los programadores estiman el tiempo de desarrollo

### 2.7.1 Historia de Usuario

La HU es la técnica utilizada en XP para especificar los requisitos del software. Estas proporcionan una breve descripción de los requisitos funcionales con los que el sistema debe cumplir. Las mismas son utilizadas para la estimación del riesgo y cuánto tiempo será empleado en su implementación. Es el cliente el encargado de asignarle una prioridad a cada HU y el equipo de desarrollo el encargado de asignar el costo. Es importante que se destaque, que las HU pueden describirse en cualquier momento, con esto se comprueba la flexibilidad de la metodología. El contenido de estas debe ser sencillo y concreto.

A continuación se describe la plantilla que se utilizará para definir las HU.

| Historia de Usuario  |   |
|--|---|
| <b>Número:</b> (Número de la historia de usuario incremental en el tiempo).                          | <b>Usuario:</b> (Usuario que protagoniza la historia de usuario).                               |
| <b>Nombre de historia:</b> (Sirve para identificarlas mejor entre los desarrolladores y el cliente). |   |
| <b>Prioridad en negocio:</b> (La importancia que tiene para el cliente). (Alta / Media / Baja).      | <b>Riesgo en desarrollo:</b> (Qué tan difícil es para el desarrollador). (Alto / Medio / Bajo). |
| <b>Puntos estimados:</b> (Tiempo que durará)   | <b>Iteración asignada:</b> (Se defina la iteración en la cual será)                             |

## Capítulo 2: Características del sistema.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

|  |                      |
|--|----------------------|
| el desarrollo de la HU (Estimado en semanas)).   | desarrollada la HU). |
| <b>Programador responsable:</b> Persona responsable de la implementación de la de la historia de usuario).           |                      |
| <b>Descripción:</b> (Se especifican las operaciones, que deberá realizar el sistema y las respuestas que este dará). |                      |
| <b>Observaciones:</b> (Algunas observaciones de interés como información sobre usuarios, condiciones, etc.).         |                      |

**Tabla 4: Plantilla para definir las HU.**

En las siguientes tablas se muestran algunas de las HU: priorizadas como Alta. El resto de las HU se incluyen en el Anexo 3.

| Historia de Usuario   |                                    |
|---|------------------------------------|
| <b>Número:</b> 6.   | <b>Usuario:</b> Documentador.      |
| <b>Nombre de historia:</b> Gestionar Comités de Base.   |                                    |
| <b>Prioridad en negocio:</b> Alta.  | <b>Riesgo en desarrollo:</b> Alta. |
| <b>Puntos estimados:</b> 2.   | <b>Iteración asignada:</b> 1.      |
| <b>Programador responsable:</b> Yilen Yamilé Reyes Montejo.   |                                    |
| <b>Descripción:</b> La aplicación debe permitir adicionar, modificar, eliminar y mostrar un comité de base. |                                    |
| <b>Observaciones:</b> El militante no puede pertenecer a otro comité de base en ese momento.                |                                    |

**Tabla 5: HU Conformar Comité de base.**

| Historia de Usuario  |                                    |
|--|------------------------------------|
| <b>Número:</b> 1.  | <b>Usuario:</b> Documentador.      |
| <b>Nombre de historia:</b> Generar plantilla del control de la militancia.                 |                                    |
| <b>Prioridad en negocio:</b> Alta.   | <b>Riesgo en desarrollo:</b> Alta. |
| <b>Puntos estimados:</b> 3.  | <b>Iteración asignada:</b> 1.      |
| <b>Programador responsable:</b> Yilen Yamilé Reyes Montejo.                                |                                    |
| <b>Descripción:</b> Muestra las movimientos ocurridos en un comité de base durante el mes. |                                    |
| <b>Observaciones:</b> Estos movimientos se realizan mensualmente.                          |                                    |

## Capítulo 2: Características del sistema.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

Tabla 6: HU Generar cómputo del ID2.

| Historia de Usuario  |                                    |
|--|------------------------------------|
| <b>Número:</b> 11.   | <b>Usuario:</b> Documentador.      |
| <b>Nombre de historia:</b> Gestionar Militante.  |                                    |
| <b>Prioridad en negocio:</b> Alta.   | <b>Riesgo en desarrollo:</b> Alta. |
| <b>Puntos estimados:</b> 1.  | <b>Iteración asignada:</b> 1.      |
| <b>Programador responsable:</b> Yilen Yamilé Reyes Montejo.  |                                    |
| <b>Descripción:</b> Permite adicionar, modificar, eliminar y mostrar los militantes así como realizar acciones con los mismo (Dar alta, baja y asignarle una responsabilidad). |                                    |
| <b>Observaciones:</b>  |                                    |

Tabla 7: HU Gestionar Militante.

| Historia de Usuario  |                                    |
|--|------------------------------------|
| <b>Número:</b> 8.  | <b>Usuario:</b> Documentador.      |
| <b>Nombre de historia:</b> Asignar responsabilidad UJC a un militante.                                     |                                    |
| <b>Prioridad en negocio:</b> Alta.   | <b>Riesgo en desarrollo:</b> Alta. |
| <b>Puntos estimados:</b> 1.  | <b>Iteración asignada:</b> 2.      |
| <b>Programador responsable:</b> Yilen Yamilé Reyes Montejo.  |                                    |
| <b>Descripción:</b> Permite asignar una responsabilidad a un militante que pertenezca a un comité de base. |                                    |
| <b>Observaciones:</b> El militante no puede tener ninguna responsabilidad.                                 |                                    |

Tabla 8: HU Asignar responsabilidad UJC a un militante.

## 2.8 Planificación

En esta fase se planifica el proceso de desarrollo de las aplicaciones de software como bien se menciona en el epígrafe 1.4.1. La planificación se apoya en la fase de exploración, ya que en esta se describen las HU, realizando así una estimación del esfuerzo que costará implementar cada HU.

## Capítulo 2: Características del sistema.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

#### 2.8.1 Estimación de esfuerzos por historias de usuario

Para la realización del sistema se hizo una estimación del tiempo previsto para cada una de las HU que se identificaron. A continuación se detalla este resultado para las HU antes mencionadas en la tabla 8. Para el resto de las HU. (Ver anexo 4).

| Historia de Usuario                         | Puntos de estimación (Semanas) |
|---|--------------------------------|
| Gestionar Comité de Base.                   | 2.                             |
| Generar cómputo del ID2.                    | 3.                             |
| Gestionar Militante                         | 1.                             |
| Asignar responsabilidad UJC a un militante. | 1.                             |

**Tabla 9: Estimación de esfuerzos por Historia de Usuarios.**

#### 2.9 Iteraciones

En esta etapa es el cliente el que decide que HU está en cada una de las diferentes iteraciones. Y es el equipo de desarrollo el que decide cuantas etapas serán realizadas, obteniendo así un mayor desempeño en el desarrollo del sistema. En el caso del Subsistema para la gestión del ID2 de la UJC de la UCI, se definieron tres iteraciones. En la siguiente tabla a continuación se muestran las iteraciones y el orden a implementar de cada HU.

| Iteración | Orden de la Historia de usuario a implementar   | Duración total de las iteraciones |
|-----------|---|-----------------------------------|
| 1         | Generar Plantillas de control del estado de la militancia.<br>Generar Cómputo del ID2.<br>Dar alta a un militante en un comité de base.<br>Dar Baja a un militante en un comité de base.<br>Gestionar comité primario.<br>Gestionar comité de base.<br>Gestionar militante. | 12 ½ semanas                      |

## Capítulo 2: Características del sistema.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

|   |  |             |
|---|--|-------------|
| 2 | Consumir de los servicios Web.<br>Gestionar asignar responsabilidad.<br>Mostrar expediente de un militante.<br>Exportar modelo del ID2 en diferentes formatos (XLS). | 5 ½ semanas |
| 3 | Buscar responsable de un comité de base.<br>Listar universo juvenil.<br>Listar militantes con edad de arribar al Partido Comunista de Cuba (PCC).                    | 2 semanas   |

**Tabla 10: Plan de duración de las iteraciones.**

#### 2.9.1 Iteración 1

En la primera iteración se implementan las HU con prioridad alta en el negocio, garantizando una primera versión del producto; implementando las funcionalidades del sistema aunque aún sin terminar, mostrando una primera vista al cliente que dará su criterio para la incorporación de nuevos elementos y modificación. Las historias de usuarios a implementar en esta iteración son: 1, 2, 3, 4, 5, 6, 11. Además, se tendrá la primera versión de prueba, la cual será mostrada al cliente con el objetivo de obtener una retroalimentación para el grupo de trabajo.

#### 2.9.2 Iteración 2

En la segunda iteración se implementarán las restantes funcionalidades con prioridad alta y otras con prioridad media. Para ello se implementan las HU: 7, 8, 9, 10. Con la culminación de esta iteración al igual que la primera se contará con otra realización de pruebas del producto.

#### 2.9.3 Iteración 3

En la tercera iteración se implementan las restantes funcionalidades de prioridad media y todas las HU pendientes que complementen la totalidad del sistema. Para ello se implementan las HU: 12, 13, 14, 15.

## Capítulo 2: Características del sistema.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

Como resultado de la iteración se obtendrá una tercera versión del sistema, la cual será sometida al igual que en las iteraciones anteriores a consideración del cliente para atender no conformidades finales.

#### 2.10 Plan de entrega

A continuación se describe el plan de entrega previsto para la fase de implementación. El mismo está concebido de acuerdo a las iteraciones previstas y a las HU.

| Historias de Usuarios                                      | Final 1ra Iteración<br>1ra semana abril | Final 2da Iteración<br>2da semana mayo | Final 3ra Iteración<br>4ta semana mayo |
|--|---|--|--|
| Generar Plantillas de Control del Estado de la Militancia. | V 1.0                                   | V 1.1                                  | Finalizado                             |
| Generar Cómputo del ID2.                                   | V 1.0                                   | V 1.1                                  | Finalizado                             |
| Dar alta a un militante en un comité de base.              | V 1.0                                   | V 1.1                                  | Finalizado                             |
| Dar Baja a un militante en un comité de base.              | V 1.0                                   | V 1.1                                  | Finalizado                             |
| Gestionar comité primario.                                 | V 1.0                                   | V 1.1                                  | Finalizado                             |
| Gestionar comité de Base.                                  | V 1.0                                   | V 1.1                                  | Finalizado                             |
| Gestionar militante.                                       | V 1.0                                   | V 1.1                                  | Finalizado                             |
| Consumir de los servicios Web.                             |   | V 1.0                                  | Finalizado                             |
| Asignar responsabilidad UJC a un militante.                |   | V 1.0                                  | V 1.1                                  |
| Mostrar expediente de un militante.                        |   | V 1.0                                  | V 1.1                                  |

*Capítulo 2: Características del sistema.*

*Subsistema para la gestión del ID2 de la UJC de la UCI.*

|   |  |       |       |
|---|--|-------|-------|
| Exportar modelo del ID2 en diferentes formatos (XLS).                     |  | V 1.0 | V 1.1 |
| Buscar responsable de un comité de base.                                  |  |       | V 1.0 |
| Listar universo juvenil.  |  |       | V 1.0 |
| Listar militantes con edad de arribar al Partido Comunista de Cuba (PCC). |  |       | V 1.0 |
| Listar militantes con edad de arribar el PCC.                             |  |       | V 1.0 |

**Tabla 11: Plan de entrega.**

**2.11 Conclusiones**

En este capítulo se definió la propuesta de solución, a través de la notación de los procesos del negocio y la descripción de las actividades presentes en el mismo. Se detallaron las HU y el esfuerzo necesario para la realización y cumplimiento de las mismas. Se describieron además los requisitos no funcionales del subsistema. Se definieron las iteraciones a realizar por el equipo de desarrollo y las HU a desarrollar en cada una de ellas. A partir de esto se comenzará a desarrollar la etapa de diseño del mismo.

## *Capítulo 3: Diseño del sistema.*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

## **Capítulo 3**

### **Diseño del sistema**

#### **3.1 Introducción**

En este capítulo se analizan los principios del diseño, de modo que sea consistente con el entorno de implementación. Se define la propuesta de solución de la arquitectura del sistema y se explican los patrones de diseño a utilizar en el desarrollo del mismo. Se detallan las tareas de ingeniería para las iteraciones de cada una de las HU. Para finalizar se propone la descripción del modelo de datos para el desarrollo del subsistema.

#### **3.2 Arquitectura**

Hoy día, es común encontrar numerosas definiciones acerca de la arquitectura de software. Cada una con diversos planteamientos acerca de este concepto. Por lo que su conceptualización sigue aún en discusión. A continuación se hace referencia a un concepto de arquitectura de software.

La arquitectura de software es una descripción de los subsistemas y componentes (computacionales) de un sistema software y las relaciones entre ellos. Es el resultado de la actividad de diseño arquitectónico del software (24).

##### **3.2.1 Patrón de Arquitectura**

Los patrones arquitectónicos expresan la estructura fundamental para el sistema de software a desarrollar. Para desarrollo del subsistema, se utilizó el patrón de arquitectura para aplicaciones web Modelo -Vista - Controlador (MVC).

Yii implementa el patrón arquitectónico Modelo-Vista-Controlador. MVC separa la lógica de negocio, de la interfaz de usuario, de la clase controladora, de modo que los desarrolladores pueden cambiar fácilmente cada parte sin afectar en gran medida a otra. En MVC:

## Capítulo 3: Diseño del sistema.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

El **modelo** representa la información (los datos) y las reglas de negocio, **la vista** contiene elementos de la interfaz de usuario, tales como entradas de texto, formularios entre otros, y **el controlador** gestiona la comunicación entre el modelo y la vista.

Yii también además del MVC, introduce un controlador frontal, llamado aplicación, el cual representa el contexto de ejecución de las peticiones en proceso. La aplicación resuelve las peticiones de los usuarios y las despacha al controlador apropiado para futuros manejos (21).

Además del MVC, Yii también introduce un front-controller llamado aplicación, el cual representa el contexto de ejecución del procesamiento del pedido. La aplicación resuelve el pedido del usuario y la dispara al controlador apropiado para tratamiento futuro (21). (Ver imagen 3).

Como se explica anteriormente el patrón de diseño que utiliza Yii es el controlador frontal; representado por la página principal index.php, el cual es el único punto de acceso del usuario a la aplicación web. En el controlador frontal se crea la instancia de la aplicación, cuya función es la de recibir los requerimientos del usuario y remitirlos a los controladores apropiados para su posterior procesamiento. El controlador frontal representa el contexto de ejecución del procesamiento de los requerimientos del usuario, por tanto se puede acceder en cualquier momento a través de la aplicación del patrón Singleton Yii::app() (21).

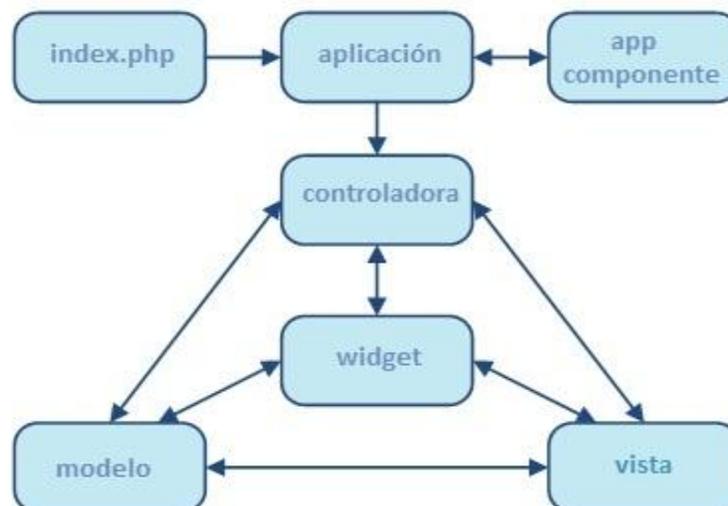


Figura 4: Estructura estática de aplicación Yii.

## Capítulo 3: Diseño del sistema.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

#### 3.3 Patrones de diseño

Según Christopher Alexander (1977) los patrones: (...) describen un problema que ocurre una y otra vez en el entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces, sin hacerlo siquiera dos veces de la misma forma (...) (25). Un patrón de diseño es una descripción de un problema, este no es más que la solución estándar para un problema común de programación. Para que la forma de resolver un problema sea considerada un patrón debe poseer ciertas características. Una de ellas es que se debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores.

##### 3.3.1 Patrones GOF (Gang of Four, Banda de Cuatro)

Los patrones GOF son aquellos que abstraen el proceso de creación de instancias, se ocupan de componer estructuras de mayor tamaño y asignan responsabilidades entre objetos. A continuación se explica el patrón utilizado para el desarrollo del subsistema.

**Singleton:** Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Yii implementa una única instancia del controlador frontal mediante este patrón con el objetivo de lograr una vía de entrada única a las peticiones de los usuarios como se muestra en la figura 5.

**Ejemplo:** El script de entrada es el script de inicio y es el que se ocupa de procesar el pedido del usuario inicialmente. Es el único script PHP que el usuario puede pedir directamente para ejecutarse (21).

##### 3.3.2 Patrones para Asignar Responsabilidades (GRASP)

Los Patrones Generales de Software para Asignar Responsabilidades o General Responsibility Assignment Software Patterns (GRASP) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (21), a continuación se explican los patrones GRASP que se utilizaron para la implementación del subsistema.

## Capítulo 3: Diseño del sistema.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

**Experto:** Plantea que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo se obtendrá un diseño con mayor cohesión, la información se mantiene encapsulada y permite contar con un sistema robusto y fácil de mantener (25).

Otros ejemplos pueden ser encontrados en las clases: **Comité de base**, comité primario, responsabilidad, recorrido, donde los métodos y la creación de objetos son realizados solo por estas clases.

**Creador:** El patrón creador permite a identificar quién debe ser el responsable de la creación o instanciación de nuevos objetos o clases (25), de forma tal que la instancia de un objeto sólo pueda ser creada por el objeto que contiene la información necesaria para ello.

Otros ejemplos de la utilización de este patrón se encuentran en las siguientes clases: **Comiteprimario** o **Comitebase**, responsabilidad, recorrido. Siendo estas creadas o instanciadas solo por aquellas clases que contienen la información necesaria para hacerlo.

**Controlador:** El patrón controlador sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, este sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto es utilizado para aumentar la reutilización de código y los componentes (25).

#### 3.4 Tarjetas Clase – Responsabilidad – Colaborador (CRC)

La metodología XP, como parte del diseño propone el modelado de CRC, constituye un modo simple de organizar las clases más notables para las funcionalidades del sistema.

Un modelo CRC es una colección de tarjetas índices estándar que representan clases. Estas tarjetas se componen por tres secciones:

- ✓ La primera se ubica a lo largo del borde superior de la tarjeta donde se escribe el nombre de la clase.
- ✓ La segunda sección es en el cuerpo de la tarjeta a la izquierda y es donde se listan las responsabilidades de la clase (que no es más que lo que la clase sabe o hace).

## Capítulo 3: Diseño del sistema.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

- ✓ La tercera sección se encuentra a la derecha donde se ubican los colaboradores (son aquellas clases con las que interactúa y recibe información para completar su responsabilidad).

En la siguiente tabla se muestra un ejemplo de una tarjeta CRC correspondiente a la clase Crear comité de base. En el anexo 5 se muestran las tarjetas más relevantes para las funcionalidades del subsistema para la gestión del ID2 de la UJC de la UCI.

| Clase: ComiteBase  |   |
|--|---|
| Responsabilidad  | Colaboración                                |
| Crear un comité de base, desintegrar un comité de base, modificar un comité de base. | Comiteprimario.<br>Militante.<br>Recorrido. |

Tabla 12: Tarjeta CRC Comité de base.

### 3.5 Modelo de Datos

La gestión de los datos en los Sistemas de Gestión de Información (SGI) es de suma importancia para el desarrollo de los mismos y aunque la modelación y gestión de la base de datos no se especifica dentro de la metodología XP, son temas necesarios de abordar debido a la repercusión que tienen sobre los SGI.

Por lo general, el modelo de datos permite describir las estructuras de la base de datos (el tipo que se incluye y la forma en que se relacionan), las restricciones de integridad (las condiciones que los mismos deben cumplir para reflejar correctamente la realidad deseada) y las operaciones de manipulación (adicionar, eliminar, modificar, entre otras).

Subsistema para la gestión del ID2 de la UJC de la UCI.

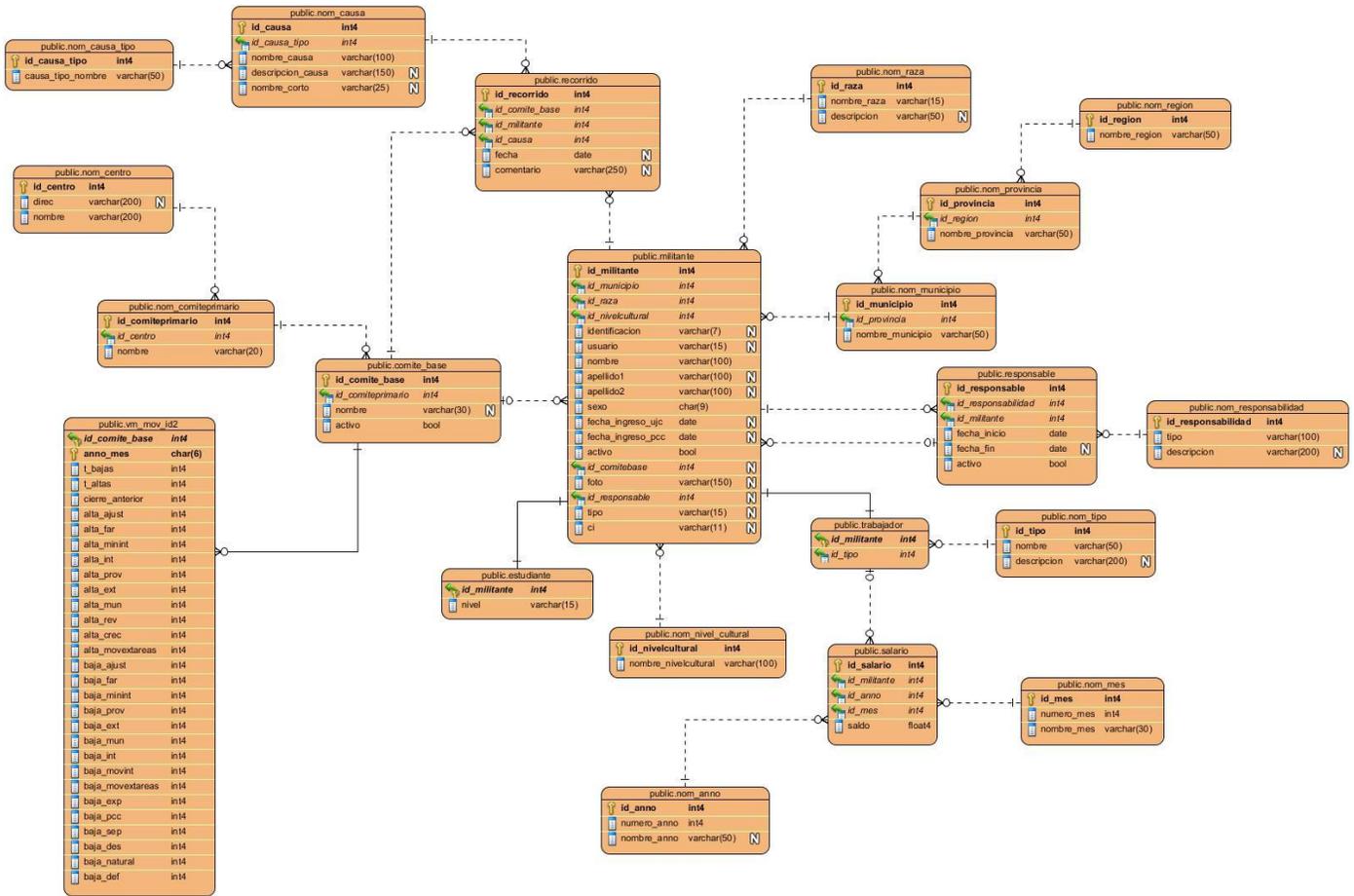


Figura 5: Modelo de datos.

3.6 Conclusiones

En el capítulo se definió la arquitectura utilizar por el subsistema, se explica brevemente el patrón de arquitectura MVC, Se explicaron los patrones de diseño utilizados y donde se utilizaron. Se dio una breve descripción de los artefactos generados por la metodología XP: las tarjetas CRC, así como se presentó la propuesta del modelo de datos.

## *Capítulo 4: Implementación y Prueba.*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

## **Capítulo 4**

### **Implementación y Prueba**

#### **4.1 Introducción**

El desarrollo de software para XP se realiza de manera iterativa, obteniendo como resultado de cada una de estas iteraciones un producto funcional; el cual debe ser sometido a pruebas y mostrado al cliente, permitiendo de esta forma una retroalimentación con los desarrolladores. En el siguiente capítulo se detallan las tres iteraciones llevadas a cabo durante la construcción del subsistema. En cada una de ellas se procede a la implementación de historias de usuarios definidas para estas. De igual modo se describen las estrategias de prueba a efectuar sobre el producto de software obtenido.

#### **4.2 Estándares**

Los estándares son acuerdos (normas) documentados que contienen especificaciones técnicas u otros criterios precisos para ser usados consistentemente como reglas, guías o definiciones de características. Para asegurar que los materiales productos, procesos y servicios se ajusten a su propósito (26). En la realización de sistemas informáticos es común la utilización de estándares de programación. Los mismos se utilizan para la correspondencia de lo que se programa, la documentación y la base de datos. El código de un software lo revisan varias personas con el objetivo de: arreglarlo, repararlo, revisarlo o ampliarlo. Por esto la necesidad de que el software y los artefactos que tributan a este, estén bien redactados y con estilo para que su significado sea claro.

##### **4.2.1 Estándar de codificación para la base de datos**

#### **Aspectos generales**

Se establece el uso del idioma español para todos los elementos que intervienen en este proceso de codificación, dígame nombre de las base de datos, tablas, atributos, vistas, funciones, disparadores, restricciones entre otros.

1. El nombre de las tablas nomencladoras deben comenzar con el prefijo nom\_, el nombre de las tablas de la base de datos se escribe con minúscula y en caso de ser nombres compuestos deberán escribirse separados por el carácter “\_”.



## Capítulo 4: Implementación y Prueba.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

- ✓ **Ejemplo:** nom\_causa, nom\_comiteprimario, responsable, militante, recorrido.
- 2. El nombre de los atributos se define en minúscula, lo más cercano a la descripción de la característica del objeto
- ✓ **Ejemplo:** nombre, apellido, usuario, sexo, activo, dirección.
- 3. Los atributos compuestos por más de una palabra deberán ir separados por el carácter “\_”.
- ✓ **Ejemplo:** fecha\_ingreso\_ujc, fecha\_ingreso\_pcc, causa\_tipo\_nombre, fecha\_inicio.
- 4. El nombre de la base de datos deberá ser escrito con minúscula.
- ✓ **Ejemplo:** ujc.
- 5. Las tablas que no poseen identificador propio, tendrán como identificador el prefijo id\_ y a continuación el nombre de la tabla a la cual hacen referencia.
- ✓ **Ejemplo:** En la tabla estudiante el identificador será id\_militante, en la tabla trabajador id\_militante. Esto es debido a la relación de herencia que existen entre estas dos tablas con respecto a la tabla militante.
- 7. Las tablas con identificador propio, utilizarán como identificador el prefijo id\_ y a continuación el nombre de la tabla.
- ✓ **Ejemplo:** id\_responsable, id\_recorrido, id\_tipo, id\_militante.

#### 4.2.2 Estándar de codificación para PHP

##### 4.2.2.1 Ubicación de las clases

Las clases del sistema se encuentran en la carpeta protected. Los modelos en /protected/models, las controladoras dentro de /protected/controllers y las vistas se encuentran en /protected/views. Como se muestra a continuación en la figura. 5.

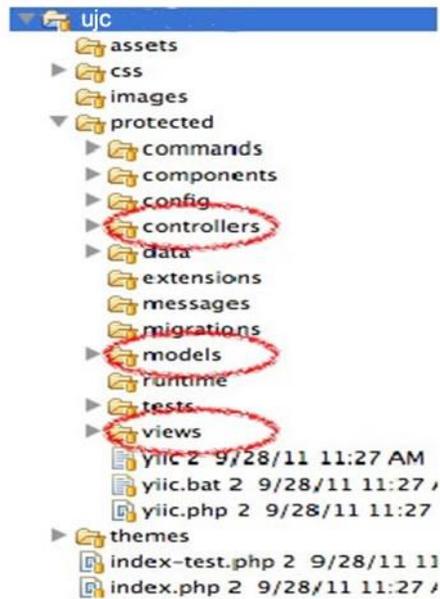


Figura 6: Ubicación de las clases.

#### 4.2.2.2 Estructura de las clases

##### 4.2.2.2.1 Clases Controladoras

El nombre de las clases controladoras debe comenzar con mayúscula y será acompañado por el sufijo Controller.

#### Ejemplo:

```
<?php

class ComiteBaseController extends Controller
{

    public $layout='//layouts/column2';

    public function filters()
    {
        return array(
            'accessControl', // perform access control for CRUD operations
        );
    }
}
```

Figura 7: Clase controladora comité de base.

## Capítulo 4: Implementación y Prueba.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

#### 4.2.2.2.2 Clases Modelos

Los nombres de las clases modelos siempre comenzaran con mayúsculas al igual que las controladoras, si están compuestas por dos palabras, la segunda también comenzara con mayúscula.

#### Ejemplo:

```
<?php

class KomiteBase extends CActiveRecord
{
    public static function model($className=__CLASS__)
    {
        return parent::model($className);
    }

    public function tableName()
    {
        return 'comite_base';
    }
}
```

Figura 8: Clase modelo comité de base.

#### 4.2.2.2.3 Clases Vistas

El nombre de estas clases debe comenzar con minúscula, si algún nombre está descrito por dos palabras, la segunda deberá comenzar con mayúscula.

#### Ejemplo:

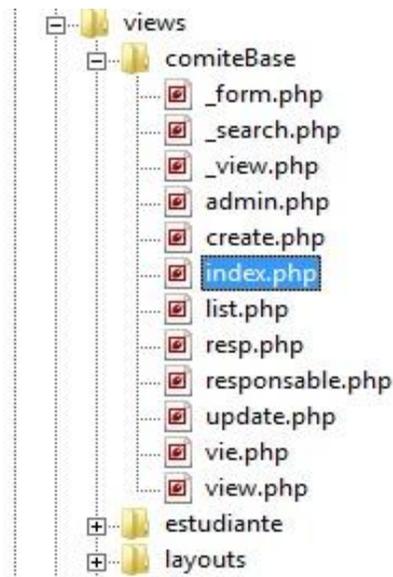


Figura 9: Clase vista comité de base.

```
<?php
$this->breadcrumbs=array(
    'Comité de Bases',
);

$this->menu=array(
// array('label'=>'Crear Comité de Bases', 'url'=>array('create', 'id'=>)),
array('label'=>'Administrar Comité de Bases', 'url'=>array('admin')),
);
?>
```

Figura 10: Acción index.php de la clase comité de base.

#### 4.2.2.3 Indentación

La indentación debe ser de 4 espacios sin caracteres de tabulación.

**Ejemplo:**

Subsistema para la gestión del ID2 de la UJC de la UCI.

```
13 |
14 |     <div class="row">
15 |         <?php echo $form->labelEx($model, 'nombre'); ?>
16 |         <?php echo $form->textField($model, 'nombre', array('size'=>30, 'maxlength'=>
17 |         30)); ?>
18 |         <?php echo $form->error($model, 'nombre'); ?>
19 |     </div>
20 |
21 |     <div class="row">
22 |         <?php echo $form->labelEx($model, 'activo'); ?>
23 |         <?php echo $form->checkBox($model, 'activo'); ?>
24 |         <?php echo $form->error($model, 'activo'); ?>
25 |     </div>
26 |
-- |
```

Figura 11: Ejemplo de indentación.

#### 4.2.2.4 Comentarios

Los comentarios deben ser (*/\* \*/* o *//*). Estos deben ser elaborados con un lenguaje formal y siempre teniendo en cuenta la ortografía, además deben de ser simples y directos.

**Ejemplo:**

```
-- |
83 |     /*Crea un militante desde el comite de base adicionandolo a la base de datos.
84 |     */
85 |     public function actionCreate($id)
86 |     {
87 |         $model=new Militante;
88 |         // Uncomment the following line if AJAX validation is needed
89 |         // $this->performAjaxValidation($model);
90 |
91 |         if(isset($_POST['Militante']))
92 |         {
```

Figura 12: Ejemplo de comentario.

#### 4.2.2.5 Definición de funciones

En las declaraciones de función, el bracket de apertura empieza debajo de la línea de definición de la función.



## Capítulo 4: Implementación y Prueba.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

#### Ejemplo:

```
84 public function actionCreate($id)
85 {
86     $model=new Militante;
87
88
89     if(isset($_POST['Militante']))
90     {
91         $model->attributes=$_POST['Militante'];
92         if($model->save())
93             $this->redirect(array('view','id'=>$model->id_militante));
94     }
95     $comitebase = ComiteBase::model()->findByPk($id);
96     $model->id_comitebase = $id;
97
```

Figura 13: Definición de función.

#### 4.2.2.6 Llamadas de función

Las funciones son llamadas sin espacios en el nombre de la función, al final lleva punto y coma.  
Los parámetros se separan por comas y sin espacios entre ellos.

#### Ejemplo:

```
2 <div class="row">
3     <?php echo $form->labelEx($model,'sexo'); ?>
4     <?php echo $form->dropDownList($model,'sexo', Militante::sexos(), array('
size'=>1,'maxlength'=>1)); ?>
5     <?php echo $form->error($model,'sexo'); ?>
```

Figura 14: Llamada de función.

#### 4.2.2.7 Llaves de apertura y cierre

El uso de las llaves se realizará en una nueva línea e individual, los bloques de código deben estar siempre confinados por llaves, excepto en los casos en que estén conformados por una



## Capítulo 4: Implementación y Prueba.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

sola línea. Las llaves siempre estarán al mismo nivel de la sentencia de código de la que proceden.

#### Ejemplo:

```
public function actionUpdate($id)
{
    $model=$this->loadModel($id);

    if(isset($_POST['Militante']))
    {
        $model->attributes=$_POST['Militante'];
        if($model->save())
            $this->redirect(array('view','id'=>$model->id_militante));
    }

    $this->render('update',array(
        'model'=>$model,
    ));
}
```

Figura 15: Apertura y cierre de llaves.

### 4.3 Fase de Implementación

En las iteraciones se implementa cada una de las HU previstas para estas. Como parte de este plan se descomponen las HU en tareas de desarrollo (Tareas de Ingeniería). Asignándole a cada tarea el responsable de su implementación. Las tareas pueden escribirse utilizando un lenguaje técnico y no necesariamente deben de ser entendidas por el cliente.

De acuerdo a la planificación realizada, se llevaron a cabo tres iteraciones de desarrollo en el sistema. Cada una de las Historias de Usuario comprendidas en las 3 iteraciones, fueron desglosadas en tareas de ingeniería (Tabla 4.1) desarrolladas por el programador.

| Historia de Usuarios | Tareas   |
|----------------------|--|
| Generar cómputo ID2. | Creación de la interfaz para la visualización de los datos.<br>Implementación de la lógica para conformar el reporte.<br>Implementación para la carga de los datos que |

## Capítulo 4: Implementación y Prueba.

### Subsistema para la gestión del ID2 de la UJC de la UCI.



|   |  |
|---|--|
|   | conforman el reporte.  |
| Dar alta a un militante.                                  | Creación de una interfaz para la captura de los datos.<br>Creación de una interfaz para la visualización de los datos.<br>Implementación de la lógica para dar alta a un militante en un comité de base.   |
| Dar baja a un militante.                                  | Creación de una interfaz para la captura de los datos.<br>Creación de una interfaz para la visualización de los datos.<br>Implementación de la lógica para dar baja a un militante en un comité de base.   |
| Exportar el modelo del ID2 a diferentes formatos (Excel). | Creación de la interfaz para la visualización de los datos.<br>Implementación de la lógica para exportar los datos a diferentes formatos (Excel).<br>Implementación de la lógica para la carga de los datos a exportar.  |
| Gestionar Comités de base.                                | Creación de una interfaz para la captura de los datos.<br>Creación de una interfaz para la visualización de los datos.<br>Implementación de la lógica para conformar los comités de base.<br>Implementación de la lógica para modificar los comités de base.<br>Implementación de la lógica para desintegrar los comités de base.<br>Implementación de la lógica para buscar y listar los comités de base.<br>Implementación de la lógica para adicionar los |

## Capítulo 4: Implementación y Prueba.

### Subsistema para la gestión del ID2 de la UJC de la UCI.



|   |   |
|---|---|
|   | militantes a un comité de base.   |
| Consumir servicio web.                                    | Creación de una interfaz para la carga de los datos de diferentes servicios.<br>Creación de una interfaz para la visualización de los datos.<br>Implementación de la lógica para la carga de datos de los diferentes servicios web.   |
| Gestionar militante.                                      | Creación de una interfaz para la captura de los datos.<br>Creación de una interfaz para la visualización de los datos.<br>Implementación de la lógica para insertar un militante.<br>Implementación de la lógica para eliminar un militante.<br>Implementación de la lógica para modificar un militante.<br>Implementación de la lógica para buscar un militante. |
| Generar Plantilla de control del Estado de la Militancia. | Creación de la interfaz para la visualización de los datos.<br>Implementación de la lógica para conformar el reporte.<br>Implementación para la carga de los datos que conforman el reporte.  |
| Gestionar comité primario.                                | Creación de una interfaz para la captura de los datos.<br>Creación de una interfaz para la visualización de los datos.<br>Implementación de la lógica para conformar los comités de primario.<br>Implementación de la lógica para modificar los   |

*Subsistema para la gestión del ID2 de la UJC de la UCI.*

|   |  |
|---|--|
|   | <p>comités de primario.</p> <p>Implementación de la lógica para desintegrar los comités de primario.</p> <p>Implementación de la lógica para buscar y listar los comités de primario.</p> <p>Implementación de la lógica para adicionar los militantes a un comité primario.</p> |
| Asignar responsabilidad UJC a un militante.   | <p>Creación de una interfaz para la captura de los datos.</p> <p>Creación de una interfaz para la visualización de los datos.</p> <p>Implementación de la lógica para asignar una responsabilidad.</p> <p>Implementación de la lógica para modificar una responsabilidad.</p>    |
| Listar integrantes de un comité de base.      | <p>Creación de la interfaz que visualice los datos.</p> <p>Programación de la lógica para listar los militantes que conforman el comité.</p>   |
| Listar universo juvenil.                      | <p>Creación de una interfaz que permita la visualización de los datos.</p> <p>Programación de la lógica para listar los militantes.</p>  |
| Listar responsables de un comité de base.     | <p>Creación de una interfaz que permita la visualización de los datos.</p> <p>Programación de la lógica para mostrar los responsables.</p>   |
| Mostrar Expediente del militante.             | <p>Creación de una interfaz para la visualización de los datos.</p> <p>Programación de la lógica para mostrar los datos referentes al militante.</p>   |
| Listar militantes con edad de arribar el PCC. | <p>Creación de una interfaz para la visualización de los datos.</p>  |

## Capítulo 4: Implementación y Prueba.

### Subsistema para la gestión del ID2 de la UJC de la UCI.



|  |   |
|--|---|
|  | Programación de la lógica para mostrar los datos referentes al militante. |
|--|---|

Tabla 13: Tareas de ingeniería por HU.

En los epígrafes siguientes se detallarán cada una de las Historias de Usuario que comprenden el desarrollo del sistema, durante las tres iteraciones previstas. Para ello se hará énfasis en cada una de las tareas comprendidas por Historia de Usuario. Para ver el resto de las HU y tareas que la comprenden ver anexo 6.

#### 4.3.1 Iteración 1.

| Historia de Usuario   | Estimación | Real |
|---|------------|------|
| Generar el Plantillas de Control del Estado de la Militancia. | 2 ½        | 2 ½  |
| Generar Cómputo del ID2.                                      | 3          | 3    |
| Dar alta a un militante en un comité de base.                 | 2          | 1    |
| Dar Baja a un militante en un comité de base.                 | 2          | 1    |
| Gestionar comité primario.                                    | 1          | 1    |
| Gestionar comité de base.                                     | 1          | 1    |
| Gestionar militante.  | 1          | 1    |
| Totales   | 12½        | 10 ½ |

Tabla 14: HU comprendidas en la iteración 1.

| Tarea de ingeniería  |   |
|--|---|
| <b>Número tarea: 1</b>   | Historia de usuario: Generar cómputo ID2. |
| <b>Nombre de tarea:</b> Programación de la lógica para conformar el reporte.             |   |
| <b>Tipo de tarea:</b> Desarrolladora   | Puntos estimados: 2 1/2                   |
| <b>Programador responsable:</b> Yilen Yamilé Reyes Montejo.                              |   |
| <b>Descripción:</b> Se conformarán las clases y métodos que permitan generar el reporte. |   |

## Capítulo 4: Implementación y Prueba.

### Subsistema para la gestión del ID2 de la UJC de la UCI.



Tabla 15: Tarea #1 de la HU Generar cómputo del ID2.

| Tarea de ingeniería   |   |
|---|---|
| <b>Número tarea: 2</b>  | Historia de usuario: Generar cómputo ID2. |
| <b>Nombre de tarea:</b> Programación para la carga de los datos que conforman el reporte.                                   |   |
| <b>Tipo de tarea: Desarrolladora</b>  | Puntos estimados: 2 1/2                   |
| <b>Programador responsable:</b> Yilen Yamilé Reyes Montejo.   |   |
| <b>Descripción:</b> Se conformarán las clases que permitan la conexión a la base de datos para cargar los datos necesarios. |   |

Tabla 16: Tarea #2 de la HU Generar cómputo del ID2.

| Tarea de ingeniería   |   |
|---|---|
| <b>Número tarea: 3</b>  | Historia de usuario: Generar cómputo ID2. |
| <b>Nombre de tarea:</b> Creación de la interfaz para la visualización de los datos.       |   |
| <b>Tipo de tarea: Desarrolladora</b>  | Puntos estimados: 2 1/2                   |
| <b>Programador responsable:</b> Yilen Yamilé Reyes Montejo.                               |   |
| <b>Descripción:</b> Se conformarán las clases que permitan la visualización de los datos. |   |

Tabla 17: Tarea #3 de la HU Generar cómputo del ID2.

| Tarea de ingeniería  |  |
|--|--|
| <b>Número tarea: 1</b>   | <b>Historia de usuario:</b> Dar alta a un militante. |
| <b>Nombre de tarea:</b> Programación de la lógica para dar alta a un militante en un comité de base.               |  |
| <b>Tipo de tarea: Desarrolladora</b>   | Puntos estimados: 2                                  |
| <b>Programador responsable:</b> Yilen Yamilé Reyes Montejo.  |  |
| <b>Descripción:</b> Se conformarán las clases y métodos que permitan dar alta a un militante en un comité de base. |  |

Tabla 18: Tarea #1 de la HU Dar alta a un militante.

| Tarea de ingeniería    |  |
|------------------------|--|
| <b>Número tarea: 2</b> | <b>Historia de usuario:</b> Dar alta a un militante. |

## Capítulo 4: Implementación y Prueba.

### Subsistema para la gestión del ID2 de la UJC de la UCI.



|  |                     |
|--|---------------------|
| <b>Nombre de tarea:</b> Creación de una interfaz para la captura de los datos.                     |                     |
| <b>Tipo de tarea:</b> Desarrolladora   | Puntos estimados: 2 |
| <b>Programador responsable:</b> Yilen Yamilé Reyes Montejo.  |                     |
| <b>Descripción:</b> Se crean las interfaces que permitan la entrada de los datos para dar el alta. |                     |

Tabla 19: Tarea #2 de la HU Dar alta a un militante.

| Tarea de ingeniería   |  |
|---|--|
| <b>Número tarea:</b> 3  | <b>Historia de usuario:</b> Dar alta a un militante. |
| <b>Nombre de tarea:</b> Creación de una interfaz para la visualización de los datos.      |  |
| <b>Tipo de tarea:</b> Desarrolladora  | Puntos estimados: 2                                  |
| <b>Programador responsable:</b> Yilen Yamilé Reyes Montejo.                               |  |
| <b>Descripción:</b> Se conformarán las clases que permitan la visualización de los datos. |  |

Tabla 20: Tarea #3 de la HU Dar alta a un militante.

#### 4.3.2 Iteración 2

| Historia de Usuario                                   | Estimación | Real |
|---|------------|------|
| Exportar modelo del ID2 en diferentes formatos (XLS). | 1          | 1    |
| Asignar responsabilidad UJC a un militante.           | 1          | ½    |
| Consumir de los servicios Web.                        | 2 ½        | 2    |
| Mostrar expediente de un militante.                   | 1          | 1    |
| Totales   | 5 ½        | 4 ½  |

Tabla 21: HU comprendidas en la iteración 2.

| Tarea de ingeniería  |   |
|--|---|
| <b>Número tarea:</b> 1   | <b>Historia de usuario:</b> Exportar el modelo del ID2 a diferentes formatos (Excel). |
| <b>Nombre de tarea:</b> Programación de la lógica para exportar los datos a diferentes formatos (Excel). |   |

## Capítulo 4: Implementación y Prueba.

### Subsistema para la gestión del ID2 de la UJC de la UCI.



|   |                     |
|---|---------------------|
| <b>Tipo de tarea: Desarrolladora</b>  | Puntos estimados: 1 |
| <b>Programador responsable:</b> Yilen Yamilé Reyes Montejo.   |                     |
| <b>Descripción:</b> Se conformarán las clases que permitan la conexión a la base de datos para cargar los datos necesarios. |                     |

Tabla 22: Tarea #1 de la HU Exportar a (Excel).

| Tarea de ingeniería   |   |
|---|---|
| <b>Número tarea: 2</b>  | <b>Historia de usuario:</b> Exportar el modelo del ID2 a diferentes formatos (Excel). |
| <b>Nombre de tarea:</b> Programación de la lógica para la carga de los datos a exportar.                  |   |
| <b>Tipo de tarea: Desarrolladora</b>  | Puntos estimados: 1   |
| <b>Programador responsable:</b> Yilen Yamilé Reyes Montejo.   |   |
| <b>Descripción:</b> Se implementarán las clases para transformar los datos cargados al formato requerido. |   |

Tabla 23: Tarea #2 de la HU Exportar a (Excel).

| Tarea de ingeniería   |   |
|---|---|
| <b>Número tarea: 3</b>  | <b>Historia de usuario:</b> Exportar el modelo del ID2 a diferentes formatos (Excel). |
| <b>Nombre de tarea:</b> Creación de la interfaz para la visualización de los datos.                                     |   |
| <b>Tipo de tarea: Desarrolladora</b>  | Puntos estimados: 1   |
| <b>Programador responsable:</b> Yilen Yamilé Reyes Montejo.   |   |
| <b>Descripción:</b> Se conformarán las clases que permitan la visualización de los datos en los formatos especificados. |   |

Tabla 24: Tarea #3 de la HU Exportar a (Excel).

| Tarea de ingeniería   |  |
|---|--|
| <b>Número tarea: 1</b>  | <b>Historia de usuario:</b> Consumir servicio web. |
| <b>Nombre de tarea:</b> Programación de la lógica para la carga de datos de los diferentes servicios web. |  |
| <b>Tipo de tarea: Desarrolladora</b>  | Puntos estimados: 2                                |
| <b>Programador responsable:</b> Yilen Yamilé Reyes Montejo.   |  |

## Capítulo 4: Implementación y Prueba.

### Subsistema para la gestión del ID2 de la UJC de la UCI.



**Descripción:** Se crea las clases y métodos necesarios para acceder a los servicios web existentes en la universidad, tomar los datos publicados de los militantes e ingresarlos a la base de datos.

Tabla 25: Tarea #1 de la HU Consumir servicio web.

| Tarea de ingeniería  |  |
|--|--|
| <b>Número tarea: 2</b>   | <b>Historia de usuario:</b> Consumir servicio web. |
| <b>Nombre de tarea:</b> Creación de una interfaz para la carga de los datos de diferentes servicios.                                       |  |
| <b>Tipo de tarea: Desarrolladora</b>   | Puntos estimados: 2                                |
| <b>Programador responsable:</b> Yilen Yamilé Reyes Montejo.  |  |
| <b>Descripción:</b> Se conformarán las clases y métodos que permitan la entrada de algún criterio de búsqueda para consultar al militante. |  |

Tabla 26: Tarea #2 de la HU Consumir servicio web.

| Tarea de ingeniería   |  |
|---|--|
| <b>Número tarea: 3</b>  | <b>Historia de usuario:</b> Consumir servicio web. |
| <b>Nombre de tarea:</b> Creación de una interfaz para la visualización de los datos.      |  |
| <b>Tipo de tarea: Desarrolladora</b>  | Puntos estimados: 2                                |
| <b>Programador responsable:</b> Yilen Yamilé Reyes Montejo.                               |  |
| <b>Descripción:</b> Se conformarán las clases que permitan la visualización de los datos. |  |

Tabla 27: Tarea #3 de la HU Consumir servicio web.

#### 4.3.3 Iteración 3

| Historia de Usuario                           | Estimación | Real |
|---|------------|------|
| Listar militantes con edad de arribar el PCC. | ½          | ½    |
| Listar universo juvenil.                      | ½          | ½    |
| Buscar responsable de un comité de base.      | ½          | ½    |
| Generar reporte de cotización.                | ½          | ½    |



|         |   |   |
|---------|---|---|
| Totales | 2 | 2 |
|---------|---|---|

Tabla 28: HU comprendidas en la iteración 3.

| Tarea de ingeniería  |   |
|--|---|
| <b>Número tarea: 1</b>   | <b>Historia de usuario:</b> Listar militantes con edad de arribar al PCC. |
| <b>Nombre de tarea:</b> Programación de la lógica para listar los militantes.                              |   |
| <b>Tipo de tarea: Desarrolladora</b>   | Puntos estimados: 1/2   |
| <b>Programador responsable:</b> Yilen Yamilé Reyes Montejo.  |   |
| Descripción: Se crean las clases y métodos que permitan obtener los militantes con edad de arribar al PCC. |   |

Tabla 29: Tarea #1 de la HU Listar militantes con edad de arribar al PCC.

| Tarea de ingeniería  |   |
|--|---|
| <b>Número tarea: 2</b>   | <b>Historia de usuario:</b> Listar militantes con edad de arribar al PCC. |
| <b>Nombre de tarea:</b> Creación de una interfaz para la visualización de los datos. |   |
| <b>Tipo de tarea: Desarrolladora</b>   | Puntos estimados: 1/2   |
| <b>Programador responsable:</b> Yilen Yamilé Reyes Montejo.                          |   |
| <b>Descripción:</b> Se crea una interfaz que permita visualizar los datos.           |   |

Tabla 30: Tarea #2 de la HU Listar militantes con edad de arribar al PCC.

#### 4.4 Pruebas

En el desarrollo del software un instrumento capaz de medir la calidad de un producto son las pruebas. Estas son dirigidas hacia el sistema de software en general; con el objetivo de medir si la aplicación cumple las funcionalidades establecidas por el cliente. Verificando al mismo tiempo la calidad, usabilidad o posibles errores en la implementación del producto.

En la metodología XP el desarrollo de software es dirigido por la realización de pruebas tanto como sea posible, mediante la culminación de las iteraciones. Permitiendo un aumento en la calidad del producto. Reduciendo el número de errores no detectados por el equipo de desarrollo.

## Capítulo 4: Implementación y Prueba.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

Para probar el subsistema se definió una estrategia de pruebas conformada por dos niveles de pruebas:

- ✓ El primero constituido por las pruebas unidad, mediante la utilización del método de caja blanca a través de la técnica de camino básico.
- ✓ El segundo nivel está conformado por pruebas de integración, mediante el método de caja negra y usando la técnica de partición de equivalencia.

#### 4.4.1 Pruebas unitarias

Las pruebas unitarias son llevadas a cabo por los programadores. Asegurando que cada módulo cumpla con un comportamiento esperado. Se utilizan luego de realizar modificaciones en un módulo, para verificar entradas y condiciones inusuales. Cada desarrollador tiene que ir probando constantemente lo que va obteniendo en el transcurso de la implementación de un sistema, para garantizar que las funcionalidades exigidas por el cliente estén siendo implementadas correctamente.

En las pruebas de unidad se utilizó la técnica de camino básico. A través de esta técnica se examinan los caminos lógicos del software, con el fin de descubrir errores no encontrados hasta el momento. Se inspecciona el estado del programa en varios puntos, para determinar si el estado real coincide con el esperado. Se garantiza que con la técnica de camino básico durante la prueba, se ejecuta por lo menos, una vez cada sentencia del programa. Cualquier representación del diseño procedimental se puede traducir a un grafo de flujo o grafo del programa (27).

Además se calcula la **complejidad cliclomática**, denotada como  $V(G)$ , la cual se define de tres formas diferentes y todas deben arrojar el mismo resultado.

1. El número de regiones (R) del grafo de flujo es igual a  $V(G)$ .
2. La cantidad de aristas (A) del grafo menos los nodos (N) más 2 es igual a  $V(G)$ . es decir  $V(G) = A - N + 2$ .
3. Los nodos predicados (P) más uno es igual a  $V(G)$  (un nodo predicado es el que representa una estructura condicional (if o case), es decir, que de él salen varios caminos).  $V(G) = P + 1$ .

## Capítulo 4: Implementación y Prueba.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

**Nodos:** Cada círculo representado se denomina nodo del Grafo de Flujo, el cual representa una o más secuencias procedimentales. Un solo nodo puede corresponder a una secuencia de procesos o a una sentencia de decisión. Puede ser también que hallan nodos que no se asocian, se utilizan principalmente al inicio y final del grafo.

**Aristas:** Las flechas del grafo se denominan aristas y representan el flujo de control, son análogas a las representadas en un diagrama de flujo. Una arista debe terminar en un nodo, incluso aunque el nodo no represente ninguna sentencia procedimental.

**Regiones:** Las regiones son las áreas delimitadas por las aristas y nodos. También se incluye el área exterior del grafo, contando como una región más. Las regiones se enumeran y la cantidad de regiones es equivalente a la cantidad de caminos independientes del conjunto básico de un programa (28).

Las pruebas unitarias se desarrollaban cada vez que se terminaba de implementar alguna funcionalidad, probándose directamente con datos reales. Brindándole al programador una retroalimentación de cómo está realizando su trabajo.

A continuación se muestra un ejemplo de las pruebas unitarias realizadas al subsistema.

```
private function cargarWebservice($ci)
{
    $estudiante = null; 1

    $client = new SoapClient('http://akademos2.uci.cu/servicios/v4/AkademosWS.wsdl'); 1
    $estudiante = $client->ObtenerEstudianteDadoCI($ci); 1

    if($estudiante != null)1{
        $client2 = new SoapClient('http://identificacion.uci.cu/servicios/v5/servicios.php?wsdl'); 2
        $foto = $client2->ObtenerFotoDadoIdExpediente($estudiante->IdExpediente); 2

        $estudiante->foto = $foto; 2
    }
    return $estudiante; 3
} 4
```

Figura 16 Prueba de unidad para el método cargarWebservice.

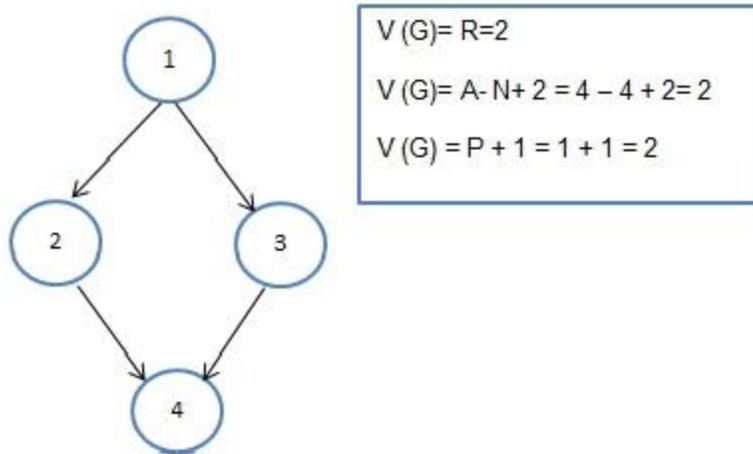


Figura 17 Grafo de Flujo para el método cargarWebservice.

**Caso de prueba para el método cargarWebservice.**

**Camino 1-2-4**

**Caso de prueba:** Cargar web service.

**Entrada:** Para el carnet de identidad válido (85070928104)

**Resultado:** Se muestran los datos del militante.

**Camino 1-3-4**

**Caso de prueba:** Cargar web service.

**Entrada:** Para un carnet de identidad no válido (8507092810)

**Resultado:** Se envía el mensaje ("No se encontraron resultados para el CI.").



## Capítulo 4: Implementación y Prueba.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

```
public function actionDelete($id)
{
    if(Yii::app()->request->isPostRequest) 1
    {
        $this->loadModel($id)->delete(); 2

        if(!isset($_GET['ajax'])) 2
            $this->redirect(isset($_POST['returnUrl']) ? $_POST['returnUrl'] : array(
                'admin')); 3
    } 4
    else 5
        throw new CHttpException(400,'Invalid request. Please do not repeat this request again. ');5
    } 6
}
```

Figura 18 Prueba unidad del método Delete (Eliminar comité de base).

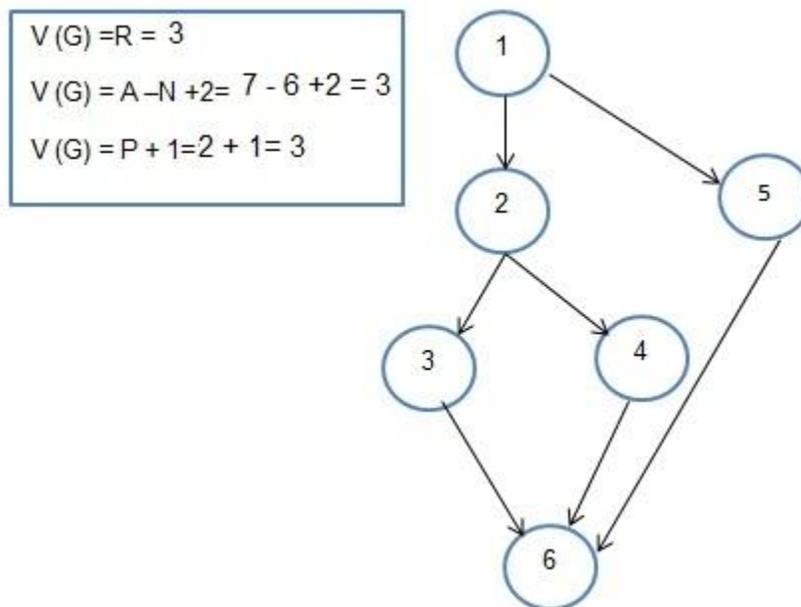


Figura 19: Grafo de Flujo para el método Delete.

**Caso de prueba para el método eliminar comité de base.**

**Camino** 1-2-3-6

**Caso de prueba:** Eliminar comité de base.

**Entrada:** Para el id (1)

**Resultado:** Se elimina el comité de base.



## Capítulo 4: Implementación y Prueba.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

#### Camino 1-2-4-6

**Caso de prueba:** Eliminar comité de base.

**Entrada:** Se eliminó el comité de base.

**Resultado:** Redirecciona hacia la página del comité primario al que pertenecía el comité de base eliminado.

#### Camino 1-5-6

**Caso de prueba:** Eliminar comité de base.

**Entrada:** Para el id (1).

**Resultado:** Muestra el mensaje de error (“El comité de base ya ha sido eliminado.”).

```
public function actionCreate($id)
{
    $model=new Recorrido; 1
    $militante = Militante::model()->findByPk($id); 1

    if($militante->activo) 1
        throw new CHttpException(404,'El militante ya pertenece a un comité de base.
. '); 2
    if(isset($_POST['Recorrido'])) 3
    {
        $model->attributes=$_POST['Recorrido']; 4
        if($model->save()) 4{

            $militante->activo=true; 5
            $militante->id_comitebase = $model->id_comite_base; 5
            if($militante->save(false)) 5
            {
                Yii::app()->user->setFlash('success', 'Se ha realizado la operaci&
oacute;n correctamente. '); 6
                $this->redirect(array('view','id'=>$model->id_recorrido)); 6
            }else 7
            {
                Yii::app()->user->setFlash('error', 'Ha ocurrido un error. '); 7
            } 8
        } 9
        $militante = Militante::model()->findByPk($id); 9
        $model->id_militante = $id; 9

        $this->render('create',array( 9
            'model'=>$model,
            'Militante'=>$militante,
            'ComiteBase'=>false,));
    } 10
}
```

Figura 20 Prueba de unidad para el método Create (Adicionar recorrido).

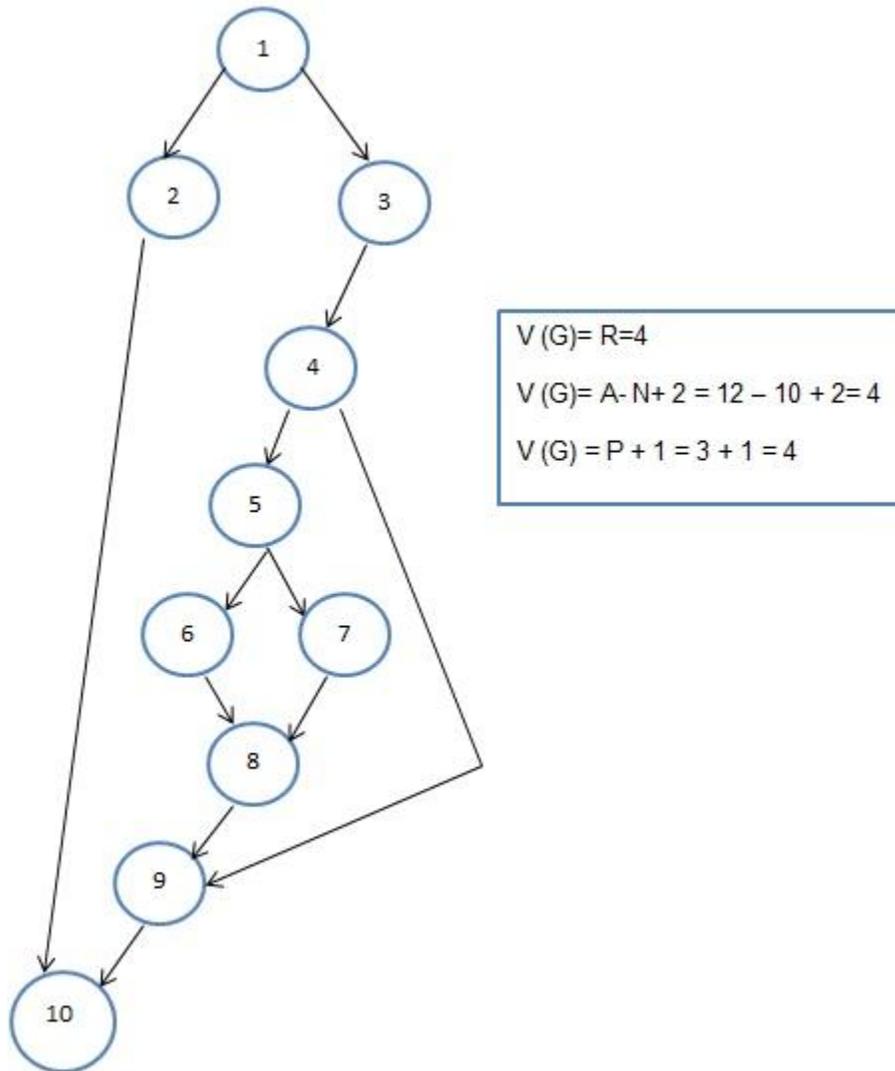


Figura 21 Grafo de Flujo para el método Create.

**Caso de prueba para el método adicionar recorrido.**

**Camino:** 1-2-10

**Caso de prueba:** Adicionar recorrido.

**Entrada:** Para el id del militante (1)

**Resultado:** Muestra el mensaje de error (“El militante ya pertenece a un comité de base.”).

**Camino:** 1-3-4-5-6-8-9-10

**Caso de prueba:** Adicionar recorrido.

## Capítulo 4: Implementación y Prueba.

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

**Entrada:** Para el id del militante (2).

**Resultado:** Adiciona el recorrido y muestra el mensaje (“Se ha realizado la operación correctamente”).

**Camino:** 1-3-4-5-7-8-9-10

**Caso de prueba:** Adicionar recorrido.

**Entrada:** Para el id del militante (2). En este caso no se salvaron todos los datos correctamente para el recorrido.

**Resultado:** Muestra el mensaje (“Ha ocurrido un error”).

**Camino:** 1-3-4-9-10

**Caso de prueba:** Adicionar recorrido.

**Entrada:** Para el id del militante (2) No introdujo todos los datos para adicionar el nuevo recorrido.

**Resultado:** Redirecciona a la interfaz de adicionar recorrido y muestra el mensaje (“Faltan campos requeridos.”).

#### 4.4.2 Pruebas de integración

En el caso de XP se utilizan las pruebas de integración para determinar si al final de cada iteración se implementan correctamente las funcionalidades requeridas, además de comprobar que dicha funcionalidad sea la que necesita el cliente. Permitiéndole al programador conocer qué es lo que resta por hacer. Las pruebas de integración son pruebas de caja negra que se crean a partir de las historias de usuario (29). El objetivo final de estas, es garantizar que los requerimientos han sido cumplidos y que el sistema está listo para ser usado por el cliente. Marcando el camino a seguir por los desarrolladores e indicando las funcionalidades más relevantes (30).

A continuación se detallan las distintas pruebas de integración empleadas para constatar el correcto funcionamiento de las HU previstas en cada una de las iteraciones. Los restantes casos de pruebas se encuentran en el anexo 7.

#### Prueba de integración

## Capítulo 4: Implementación y Prueba.

### Subsistema para la gestión del ID2 de la UJC de la UCI.



|   |  |
|---|--|
| <b>Código: 1</b>  | <b>Historia de usuario:</b> Consumir Servicio Web. |
| <b>Nombre:</b> CP <sup>2</sup> Consumir servicio web.   |  |
| <b>Descripciones:</b> Prueba funcional consumir servicio web.   |  |
| <b>Condición de ejecución:</b> El usuario debe estar autenticado y poseer el rol de documentador.   |  |
| <b>Entrada/Pasos de ejecución:</b> 1- El usuario selecciona la opción de adicionar un nuevo militante.<br>2- Introduce el criterio de búsqueda (CI).<br>3- El usuario Presiona la opción de cargar datos. |  |
| <b>Resultados esperados:</b> Se muestran los datos del militante.   |  |
| <b>Evaluación de la prueba:</b> Satisfactoria.  |  |

Tabla 31: Prueba de integración #1 de la HU Consumir servicio web.

| Prueba de integración   |  |
|---|--|
| <b>Código: 2</b>  | <b>Historia de usuario:</b> Generar Cómputo del ID2. |
| <b>Nombre:</b> CP Generar cómputo del ID2.  |  |
| <b>Descripciones:</b> Prueba funcional generar cómputo del ID2.   |  |
| <b>Condición de ejecución:</b> El usuario debe estar autenticado y poseer el rol de documentador.   |  |
| <b>Entrada/Pasos de ejecución:</b> 1- El usuario selecciona la opción de ver un comité primario.<br>2- Se muestran los datos del comité primario.<br>3- El usuario presiona la opción de Generar cómputo del ID2. |  |
| <b>Resultados esperados:</b> Se muestra una nueva vista con el cómputo del comité primario.   |  |
| <b>Evaluación de la prueba:</b> Satisfactoria.  |  |

Tabla 32: Prueba de integración #1 de la HU Generar cómputo del ID2.

| Prueba de integración   |   |
|---|---|
| <b>Código: 3</b>  | <b>Historia de usuario:</b> Dar alta a un militante en un comité de base. |
| <b>Nombre:</b> CP Dar alta a un militante en un comité de base. |   |

<sup>2</sup> Caso de prueba.

## Capítulo 4: Implementación y Prueba.

### Subsistema para la gestión del ID2 de la UJC de la UCI.



|   |
|---|
| <b>Descripciones:</b> Prueba funcional dar alta a un militante en un comité de base.  |
| <b>Condición de ejecución:</b> 1- El usuario debe estar autenticado y poseer el rol de documentador.<br>2- El militante no puede pertenecer a ningún comité de base en ese momento.   |
| <b>Entrada/Pasos de ejecución:</b> 1- El usuario selecciona la opción de ver un militante.<br>2- Se muestran los datos del militante.<br>3- El usuario presiona la opción de dar alta.<br>4- El usuario introduce los datos.<br>5- El usuario presiona el botón "Dar Alta". |
| <b>Resultados esperados:</b> Se muestra una nueva vista con los datos del nuevo movimiento.   |
| <b>Evaluación de la prueba:</b> Satisfactoria.  |

Tabla 33: Prueba de integración #1 de la HU Dar alta a un militante en un comité de base.

| Prueba de integración   |   |
|---|---|
| <b>Código:</b> 4  | <b>Historia de usuario:</b> Dar baja a un militante en un comité de base. |
| <b>Nombre:</b> CP Dar baja a un militante en un comité de base.   |   |
| <b>Descripciones:</b> Prueba funcional dar baja a un militante en un comité de base.  |   |
| <b>Condición de ejecución:</b> 1- El usuario debe estar autenticado y poseer el rol de documentador.<br>2- El militante tiene que pertenecer a un comité de base en ese momento.  |   |
| <b>Entrada/Pasos de ejecución:</b> 1- El usuario selecciona la opción de ver un militante.<br>2- Se muestran los datos del militante.<br>3- El usuario presiona la opción de dar baja.<br>4- El usuario introduce los datos.<br>5- El usuario presiona el botón "Dar Baja". |   |
| <b>Resultados esperados:</b> Se muestra una nueva vista con los datos del nuevo movimiento ocurrido.  |   |
| <b>Evaluación de la prueba:</b> Satisfactoria.  |   |

Tabla 34: Prueba de integración #1 de la HU Dar baja a un militante en un comité de base.

### Prueba de integración

|  |  |
|--|--|
| <b>Código: 8</b>   | <b>Historia de usuario:</b> Gestionar comité primario. |
| <b>Nombre:</b> CP Insertar comité de base.   |  |
| <b>Descripciones:</b> Prueba funcional eliminar comité primario.   |  |
| <b>Condición de ejecución:</b> 1- El usuario debe estar autenticado y poseer el rol de documentador.   |  |
| <b>Entrada/Pasos de ejecución:</b> 1- El usuario selecciona la opción de ver un comité primario.<br>2- Se muestran los datos del comité primario.<br>3- El usuario selecciona la opción de adicionar un comité base. |  |
| <b>Resultados esperados:</b> Ver CP crear comité de base.  |  |
| <b>Evaluación de la prueba:</b> Satisfactoria.   |  |

Tabla 35: Prueba de integración #1 de la HU Gestionar comité primario.

#### 4.5 Conclusiones

Con el resultado del trabajo realizado y una vez terminadas las iteraciones propuestas, se cumplieron los objetivos trazados para la realización del subsistema, se reflejó el estándar de codificación utilizado, permitiendo un mejor entendimiento del código. Se especificaron las tareas a desarrollar por el equipo, como parte de la implementación de cada una de las HU. Se realizaron las pruebas de unidad e integración al subsistema, evidenciando la funcionalidad del producto final.

## *Conclusiones Generales.*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

#### **Conclusiones Generales**

Se culmina el presente trabajo, donde se ha logrado cumplir los objetivos trazados con las técnicas y la metodología para cada uno de ellos. Se aborda acerca del proceso de desarrollo del subsistema para la gestión del ID2 de la UJC de la UCI, que forma parte del Sistema de Información para la UJC de la UCI. Luego de la investigación desarrollada, se puede concluir los con siguientes resultados:

- ✓ Se formalizó un estudio acerca de los procesos que se realizan dentro del Departamento de Documentación y Control, en específico, los movimientos de las altas y bajas que se realizan, la gestión de información del ID2, el universo juvenil y las responsabilidades que un militante puede tener dentro de una organización de base.
- ✓ Se fundamentó la selección de la metodología, lenguaje de modelado, tecnologías y herramientas para el desarrollo del subsistema.
- ✓ Se desarrolló un estudio con respecto a los sistemas de gestión similares que se han realizados para la gestión de la información de la UJC.
- ✓ Se obtuvo como resultado un subsistema que provee un mecanismo para la gestión de la información del ID2 de la UJC en la UCI, de manera tal que al usuario le sea fácil interactuar con él y gane en tiempo en el desarrollo de las tareas a realizar.
- ✓ Se definió una estrategia de pruebas que comprende los niveles unidad y de integración que permitieron probar las funcionalidades del sistema.

*Recomendaciones.*

*Subsistema para la gestión del ID2 de la UJC de la UCI.*

## **Recomendaciones**

Como resultado final del proceso investigativo, se obtuvo una serie de recomendaciones a tener en cuenta para desarrollos futuros que den continuidad al proceso. A continuación se detallan las de mayor relevancia.

- ✓ Trabajar en el desarrollo de nuevas funcionalidades que permitan una mayor usabilidad del sistema por ejemplo, incluir reportes de cotización para los trabajadores y comités de base.
- ✓ Mantener una constante interacción con los usuarios de la aplicación que propicie una retroalimentación, dando así margen a la corrección de posibles errores que no hayan sido localizados.
- ✓ Elaborar un manual de usuario para familiarizar al cliente en el manejo del subsistema.
- ✓ Realizar pruebas de sistema, con el objetivo de ver la respuesta del mismo, frente a distintas situaciones; así como pruebas de aceptación, para medir el nivel de cumplimiento de las necesidades del cliente.

*Subsistema para la gestión del ID2 de la UJC de la UCI.*

**Bibliografía**

1. Intranet2. *Intranet2*. [En línea] UCI. [Citado el: 21 de marzo de 2012.] [http://intranet2.uci.cu/sites/default/files/pdf\\_organizaciones/ESTATUTOS%20DE%20LA%20UNION%20DE%20JOVENES%20COMUNISTAS%20FEBRERO%202005.pdf](http://intranet2.uci.cu/sites/default/files/pdf_organizaciones/ESTATUTOS%20DE%20LA%20UNION%20DE%20JOVENES%20COMUNISTAS%20FEBRERO%202005.pdf).
2. UCI. *UCI*. [En línea] UCI, 2012. [Citado el: 15 de abril de 2012.] <http://www.uci.cu/mision>.
3. **Laguna, Lic. Omar Gómez Delgado y Lic. Yosmicel Mendoza.** GestioPolis. *GestioPolis*. [En línea] 1 de Enero de 2008. [Citado el: 2 de Diciembre de 2011.] <http://www.gestiopolis.com/administracion-estrategia/sistemas-de-gestion-como-estrategia-dentro-de-la-administracion.htm>.
4. Universidad de Alcalá | Protección de Datos de Carácter Personal. *Universidad de Alcalá | Protección de Datos de Carácter Personal*. [En línea] [Citado el: 08 de 06 de 2012.] [https://portal.uah.es/portal/page/portal/proteccion\\_datos/conceptos/tratamiento\\_datos](https://portal.uah.es/portal/page/portal/proteccion_datos/conceptos/tratamiento_datos).
5. Biblioteca virtual de las ciencias en Cuba. *Biblioteca virtual de las ciencias en Cuba*. [En línea] 3 de enero de 2012. [Citado el: 10 de febrero de 2012.] <http://www.bibliociencias.cu/gsd/collect/revistas/archives/HASH01f2/d6d2ce0b.dir/doc.pdf>.
6. BuenasTareas. *BuenasTareas*. [En línea] 2012. [Citado el: 10 de abril de 2012.] <http://www.buenastareas.com/materias/xp-metodologia-esta-es-una-de-las-metodolog%C3%ADas-de-desarrollo-de-software-m%C3%A1s-utilizada-en-la-actualidad/20>.
7. **ONess, Aplicación de eXtreme Programming en.** Oness. *Oness*. [En línea] [Citado el: 17 de marzo de 2012.] <http://oness.sourceforge.net/proyecto/html/ch05s02.html>.
8. Bizagi. *Bizagi*. [En línea] [Citado el: 20 de marzo de 2012.] <https://www.bizagi.com/docs/BPMNNbyExampleSPA.pdf>.
9. **Informática, Sub-Jefatura de.** *Herramientas CASE*. s.l. : Impreso en los Talleres de la Oficina de Impresiones de la Oficina Técnica de Difusión, 1999.
10. visual paradigm. *visual paradigm*. [En línea] [Citado el: 03 de mayo de 2012.] <http://www.programas-gratis.net/b/visual-paradigm>.
11. <http://www.chacharaselnido.com>. [En línea] [Citado el: 11 de abril de 2012.] [http://www.chacharaselnido.com/materias\\_ing\\_alvarez/apuntes\\_u1.pdf](http://www.chacharaselnido.com/materias_ing_alvarez/apuntes_u1.pdf).
12. Maestrosdelweb. *Maestrosdelweb*. [En línea] 2012. [Citado el: 12 de abril de 2012.] <http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>.
13. PostgreSQL. *PostgreSQL*. [En línea] 2 de Octubre de 2010. [Citado el: 07 de Diciembre de 2011.] [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).

## Glosario de Términos.

### Subsistema para la gestión del ID2 de la UJC de la UCI.

14. Buenastareas. *Buenastareas*. [En línea] [Citado el: 11 de abril de 2012.] <http://www.buenastareas.com/ensayos/Que-Es-Un-Ide-De-Programacion/163537.html>.
15. Argentirawarez. *Argentirawarez*. [En línea] [Citado el: 13 de abril de 2012.] <http://www.argentinawarez.com/programas-gratis/2160821-netbeans-ide-7-1-1-totalmente-full-1-link.html>.
16. Masadelante. *Masadelante*. [En línea] [Citado el: 13 de abril de 2012.] <http://www.masadelante.com/faqs/servidor>.
17. Tareas 4107 informaticas. [En línea] [Citado el: 15 de marzo de 2012.] <http://jose-sosa-95.blogspot.com/2012/03/plataformas.html>.
18. **C, Louden Kenneth**. Lenguajes de Programacion. *Principios y Practicas*. 2004.
19. PHP. *PHP*. [En línea] 14 de Enero de 2001-2012. [Citado el: 14 de Enero de 2012.] <http://www.php.net/>.
20. **Gutiérrez, Javier J.** [En línea] [Citado el: 09 de Enero de 2012.] [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf).
21. Yii Framework en Español. *Yii Framework en Español*. [En línea] 21 de Enero de 2009. [Citado el: 09 de Enero de 2012.] <http://yiiframework.wordpress.com/>.
22. Dzone. *Dzone*. [En línea] [Citado el: 21 de abril de 2012.] <http://php.dzone.com/news/top-10-php-frameworks>.
23. **Addiel Rodríguez Guadarrama, Thaymí Mazorra O’Farrill**. *SemanaTecnologica*. *SemanaTecnologica*. [En línea] 2011. [Citado el: 20 de marzo de 2012.] [semanatecnologica.fordes.co.cu/ocs-2.3.2/public/site/239.pdf](http://semanatecnologica.fordes.co.cu/ocs-2.3.2/public/site/239.pdf).
24. **A. Olivé Ramon, C. Gómez Seoane**. *Diseño de sistemas de software en UML*. julio del 2003.
25. **Alexander, Christopher**. *A Pattern Language*. 1997.
26. **Martinez, Evelio**. *eveliux*. *eveliux*. [En línea] 24 de julio de 2007. [Citado el: 29 de abril de 2012.] <http://www.eveliux.com/mx/estandares-y-organizaciones.php>.
27. **GROUP, INFORMATION SYSTEMs**. INFORMATION SYSTEMs GROUP. *INFORMATION SYSTEMs GROUP*. [En línea] [Citado el: 10 de 06 de 2012.] <http://indalog.ual.es/mtorres/LP/Prueba.pdf>.
28. **Booch, G.: Rumbaugh, J. y Jacobson, I y Wesley, Addison**. “*El Lenguaje Unificado de Modelado*”. 2000.
29. **Crispin, L. y House**. *Testing Extreme Programming*. 2002.

## *Glosario de Términos.*

### *Subsistema para la gestión del ID2 de la UJC de la UCI.*

30. Extreme Programming:. *Extreme Programming*:. [En línea] [Citado el: 24 de marzo de 2012.] <http://www.extremeprogramming.org/>.
31. AgileModeling. *AgileModeling*. [En línea] [Citado el: 5 de Diciembre de 2011.] <http://translate.google.com/cu/translate?hl=es&langpair=en|es&u=http://www.agilemodeling.com/essays/agileModelingXPLifecycle.htm>.
32. **Orallo, Enrique Hernández**. El Lenguaje Unificado de Modelado (UML). *El Lenguaje Unificado de Modelado (UML)*. 2002.
33. Tareas 4107 informaticas. [En línea] [Citado el: 13 de marzo de 2012.] <http://jose-sosa-95.blogspot.com/2012/03/plataformas.html>.
34. **EdwinRomanCastrillon**. slideshare. *slideshare*. [En línea] [Citado el: 22 de abril de 2012.] <http://www.slideshare.net/EdwinRomanCastrillon/diseo-de-patrones>.
35. **Hernandez Dominguez, Antonio y Pimentel Rivero, Annia**. *Sistema Informático para la Gestión de Auditoría y Control (SIGAC). Módulo de Planificación*. Habana : s.n., 2009.
36. **Esquivel Peres, José Andrés y Suarez Izquierdo, Leodan**. *Sistema de Gestión por Competencias (SGC)*. Habana : s.n., 2009.
37. **Olivares Castellanos, Abel y López Consuegra, William** . *Sistema Informático para la Gestión de la Información de La Unión de Jóvenes Comunistas de la Facultad 2*. Habana : s.n., 2009.
38. **Rojas Hernández, Jaliel Bárbara y Matos Casti, Daylin**. *SITIO DE LA UJC DE LA UCI*. Habana : s.n., 2008.