



**Universidad de las Ciencias
Informáticas**

Facultad 2

“Sistema Informático para la Gestión de la Información de La Unión de Jóvenes Comunistas de la Facultad 2”

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

Autores

Abel Olivares Castellanos

William López Consuegra

Tutor

Ing. José Raúl Perera Morales

Ciudad de la Habana, Junio de 2009

“Año del 50 aniversario del Triunfo de la Revolución”



DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos al Comité Primario de la Unión de Jóvenes Comunistas (UJC) de la Facultad 2 de la Universidad de las Ciencias Informáticas (UCI); así como a dicho centro para que hagan el uso que estimen pertinente con el mismo.

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

Abel Olivares Castellanos
Firma del Autor

William López Consuegra
Firma del Autor

Ing. José Raúl Perera Morales
Firma del Tutor



DEDICATORIA

ABEL:

DEDICO ESTE TRABAJO, TODO EL ESFUERZO QUE HE REALIZADO PARA VERLO HECHO REALIDAD, Y MIS AÑOS DE ESTUDIOS Y DEDICACIÓN A:

✓ Mi madre y mi abuela ante todo. A ellas por estar siempre conmigo desde pequeño y nunca, nunca abandonarme. Todo lo que soy se los debo a ellas que me supieron criar y guiar por el buen camino, aconsejándome, regañándome, cuidándome, acariciándome y mimándome todo el tiempo. A mi madre por ser madre y padre, para mí es lo mejor que el mundo me ha dado. Siempre ha estado conmigo en las buenas y en las malas, fue la maestra que me enseñó a leer y a escribir, ha sido y será siempre mi guía y ejemplo a seguir: A mi abuela por cuidarme, mimarme, alimentarme y malcriarme, me cuentan que fue a la que primero le dije mami. A mi abuelo Pablo y mi bisabuela Hortensia que aunque no están presente físicamente siempre llevo en mi mente todo lo bueno que fueron conmigo y sé que ellos me cuidan de alguna forma.

✓ A mi tía Tensy, a mi tío Roly, a mi tío Silvio, a mi tío Rubén, a Pochy, a Isabel, a Haydee , por siempre estar pendiente de mis problemas y preocuparse por mí.

✓ A mi primo Edgar y mi prima Lily por ser como los hermanos que siempre quise tener.

✓ A mis tías abuelas, a mis primos y primas.

✓ A Yusleidi por su paciencia, entrega, dedicación cuando estoy con ella, sobre todo a su espera incondicional y por cuidarme y quererme a mi abuela, a sus padres y su tía Marilú por ser tan buenos conmigo.

✓ A mi prima Leo, a Andresito y mi primo Reinier por estar siempre pendiente de mí y por haberme dado la mano cuando más lo he necesitado ya que he estado tan lejos de mi casa.



DEDICATORIA

- ✓ A mi tío Juan y mi tía Ela, no tengo palabras para agradecerle todo lo que han hecho por mí, por haber sido tan buenos conmigo en estos años de mi vida que he podido compartir con ellos.
- ✓ A mi primo Andresito y su familia por haberme brindado su apoyo desinteresado.
- ✓ A mi abuela Ana Delia, a mi abuelo Luis, a mi tío Angel Luis y a mis tíos y tías por preocuparse por mí.
- ✓ A mi hermano Luis Angel y su familia por ser tan atentos conmigo.
- ✓ A mis compañeros y amigos desde la Primaria hasta la Vocacional. Y una dedicatoria especial a mis compañeros y amigos de la UCI, es para los que están y los que no también, que durante estos cinco años de veras me han ayudado mucho y he aprendido algo de cada uno de ellos.
- ✓ Y a mis vecinos y amigos todos, porque de una forma u otra me admiran y me respetan.



DEDICATORIA

WILLIAM:

El presente trabajo es dedicado en especial a mi madre María Victoria Consuegra, por su amor, su dedicación, por ser luz cuando ya el camino se tornaba oscuro. A ti mamita linda, he aquí tu sueño convertido en realidad, he aquí el resultado de todos los esfuerzos que has hecho, por no dormir cuando yo no lo hacía, por siempre guiarme cuando más perdido estaba, gracias mamá. Dedicado también a mi papá José Wilfredo (William), por ser más que padre, por ser un amigo especial, por contar siempre conmigo y por siempre darme confianza. A ti mi hermanito, Yoandry, también va dedicado este trabajo, por cuidar a mami mientras no estuve y por quererme como sabes que te quiero a ti. También para Ariel, mi otro hermano, para Adrián, el más chiquito de todos, y para Ileana que incondicionalmente me ha apoyado también es dedicado este trabajo. A toda mi familia que siempre me ha apoyado y preocupado por mí, a mis vecinos, mis amigos, amigas, que a todos los quiero. En fin dedicado a todos aquellos que siempre han estado presentes en mi vida.



AGRADECIMIENTOS

DE NOSOTROS:

- ✓ Al pueblo cubano, por poner en nuestras manos la Universidad de las Ciencias Informáticas (UCI).
- ✓ A nuestras familias, por guiarnos siempre por el mejor de los caminos.
- ✓ A nuestro tutor.
- ✓ A Antonio Hernández por siempre sacarnos de esos errores que nadie ve y siempre estar dispuesto a ayudarnos.
- ✓ A Harley por estar siempre junto a nosotros y preocuparse por nuestra tesis como si fuera también suya.
- ✓ A Yunieski Zamora por brindarnos su ayuda desinteresada casi sin conocernos, de veras un millón de gracias.
- ✓ A Susy por ser la única que se brindo a ayudarnos, cuando estábamos un poco atrasados.
- ✓ A Madelis por siempre darnos aliento, exigimos, estar pendiente del informe y de nosotros en los distintos cortes, y hacernos creer que si podíamos, gracias de veras.
- ✓ A Henry, al profesor Yaser, a Yusbel, a Dennis, a Raudel, a Vladimir, a Jorge, a Adrián y a todos los compañeros del Laboratorio que de una forma u otra nos ayudamos uno a los otros, aún cuando el calor era irresistible.
- ✓ A nuestros compañeros del grupo que nos dieron aliento y nos hicieron creer que si podíamos.

A todos ellos por ser de esas personas que utilizan su conocimiento para ayudar a los demás y no para vanagloriarse de sí mismo.



RESUMEN

En la Facultad 2 de la Universidad de las Ciencias Informáticas (UCI) la información relacionada con **La Unión de Jóvenes Comunistas (UJC)** se controla manualmente, ya que no se cuenta con un sistema encargado de su gestión, por lo que pueden suceder retrasos en las entregas de las actas de los comités de base y obtención de reportes, así como pérdidas de datos. En el presente Trabajo de Diploma titulado: “Sistema Informático para la Gestión de la Información de La Unión de Jóvenes Comunistas” (F2-UJC), siguiendo el Proceso Unificado de Desarrollo (*Rational Unified Process* en inglés, habitualmente resumido como **RUP**) como metodología de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, se llevan a cabo la modelación del negocio, el levantamiento de requisitos, el diseño, así como la implementación y prueba de un sistema, que permite gestionar con mayor eficiencia la información referente a La Unión de Jóvenes Comunistas en la facultad antes mencionada, basada en la rapidez a partir de la automatización de varios de los procesos que se desarrollan en dicha área.



SUMMARY

In the school 2 of the University of Information Sciences (UCI) the information related to the Young Communists League (UJC) was controlled manually, because there was not any system that could manage it, so delays were faced when supplying the meeting reports of each committee and while obtaining basic reports, there were also data loss. In this work, entitled: " Computer system for Managing the Information of the Young Communists League (F2-UJC), we followed the steps of the Rational Unified Process (usually summarized as RUP) we also used the methodology and software development with the Unified Modeling Language UML, taking out the business model, the removal of requirements, design, and implementation and testing of a system that allowed us to manage with more efficient information regarding the Young Communists League on the university, based on the quickness from the automation of various processes that were developed in the area.



ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO1: FUNDAMENTACIÓN TEÓRICA.....	5
INTRODUCCIÓN	5
1.1 Sistemas para la gestión de la información	5
1.2 Metodologías, Lenguajes de modelado y herramientas	8
1.3 Lenguajes de programación, Herramientas y Tecnologías actuales	12
1.4. Arquitectura de software	21
1.5. Patrones de Diseño	24
CONCLUSIONES	27
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	28
INTRODUCCIÓN	28
2.1. Estudio preliminar del negocio.....	29
2.2. Modelo de negocio	31
2.3. Descripción textual de los Casos de Uso del Negocio	34
2.4. Diagramas de Actividad.....	40
2.5. Modelo de objetos	44
2.6. Reglas del Negocio	44
2.7. Relación de los Requerimientos del sistema	45
2.8. Modelo de Casos de Uso del Sistema.....	51
2.8.2 Paquetes y relación entre ellos.....	52
2.8.3 Diagrama de casos de uso del sistema a automatizar.	53
2.8.4 Descripción de los Casos de uso del sistema.	56
CONCLUSIONES	70
CAPÍTULO 3: DISEÑO DEL SISTEMA	71
INTRODUCCIÓN	71
3.1. DISEÑO	71
3.1.1 Diagramas de interacción.	71
3.1.2 Diagrama de clases del Diseño.....	77
3.2. DISEÑO DE LA BASE DE DATOS.....	85
3.2.1 Modelo lógico de datos (diagrama de clases persistentes).	85
3.2.2 Modelo físico de datos (modelo de datos).	86
CONCLUSIONES	87
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA	88
INTRODUCCIÓN	88
4.1. DIAGRAMA DE DESPLIEGUE	88
4.2. DIAGRAMA DE COMPONENTES.....	90
4.3. PRUEBA	98
4.3.1 Pruebas de Caja Negra:.....	98



ÍNDICE

CONCLUSIONES	101
CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD.....	102
INTRODUCCIÓN	102
5.1. APLICAR MÉTODO DE ESTIMACIÓN PUNTOS POR CASOS DE USO	102
5.1.1 Cálculo de Puntos de Casos de Uso sin ajustar.....	102
5.1.2 Cálculo de Puntos de Casos de Uso ajustados.....	104
5.2. CÁLCULO DEL ESFUERZO DEL FT IMPLEMENTACIÓN.....	108
5.3. DISTRIBUCIÓN DEL ESFUERZO ENTRE LAS DIFERENTES ACTIVIDADES DE UN PROYECTO	110
5.4. BENEFICIOS TANGIBLES E INTANGIBLES.	111
5.5. ANÁLISIS DE COSTO.....	112
CONCLUSIONES	113
CONCLUSIONES GENERALES	114
RECOMENDACIONES.....	115
BIBLIOGRAFÍA.....	116
REFERENCIAS BIBLIOGRÁFICAS	118
ANEXOS	120
GLOSARIO DE TÉRMINOS.....	147



ÍNDICE

ÍNDICE DE FIGURAS

Figura 1: Flujo de Trabajo de RUP.....	9
Figura 2: Patrón MVC.....	23
Figura 3: Negocio de la UJC de la Facultad 2.....	29
Figura 4: Diagrama de Casos de Uso del Negocio.	34
Figura 5: Diagrama de actividad del CUN Llenar ID2.....	40
Figura 6: Diagrama de Actividad del CUN Realizar Reunión.	41
Figura 7: Diagrama de Actividad del CUN Llenar Sanción.....	42
Figura 8: Diagrama de Actividad del CUN Hacer Cierre de Funcionamiento.....	43
Figura 9: Modelo de objetos.....	44
Figura 10: Diagrama de Paquete.	52
Figura 11: Diagrama de Casos de Uso del Sistema del paquete de Seguridad.	53
Figura 12: Diagrama de Casos de Uso del Sistema del paquete de Gestionar Dato.....	54
Figura 13: Diagrama de Casos de Uso del Sistema del paquete de Emitir Reporte.	55
Figura 14: DS del Caso de Uso del Sistema Gestionar Usuario (Escenario Autenticar Usuario).....	72
Figura 15: DS del Caso de Uso del Sistema Gestionar Usuario (Escenario Insertar Usuario).	73
Figura 16: DS del Caso de Uso del Sistema Gestionar Usuario (Escenario Modificar Usuario).	74
Figura 17: DS del Caso de Uso del Sistema Gestionar Acta (Escenario Insertar Acta).....	75
Figura 18: DS del Caso de Uso del Sistema Gestionar Acta (Escenario Modificar Acta).	76
Figura 19: Arquitectura Básica.....	77
Figura 20: DCD usando estereotipos web del Caso de Uso del Sistema Autenticar Usuario.	79
Figura 21: DCD usando estereotipos web del Caso de Uso del Sistema Gestionar Comité Base.	80



ÍNDICE

Figura 22: DCD usando estereotipos web del Caso de Uso del Sistema Registrar Asistencia.....	81
Figura 23: DCD usando estereotipos web del Caso de Uso del Sistema Gestionar Acta.	82
Figura 24: DCD usando estereotipos web del Caso de Uso del Sistema Emitir Reporte.....	83
Figura 25: DCD usando estereotipos web del Caso de Uso del Sistema Gestionar ID2.	84
Figura 26: Diagrama de Clases Persistentes del Sistema.....	85
Figura 27: Modelo Entidad-Relación del Sistema.	86
Figura 28: Diagrama de Despliegue Óptimo del F2-UJC.....	89
Figura 29 : Diagrama de Despliegue Actual del F2-UJC.....	90
Figura 30: Diagrama de componentes del Caso de Uso del Sistema Autenticar Usuario.	91
Figura 31: Diagrama de componentes del Caso de Uso del Sistema Gestionar Comité Base.....	92
Figura 32: Diagrama de componentes del Caso de Uso del Sistema Registrar Asistencia.....	93
Figura 33: Diagrama de componentes del Caso de Uso del Sistema Gestionar Acta.	94
Figura 34: Diagrama de componentes del Caso de Uso del Sistema Gestionar Sanción.	95
Figura 35: Diagrama de componentes del Caso de Uso del Sistema Emitir Reporte.....	96
Figura 36: Diagrama de componentes del Caso de Uso del Sistema Generar Cierre de Funcionamiento.	97



ÍNDICE

ÍNDICE DE TABLAS

Tabla 1: Descripción de los actores del negocio.	32
Tabla 2: Descripción de los trabajadores del negocio.	33
Tabla 3: Descripción Textual del Caso de Uso Realizar Reunión.	36
Tabla 4: Descripción Textual del Caso de Uso Llenar ID2.	37
Tabla 5: Descripción Textual del Caso de Uso Hacer Cierre de Funcionamiento.	38
Tabla 6: Descripción Textual del Caso de Uso Llenar Sanción.	39
Tabla 7: Definición de los actores del sistema.	51
Tabla 8: Descripción del caso de uso del sistema Autenticar Usuario.	57
Tabla 9: Descripción del caso de uso del sistema Gestionar Comité Base	60
Tabla 10: Descripción del caso de uso del sistema Registrar Asistencia.	61
Tabla 11: Descripción del caso de uso del sistema Gestionar Acta.	63
Tabla 12: Descripción del caso de uso del sistema Gestionar Sanción.	66
Tabla 13: Descripción del caso de uso del sistema Emitir Reporte.	68
Tabla 14: Descripción del caso de uso del sistema Gestionar Cierre de Funcionamiento.	69
Tabla 15: Pruebas de Caja Negra para el caso de uso “Autenticar Usuario”	98
Tabla 16: Pruebas de Caja Negra para el caso de uso “Gestionar sanción”: Sección: insertar sanción	99
Tabla 17: Pruebas de Caja Negra para el caso de uso “Gestionar sanción”: Sección: modificar sanción	100



INTRODUCCIÓN

La gestión de la información es un elemento fundamental para el control y la toma de decisiones sobre los procesos que se desarrollan en las organizaciones.

En el mundo actual existe una proliferación de sistemas de este tipo, por su gran utilidad en los procesos organizacionales, al igual que en el mundo económico o productivo en las organizaciones gubernamentales o políticas de los países también se ha aplicado dentro de todo el proceso de informatización de dichas instituciones.

Obviamente para la utilización de la información, es necesario gestionarla; es decir: recolectarla, registrarla y analizarla; esto implica determinar la información precisa, cuando sea necesaria y por supuesto utilizarla y divulgarla.

Un buen sistema de gestión de la información debe, por tanto, ayudar a los administradores del proyecto a saber qué información necesitan obtener; para tomar diferentes decisiones en distintos momentos.

Cuba no ha permanecido al margen de la informatización de sus instituciones y la automatización de procesos productivos ha alcanzado un alto nivel, aunque en instituciones políticas y gubernamentales no se ha notado un gran avance. En estos momentos el país se encuentra en la informatización de los procesos de la sociedad donde se inserta, entre otras acciones, la creación de la Universidad de las Ciencias Informáticas, que representa un salto en el estudio de la informática y en la producción de software en el país.

En la propia universidad como comunidad universitaria al fin, conviven diferentes organizaciones políticas que tienen una gran incidencia en la vida universitaria y llevan la vida política de cada una de las personas que trabajan o estudian en ella, una de ellas, la Unión de Jóvenes Comunistas, es la organización política de la juventud cubana, que agrupa la vanguardia juvenil y es la cantera fundamental de crecimiento del Partido Comunista de Cuba (PCC).

La UJC en la UCI organiza, dirige y controla un gran cúmulo de tareas simples y complejas que involucran a los militantes y se hacen en condiciones difíciles debido a la dinámica de la universidad.



INTRODUCCIÓN

Las propias condiciones tecnológicas creadas para los procesos docentes y productivos de la universidad pueden ser usadas a favor de la automatización de la gestión de la información de dicha organización en la universidad y particularmente en la Facultad 2 donde se evidencia las mismas necesidades que en el resto de las facultades.

Existe hoy, un portal que se encuentra hospedado en la intranet universitaria al que puede acceder todo el personal del centro; pero que sólo se utiliza como medio para mantener informada a la militancia.

Los distintos actores dentro de la dirección de los procesos de la organización en la facultad realizan procesos tales como: la constitución de Comités de Base, registro de universo juvenil, registro de actas de las reuniones mensuales de cada comité de base, recogida de la cotización de la organización, y por supuesto la generación de toda la información que desde los organismos superiores se les exige, de forma manual y en alguna medida semiautomática por la utilización de herramientas como el Microsoft Word y Excel, lo cual imposibilita la disponibilidad y el acceso a la información en el tiempo requerido.

A partir de entrevistas realizadas a los distintos niveles de dirección de la organización se han identificado las siguientes deficiencias que esto provoca:

- Secretario de Comité Primario:
 1. La entrega de la información (Actas de las reuniones, informes, etc.) tiene demoras por el propio manejo que se le da a la información
 2. Existen inconsistencias en la información (Actas de las reuniones, informes, etc.) que llega de los niveles inferiores.
 3. Se dificulta la información estadística de los datos (militantes que incumplen, % de asistencia, cantidad de opiniones, etc.).
- Activista de Funcionamiento:
 1. Difícil control de la entrega de todas las actas que se generan después de realizar mensualmente cada una de las reuniones del comité de base.



INTRODUCCIÓN

2. Dificil confección del Cierre de Funcionamiento al tener que hacerlo manualmente.
- Activista de Procesos Políticos:
 1. Dificil control de las sanciones de los militantes al tener que ser buscadas en archivos de los expedientes de los militantes.
 - Documentador:
 1. Dificil control de las Plantillas de Control de la Militancia de la UJC (ID2) al tener que ser archivadas en herramientas semiautomáticas como el Microsoft Excel.
 - Secretario de Acta de los Comité de Base:
 1. Dificultad en el control de la asistencia de los militantes a las reuniones.
 2. Dificil registro de las actas en el momento de la reunión del comité de base.
 3. Retraso en la recogida y entrega de la información a los niveles superiores de dirección. (Opiniones, Acuerdos, Plan de Trabajo Trimestral)
 4. Dificultades en cuanto a la gestión de acuerdos.
 - Militante:
 1. Dificil acceso a la información que se genera de las actas de los propios comités de base.

Estas dificultades conllevan a una situación difícil dentro de la organización, que aparejado a la gran cantidad de personas que militan en ella dentro de la facultad, se hace en este momento casi imposible el control de toda la información que es tan necesaria.

A partir de dichas dificultades se ha creado una necesidad inmediata en la organización relacionada con la automatización de estos procesos por lo que se ha planteado el siguiente **problema**:

¿Cómo mejorar el registro, control y recuperación de la información que se genera en los procesos de la UJC de la Facultad 2?

El **objeto de estudio** de dicha investigación radica en Los Sistemas Informáticos de Gestión de Información y el **campo de acción** son los Sistemas de Gestión de la Información de la UJC.



INTRODUCCIÓN

El **objetivo del presente trabajo** es desarrollar un sistema informático capaz de llevar la gestión de la información de la UJC en la Facultad 2.

Para cumplir con el objetivo antes propuesto se realizarán las siguientes **tareas de investigación**:

1. Realizar un estudio del Proceso de Desarrollo de Software de Gestión de Información.
2. Analizar las características del proceso de gestión de la información de la UJC.
3. Estudio de las principales metodologías, herramientas y tecnologías de desarrollo de software que se utilizan para los Sistemas de Gestión.

El **resultado que se espera** del trabajo es la obtención de un software capaz de informatizar los procesos que se llevan a cabo en la UJC de la Facultad 2, el cual permita el acceso rápido a la información y un registro amigable de la misma.

Para el desarrollo del trabajo se utilizarán los siguientes métodos:

Métodos teóricos:

- ✓ **Análisis y Síntesis**: para el procesamiento de la información del negocio de la organización y arribar a las conclusiones de la investigación, así como para precisar las características del modelo arquitectónico propuesto.
- ✓ **Histórico - Lógico**: Para determinar las tendencias actuales de los sistemas de gestión de la información, de los modelos de desarrollo y las técnicas, lenguajes y herramientas utilizadas durante la investigación.
- ✓ **Inducción y deducción**: A partir del estudio de distintos estilos y modelos de arquitecturas arribar a proposiciones de estilos de arquitecturas específicas.
- **Método Sistemico**: Para determinar los componentes y definir las relaciones entre estos.

Métodos empíricos:

- **Observación**: Para la percepción selectiva de las restricciones y propiedades del sistema y para la determinación de la problemática que da origen a la investigación.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

INTRODUCCIÓN

En el desarrollo de este capítulo se realizará un estudio del estado arte donde se abordará de forma valorativa y crítica la situación actual del proceso de gestión de la información de la UJC en la Facultad 2 de la Universidad de las Ciencias Informáticas. Se ofrecerán elementos de las posibles herramientas y tecnologías que se pueden utilizar en el desarrollo, así como los lenguajes de programación y metodologías posibles a utilizar para la realización del sistema, de forma tal que quede especificado el marco teórico de la investigación.

1.1 SISTEMAS PARA LA GESTIÓN DE LA INFORMACIÓN

Software de Gestión

A continuación se muestran algunas definiciones formales de lo que es Gestión.

1ra definición de gestión: Es el proceso mediante el cual se obtiene, despliega o utiliza una variedad de recursos básicos para apoyar los objetivos de la organización (1).

2da definición de gestión: Proceso emprendido por una o más personas para coordinar las actividades laborales de otros individuos (2).

3da definición de gestión: Gestionar es coordinar todos los recursos disponibles para conseguir determinados objetivos, implica amplias y fuertes interacciones fundamentalmente entre el entorno, las estructuras, el proceso y los productos que se desean obtener (3).

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



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Planificar: Es el proceso de establecer objetivos con el fin de alcanzar 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 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.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Con el surgimiento de la teoría de la organización, se acentuó la importancia de la información. Una organización es un sistema conformado por personas, recursos materiales e información. Esta última determina el "orden y el caos" (4) entre los individuos, los recursos y en la interrelación personas-recursos. Por esta razón, debe considerarse a las organizaciones como sistemas de información.

En el mundo se han concebido miles de sistemas que realizan la gestión de la información, cada uno de ellos orientados hacia un área específica como por ejemplo: Sistemas de Gestión de la información académica, sistemas de gestión de la información contables, económicos y financieros, sistemas para la gestión gubernamental, entre otras muchas áreas de comercio.

En la universidad existen o se encuentran en desarrollo proyectos de gestión de la información por ejemplo: AKADEMOS (Sistema de Gestión Académica) el cual permite la gestión de la información que se genera del proceso docente; el proyecto CEDRUX (ERP cubano) el cual constituye un sistema para la gestión de recursos empresariales. Estos dos sistemas por solo citar alguno de ellos son sistemas de gestión de la información pero que por sus propias especificidades son imposibles de adaptar pues pertenecen a distintas áreas la información que gestionan pero sirven como referencia para el estudio de las tecnologías utilizadas y otros aspectos.

En la universidad, ni en ninguna otra institución del país, existe una herramienta que sea capaz de llevar a cabo el control de la información de la UJC, razón por la cual se presenta la necesidad en la Facultad 2 de desarrollar un **sistema automatizado, capaz de cumplir con los principales requerimientos de la organización.**

En la Facultad 1, se está trabajando en la creación de un Sistema para la Gestión de la Información de la UJC, proyecto el cual lo han llamado "Caino" pero todavía está en la fase de implementación, que es una de las más importantes porque es en ella donde se obtienen las principales funcionalidades de la primera versión del sistema y es una de las que más tarda. Debido a ello que se haya propuesto el desarrollo de



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

un nuevo sistema para la gestión de la información de la UJC en la Facultad 2, que permita darle solución a la problemática existente (Ver introducción).

1.2 METODOLOGÍAS, LENGUAJES DE MODELADO Y HERRAMIENTAS

La tendencia en este trabajo es mejorar el proceso de gestión de la información de la UJC en la Facultad 2 logrando su informatización, así como todas las demás actividades que lleva implícito este proceso.

Para el desarrollo del sistema se realizó un estudio sobre las posibles herramientas a utilizar en su construcción. Teniendo en cuenta la tendencia actual y las novedades en este campo se hace la siguiente propuesta.

1.2.1. Proceso Unificado de Desarrollo de Software.

El Proceso Unificado de Desarrollo de Software RUP (del inglés: *Rational Unified Process*) es una metodología de desarrollo de software no ágil. Constituye la metodología estándar más utilizada para el diseño, implementación y documentación de sistemas orientados a objetos. No tiene pasos firmemente establecidos, sino un conjunto de flujos de trabajo adaptables al contexto y necesidades de cada organización.

El ciclo de vida de RUP se caracteriza por:

1. **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
2. **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. La arquitectura invoca los elementos más significativos del sistema y está influenciada entre otros por plataformas de software, sistemas operativos, gestores de bases de datos, protocolos, consideraciones generales y requisitos no funcionales. La arquitectura abarca decisiones importantes sobre la organización del sistema, los elementos estructurales, sus interfaces y comportamientos.

- 3. Iterativo e Incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración. Es práctico dividir el trabajo en partes más pequeñas o mini-proyectos. Cada mini-proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. Cada iteración se realiza de forma planificada es por eso que se dice que son mini-proyectos.

Las fases antes mencionadas están formadas por nueve flujos de trabajo principales. Los seis primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo:

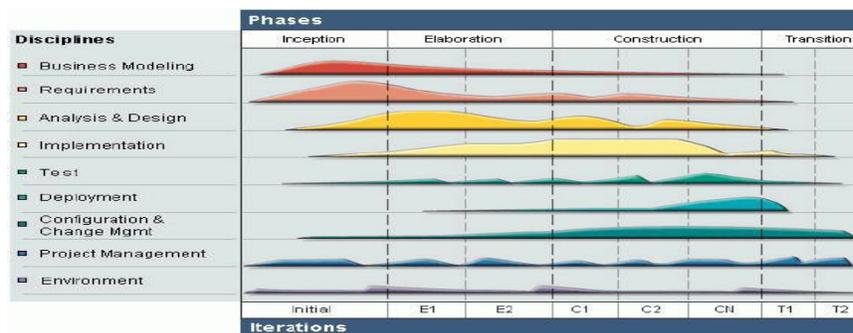


Figura 1: Flujo de Trabajo de RUP.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Después de realizar el estudio de las principales metodologías de desarrollo de software por parte del equipo de desarrollo se decidió la utilización de RUP como proceso rector por adaptarse a las características y complejidad de este proyecto de software. Esta metodología hace énfasis en una buena captura de los requisitos lo que garantiza la satisfacción de los clientes, es apropiada para proyectos donde la documentación juega un papel importante por si necesita cambios en los procesos, cuestiones estas que no son abordadas de esta manera por las metodologías ágiles. Otras características que llevaron a optar por RUP es que posee características muy importantes como: dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental, lo cual permite darse cuenta a tiempo de los errores cometidos y solucionarlos, además constituye una metodología de la cual se tiene amplio conocimiento en la universidad donde sus procesos, técnicas, y herramientas para su automatización son estudiadas en los cursos de Ingeniería de Software I y II.

1.2.2. Lenguaje Unificado de Modelado.

El Lenguaje Unificado de Modelado UML (del inglés: *Unified Modeling Language*) es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Los diagramas fundamentales que emplea UML son los siguientes:

- *Diagrama de casos de uso*: Especifica las funcionalidades y el comportamiento de un sistema, su interacción con los usuarios del mismo y otros sistemas.
- *Diagrama de Comportamiento o Interacción*: Muestra las interacciones entre objetos ocurridas en un escenario o parte del sistema.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- *Diagrama de Clases*: Representa un conjunto de elementos del modelo que son estáticos. Estos aspectos estáticos modelan características del software como pueden ser su estructura interna y la representación que se hará de la información en la aplicación.
- *Diagrama de Implementación*: Muestra los aspectos físicos del sistema e incluye la estructura del código fuente y la implementación.

1.2.3. Herramienta CASE

Visual Paradigm es una Suite de herramientas CASE (varias herramientas en una sola), dotada de una buena cantidad de productos o módulos para facilitar el trabajo durante la confección de un software, las transiciones del análisis al diseño, y de este a la implementación, están adecuadamente integradas dentro de la herramienta, de manera que reduce significativamente los esfuerzos de todas las etapas del ciclo de desarrollo de software, lo cual garantiza la calidad del producto final. Es fácil de usar, es amigable, soporta ingeniería inversa, generación de código, importación desde Rational Rose, exportación/importación XMI, generador de informes y editor de figuras entre otras cualidades. Posibilita crear un gran conjunto de artefactos de las distintas fases del desarrollo del software. Presenta sincronización entre diagramas de entidad-relación y diagramas de clases persistentes, interoperabilidad con otras aplicaciones, disponibilidad de múltiples versiones, para cada necesidad, disponibilidad de integrarse en los principales IDEs y en múltiples plataformas. (5).

El sistema que se propone será desarrollado con las técnicas de POO, apoyándose en *Visual Paradigm* y usando como notación el lenguaje UML, que es la base del funcionamiento visual de RUP.

La selección de dicha herramienta CASE viene dada fundamentalmente por estas tres características:

- Un entorno de desarrollo interactivo con un tiempo de respuesta rápido, recursos dedicados y una comprobación de errores desde el principio.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- La automatización de muchas tareas de desarrollo y mantenimiento del software.
- La programación Visual proporcionada por potentes interfaces gráficas.

Además es un hecho importante mencionar que en la universidad se tiene la licencia para su uso, aspecto de gran importancia para el logro de la soberanía informática.

1.3 LENGUAJES DE PROGRAMACIÓN, HERRAMIENTAS Y TECNOLOGÍAS ACTUALES

Un lenguaje de programación es un medio de comunicación necesario para que las computadoras puedan interpretar las instrucciones que el hombre les da y mediante estas pueda controlar su comportamiento, las instrucciones que forman dicho programa son conocidas como código fuente, las instrucciones consisten en un grupo de reglas sintácticas que definen su estructura y en un grupo de reglas semánticas que definen el significado de sus elementos. Estas reglas son utilizadas por el programador a través de las cuales crea un programa o subprograma.

1.3.1. ¿Por qué Aplicación Web y no Desktop?

El sistema se desarrollará mediante una aplicación web porque las condiciones naturales del negocio las permiten y se pueden adaptar muy fácilmente a estos tipos de sistemas, en otras palabras el negocio es por naturaleza adaptable a un sistema web. Los programas de computación basados en la Web son multiplataforma para el cliente, no hay necesidad de actualizaciones y son centralizados; a diferencia de los desktop, siempre están al día e instantáneamente disponibles, no importa dónde esté el usuario o qué sistema operativo esté utilizando.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.3.2. Programación Orientada a Objetos.

La Programación Orientada a Objetos (POO) es soportada por las nuevas versiones de muchos de los lenguajes de programación de alto nivel, es una técnica de estructuración en la que los objetos son los principales elementos de construcción que se conectan entre sí por medio de mensajes, y pueden ser entidades reales e incluso entes conceptuales que con frecuencia se encuentran en el mundo que nos rodea. Las principales ventajas que brinda esta técnica vienen determinadas por el encapsulamiento, la herencia y el polimorfismo:

- 1) El encapsulamiento es el proceso de mantener agrupado bajo una misma cápsula o unidad la estructura y el comportamiento de los objetos
- 2) La herencia es el mecanismo que permite construir clases nuevas a partir de clases existentes
- 3) y el polimorfismo constituye la capacidad que poseen objetos pertenecientes a clases diferentes de comportarse en forma (morfo) diferente (poli) ante la recepción de un mismo mensaje.

1.3.3. Lenguaje de Mercado de Hipertexto.

El lenguaje de marcas HTML (del inglés: *HyperText Marckup Language*), se utiliza para definir las páginas web, es muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, fotos, sonido...). Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web, es decir se utiliza para crear las páginas web y les indica a los navegadores cómo deben mostrar el contenido. (6)

El lenguaje html contiene dos partes:

- **Contenido**, que es el texto que se verá en la pantalla de un ordenador
- **Etiquetas y atributos** que estructuran el texto de la página web en encabezados, párrafos, listas, enlaces, etc. y normalmente no se muestra en pantalla. (6)



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.3.4. Hojas de Estilo en Cascada.

Hojas de Estilo en Cascada CSS (del inglés: *Cascading Style Sheets*), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. Se utiliza para dar estilo a documentos HTML, separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos HTML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento, funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne (7).

1.3.5. JavaScript

Javascript es el lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar los códigos, razón por la cual se dice que es un lenguaje interpretado. Es un lenguaje que utiliza Windows y sistemas X-Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros. Es necesario resaltar que hay dos tipos de JavaScript: por un lado está el que se ejecuta en el cliente, este es el Javascript propiamente dicho, aunque técnicamente se denomina Navigator JavaScript. Pero también existe un Javascript que se ejecuta en el servidor, es más reciente y se denomina LiveWire Javascript. (8)

Algunas características principales del lenguaje son:



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Su sintaxis es similar a la usada en C, al ser un lenguaje del lado del cliente este es interpretado por el navegador, no se necesita tener instalado ningún Framework.
- Permite la programación orientada a objetos.
- Las variables pueden ser definidas como: string, integer, float, boolean simplemente utilizando “var”. Se puede usar “+” para concatenar cadenas y variables.
- Javascript es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros. (8)

1.3.6. Pre-procesador de Hipertexto.

Pre-procesador de Hipertexto PHP (del inglés: *Hypertext Preprocessor*) es un Lenguaje de programación del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Es un lenguaje interpretado de propósito general, que está diseñado especialmente para desarrollo web y puede ser introducido dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores. Es un lenguaje multiplataforma, tiene capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda, es libre por lo que se presenta como una alternativa de fácil acceso para todos, permite las técnicas de Programación Orientada a Objetos, biblioteca nativa de funciones sumamente amplia e incluida, no requiere definición de tipos de variables y tiene manejo de excepciones (desde PHP5). Las principales características para tener en cuenta en un lenguaje script son: velocidad, estabilidad, seguridad y simplicidad y PHP cuenta con cada una de estas características en buena medida: (9)



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- **Velocidad:** No solo la velocidad de ejecución, que es importante, sino también no crear demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien a otro software, especialmente bajo ambientes Unix y cuando se configura como módulo de Apache.
- **Estabilidad:** La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de errores, pero teniendo como respaldo una increíble comunidad de programadores y usuarios, es mucho más difícil que los errores sobrevivan. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
- **Seguridad:** El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad que pueden ser configurados desde el archivo (.ini).
- **Simplicidad:** Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente.

1.3.7. Framework (Symfony).

¿Qué hace un framework (del español: Marco de trabajo)?

- Simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes.
- Proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener.
- Facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. (10)



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Marco de trabajo: Symfony

Este está diseñado para optimizar el desarrollo de aplicaciones web, utilizando el lenguaje PHP5, el mismo separa la lógica de negocio de la lógica de servidor y la presentación, aplicando estilos de arquitectura en Capas y Modelo Vista Controlador.

El framework proporciona un conjunto de clases y herramientas encaminadas a disminuir el tiempo de desarrollo de una aplicación, ya que permite la reutilización.

Características

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo. (9)

A partir de las características del proyecto a ejecutarse, las cuales indican que serán un sistema de complejidad media, que realmente no es muy grande pero que pudiera crecer con el incremento de otras funcionalidades -las cuales no se tendrán en cuenta en esta versión para el sistema-, y para agilizar el desarrollo del sistema se decidió la utilización de un framework como este que permita su mantenimiento por el uso de código ligero, legible y efectivo; además por las propias características de que sea una aplicación web con interfaces amigables y rápida también es recomendable el uso del framework.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.3.8. Zend Studio for Eclipse.

Es un plugin desarrollado para ayudar en el desarrollo PHP por medio de la IDE Eclipse. Zend brinda la posibilidad de utilizar el Zend Studio for Eclipse para mejorar la calidad de los proyectos en PHP, agilizar los ciclos de desarrollos y simplificar la complejidad de los proyectos. El plugin incluye herramientas para edición, debugging, análisis, optimización y bases de datos. IDE totalmente Open Source que mejora los debugger y ayudantes de código, tiene soporte para extensiones de PHP y una de las mejoras principales es que cuenta con un ayudante nativo para HTML, JS y CSS donde todo viene integrado y listo para usar. (11)

1.3.9. Servidor Web.

Un servidor web sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante HTTP, es decir, es un software que funciona en un ordenador y maneja la entrega de los componentes de las páginas como respuesta a peticiones de los navegadores de los clientes. Se pueden utilizar varias tecnologías en el servidor para aumentar su potencia más allá de su capacidad de entregar páginas HTML; éstas incluyen seguridad SSL entre otras. (12)

1.3.10. Servidor Web Apache.

Apache es el servidor web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Apache es una muestra, al igual que el sistema operativo Linux, de que el trabajo voluntario y cooperativo dentro de Internet es capaz de producir aplicaciones de calidad profesional difíciles de igualar. (13)

La licencia Apache es una descendiente de la licencias BSD, no es GPL. Esta licencia te permite hacer lo que quieras con el código fuente siempre que les reconozcas su trabajo.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Características que hacen popular a este software libre:

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Apache es una tecnología gratuita de código fuente abierto. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Esto le da una transparencia a este software de manera que si se quiere ver que es lo que se está instalando como servidor, lo puede saber sin ningún secreto.
- Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este, y están ahí para que se instalen cuando se necesite
- Otra cosa importante es que cualquiera que posea una experiencia decente en la programación de C o Perl puede escribir un módulo para realizar una función determinada.
- Apache te permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor. (10)

1.3.11. Servicios Web.

Los Servicios Web WS (del inglés: *Web Service*) son un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos. Estas aplicaciones intercambian datos entre sí con el objetivo de ofrecer unos servicios para presentar información dinámica al usuario. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En todo este proceso intervienen una serie de tecnologías que hacen posible esta circulación de información. Por un lado, estaría SOAP (Protocolo Simple de Acceso a Objetos). Se trata de un protocolo basado en XML, que permite la interacción entre varios dispositivos y que tiene la capacidad de transmitir información compleja. Los datos pueden ser transmitidos a través de HTTP, SMTP, etc. SOAP especifica el formato de los mensajes. El mensaje SOAP está compuesto por un envelope (sobre), cuya estructura está formada por los siguientes elementos: header (cabecera) y body (cuerpo). Por otro lado, WSDL (Lenguaje de Descripción de Servicios Web), permite que un servicio y un cliente establezcan un acuerdo en lo que se refiere a los detalles de transporte de mensajes y su contenido, a través de un documento procesable por dispositivos. WSDL representa una especie de contrato entre el proveedor y el que solicita. WSDL especifica la sintaxis y los mecanismos de intercambio de mensajes. (14)

1.3.12. Sistema de Gestión de Base de Datos

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos de software libre, por ende open *source* y publicado bajo la licencia BSD.

Algunas de sus principales características son:

Alta concurrencia: Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

Amplia variedad de tipos nativos: PostgreSQL provee nativamente soporte para números de precisión arbitraria, texto de largo ilimitado, figuras geométricas (con una variedad de funciones asociadas), direcciones MAC, arrays y adicionalmente los usuarios pueden crear sus propios tipos de datos.

Otras de las características son las claves ajenas también denominadas Llaves ajenas o Claves Foráneas (*foreign keys*), Disparadores (*triggers*): Un disparador o trigger se define en una acción específica basada en algo ocurrente dentro de la base de datos. En PostgreSQL esto significa la ejecución de un



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

procedimiento almacenado basado en una determinada acción sobre una tabla específica, vistas, procedimientos almacenados, integridad transaccional, herencia de tablas.

PostgreSQL ofrece muchas ventajas respecto a otros sistemas de bases de datos, entre estas se encuentra una de las principales: instalación ilimitada, no hay costo asociado a la licencia del software. Estabilidad y confiabilidad legendarias, es extensible es decir el código fuente está disponible para todos sin costo. Si su equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días. Multiplataforma, está disponible en casi cualquier Unix (34 plataformas en la última versión estable). (15)

1.4. ARQUITECTURA DE SOFTWARE

La Arquitectura de Software (AS)

Según la IEE, la Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.

Es el esqueleto o base de una aplicación donde es analizada desde varios puntos de vista. Constituye la nueva visión que los desarrolladores tienen de los sistemas de software en la última década. Desde los pequeños programas hasta los sistemas más grandes poseen una estructura y un comportamiento que los hace clasificables según su "arquitectura". (16)

Este nuevo aspecto hace posible el estudio de sistemas ya implementados así como el desarrollo de nuevos, según la categoría del problema que resuelven y el tipo de "arquitectura". Los distintos niveles de abstracción de la funcionalidad de los sistemas, están asociados con diferentes aspectos y componentes de su "arquitectura". Esta asociación se manifiesta en forma clara en las distintas etapas del proceso de desarrollo de software. Existen distintos tipos de arquitecturas de sistemas que se clasifican en patrones de diseño y que tienen aplicaciones a lo largo del proceso de desarrollo.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.4.1 Estilos Arquitectónicos

Arquitectura n-Capas

Con el objetivo de solventar los problemas de escalabilidad, disponibilidad, seguridad e integración que pueden presentar las aplicaciones compactas, se ha generalizado la división de las aplicaciones en capas. La capa que se agrega es la que surge de separar definitivamente las reglas de negocio de la de Acceso a Datos. Esta arquitectura trae consigo la ventaja de aislar definitivamente la lógica de negocios de todo lo que tenga que ver con el origen de datos, de modo que ante cualquier eventual cambio, solo se deberá tocar un módulo específico, así como al momento de plantear la escalabilidad del sistema no se afrontarán grandes modificaciones. Dentro de este tipo de arquitectura una de las divisiones muy utilizada en el desarrollo de sistemas es la arquitectura en tres capas.

Patrón de arquitectura Modelo Vista Controlador (MVC)

El MVC (Model/View/Controller) fue diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas. MVC es un patrón de arquitectura de software clásico que está formado por tres niveles:

- **El modelo**, que representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- **La vista**, que transforma el modelo en una página Web que permite al usuario interactuar con ella.
- **El controlador**, que se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En la figura se puede observar el patrón MVC.

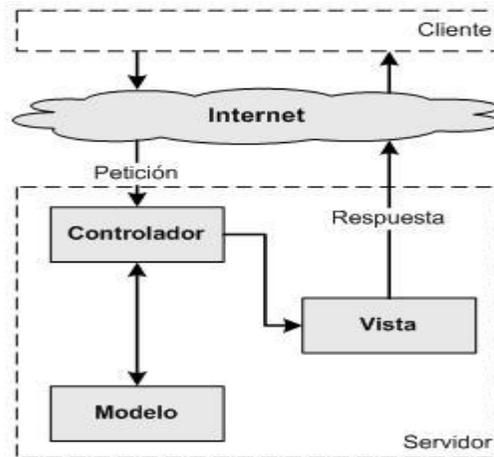


Figura 2: Patrón MVC.

El patrón MVC separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. En MVC, la capa de presentación está partida en controlador y vista. La principal separación es entre presentación y dominio; la separación entre vista y controlador es menos clara.

Este modelo de arquitectura presenta varias ventajas:

- Hay una clara separación entre los componentes de un programa; lo cual permite implementarlos por separado.
- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

Al incorporar el modelo de arquitectura MVC a un diseño, las piezas de un programa se pueden construir por separado y luego unirlos en tiempo de ejecución. Si uno de los componentes, posteriormente, se observa que funciona mal, puede reemplazarse sin que las otras piezas se vean afectadas.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.5. PATRONES DE DISEÑO

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Identifica: Clases, Instancias, Roles, Colaboraciones y la distribución de responsabilidades.

Patrones GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (Patrones Generales de Software para Asignar Responsabilidades). El nombre se eligió para indicar la importancia de captar (grasping) estos principios, si se quiere diseñar eficazmente el software orientado a objetos. Estos fueron los patrones que se aplicaron para realizar el diseño de la aplicación que propone este trabajo:

Experto: Este patrón define que la responsabilidad de creación de un objeto debe recaer sobre la clase que conoce toda la información necesaria para crearlo, es un principio básico que suele utilizarse en el diseño orientado a objetos. Este es uno de los más utilizados en el desarrollo de la aplicación, por ejemplo Propel es la librería externa que utiliza Symfony como ORM, encapsula toda la lógica de los datos y son generadas las clases con todas las funcionalidades comunes de las entidades.

Controlador: Todas las peticiones deben ser manejadas por un único controlador frontal, que es el punto de entrada a la aplicación. Este es el caso de *front-controller* definido por Symfony para cada entorno. Quien se encarga de recibir y delegar al controlador del módulo todas las peticiones así como de mostrar las respuestas de dichas peticiones.

Alta Cohesión: La información de una clase debe ser coherente y estar relacionada con la clase. Asignar una responsabilidad de modo que la cohesión siga siendo alta. Una clase con baja cohesión hace muchas



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

cosas no afines o un trabajo excesivo. Una clase con mucha cohesión es útil porque es bastante fácil darle mantenimiento, entenderla y reutilizarla. La ventaja que significa una gran funcionalidad también soporta un aumento de la capacidad de reutilización. Un ejemplo de esto es que Symfony permite asignar responsabilidades con una alta cohesión, por ejemplo la clase *Actions* tiene la responsabilidad de definir las acciones para las plantillas y colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las *properties*, es decir, está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas proporcionando que el software sea flexible frente a grandes cambios.

Bajo Acoplamiento: Las clases deben comunicarse con un número pequeño de clases tanto como sea posible. Ejemplo de esto se evidencia en la clase *Action*, la cual hereda solamente de *sfActions* para lograr un bajo acoplamiento de clases.

Otros Patrones que utiliza Symfony:

Según el libro GOF existen 3 tipos de patrones:

- **De Creación:** abstraen el proceso de creación de instancias.
- **Estructurales:** se ocupan de cómo clases y objetos son utilizados para componer estructuras de mayor tamaño.
- **De Comportamiento:** atañen a los algoritmos y a la asignación de responsabilidades entre objetos.

Patrón Creacional

Singleton (Instancia única):

Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. En Symfony existe un Singleton del contexto mediante el cual se puede acceder a todos los objetos del núcleo del Framework a través del método `getContext ()` en las acciones y de esta forma obtener cualquier tipo de información relacionada con la petición.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Patrones Estructurales

Decorador (Envoltorio):

Añade funcionalidad a una clase dinámicamente. Este patrón se manifiesta en la capa de la vista la cual se compone de un *layout* general que se puede definir para la aplicación entera o para un determinado módulo y que decora o que se compone en su interior de las plantillas.

Adapter (Adaptador):

Adapta una interfaz para que pueda ser utilizada por una clase que de otro modo no podría utilizarla. Symfony implementa este patrón ya que es posible cambiar en cualquier momento la base de datos, simplemente configurando unos archivos y este cambio no sería problema para las clases que usan el modelo, ya que estas últimas brindan una interfaz común independientemente del ambiente en el cual se ejecute. (2)



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

CONCLUSIONES

En este capítulo se pueden apreciar la selección de la metodología, el lenguaje de modelado, herramientas, lenguajes de programación y el sistema gestor de base de datos a utilizar en el desarrollo de la aplicación que le dará solución al problema que se planteó en el comienzo de este trabajo el cual tiene que ver con la automatización de la información de la UJC en la Facultad 2 de la UCI. También se dan explicaciones y justificaciones del porqué de la elección de la Metodología RUP como guía para el desarrollo del software y la complementación de la misma con el lenguaje de modelado UML para como su nombre lo indica modelar un "plano" del sistema, incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación y esquemas de bases de datos que será de mucha importancia para que cualquier otro desarrollador pueda familiarizarse con el sistema e incluso poder hacerle cambios; por otra parte y para concluir se puede observar el porqué de: PHP como lenguaje de programación, la utilización de *Symfony* como Framework para agilizar el desarrollo del sistema, *PostgreSQL* como Sistema Gestor de Base de Datos y una explicación de lo que es Gestión sus características y conceptos fundamentales.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

INTRODUCCIÓN

En el desarrollo de este capítulo se dará a conocer la situación problemática planteada; pero para empezar a construir el sistema es necesario detenerse a analizar los procesos que se desarrollan en el área que se pretende automatizar, la descripción de los procesos fundamentales del negocio se hace para lograr una mayor comprensión del problema que se desea resolver. La modelación del negocio, es decir, los actores y trabajadores del negocio y por qué lo son, el Diagrama de Casos de Uso del Negocio, la Descripción textual de los casos de uso del Negocio, los diagramas de actividad de cada uno de los casos de uso del negocio, el modelo de objetos y las reglas del negocio permiten que se logre un entendimiento entre clientes y desarrolladores. En este capítulo también se describe el objeto de estudio, se muestran los diagramas de casos de uso del sistema, la descripción de los casos de uso, los actores del sistema y se da a conocer la relación de los requerimientos funcionales y no funcionales.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1. ESTUDIO PRELIMINAR DEL NEGOCIO

El Comité Primario de la UJC de la Facultad 2, está constituido de la siguiente manera:

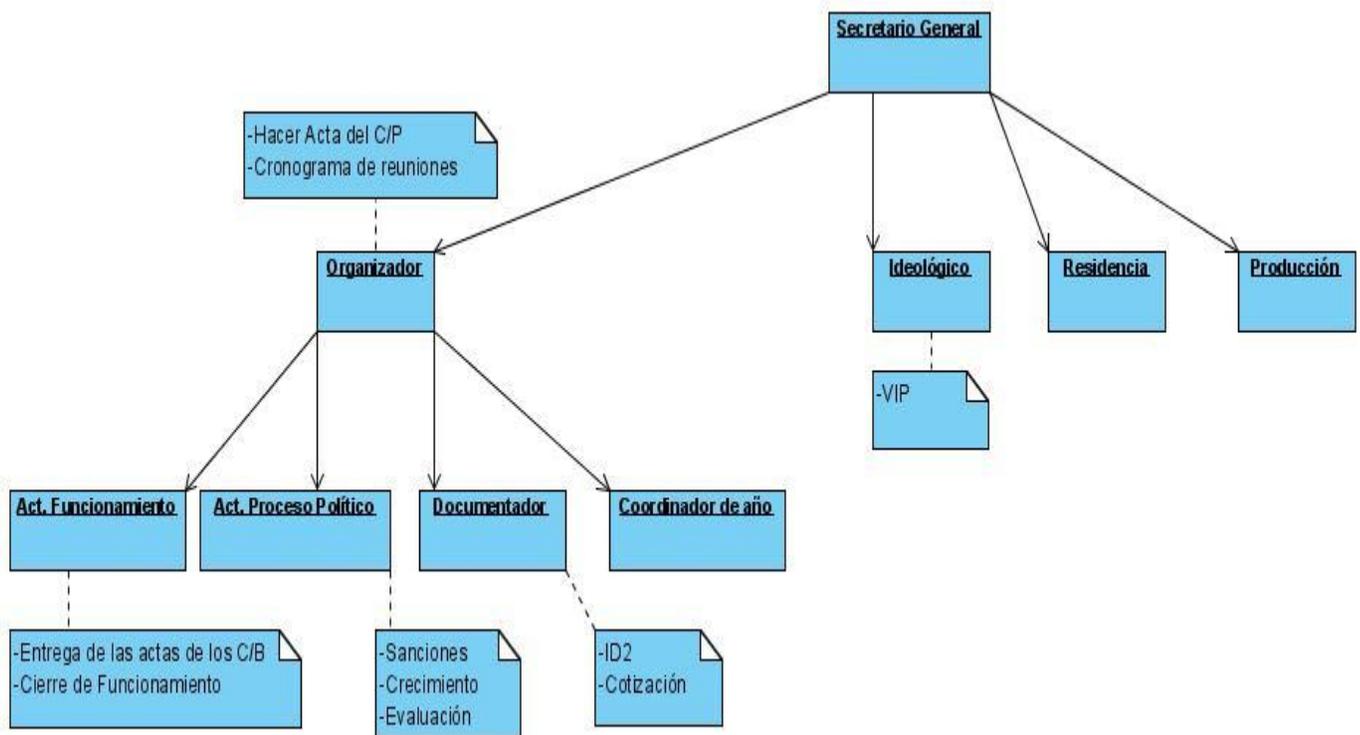


Figura 3: Negocio de la UJC de la Facultad 2.

El Secretario General es la máxima autoridad de la UJC en la facultad, es el responsable de todas las tareas que se lleven a cabo en la organización política. Tiene como subordinados al Organizador-es el encargado de organizar y planificar todas las tareas que se lleven a cabo dentro del comité de base-, el Ideológico -es el encargado de la Vía de Instrucción Política -, el que atiende Residencia-es el encargado de la parte de la Beca, limpieza de aptos, cuartería, etc.- y Producción-es el encargado de la producción



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

en la facultad-. El Organizador tiene como subordinados al Activista Funcionamiento, Activista Proceso Político, el Documentador y los Coordinadores de año. Los mismos llevan a cabo una serie de procesos que son importantes en el negocio ya que con la realización de los mismos es que puede funcionar el comité primario. El Activista Funcionamiento tiene que recoger todas las actas de **las reuniones que se realizan** en los distintos comités de base, para luego **hacer el cierre de funcionamiento**. El Activista Proceso Político tiene que **controlar las sanciones** una vez que un militante haya incurrido en alguna violación del reglamento de la UJC, debe de **llenar un modelo de sanción** el cual va a ser registrado en el expediente del militante; es el encargado de **hacer todo el proceso de crecimiento**, que no es más que los estudiantes que aspiran a ingresar a la UJC se someten a un riguroso proceso, el cual termina con el ingreso a las filas o la no aprobación y por último es el encargado de **llevar el proceso de evaluación a los militantes**. El documentador es el encargado de **llenar el ID2** que no es más que la Plantilla de control de la Militancia de la UJC y **recoge la cotización** que por mes entrega cada uno de los comités de base.

Anteriormente se han mencionado los principales procesos de negocio que se llevan a cabo dentro del comité primario de la UJC de la Facultad 2. En la realización de algunos de estos procesos se han detectado algunos problemas. Al terminar cada reunión del comité de base debe quedar plasmado en un acta todo lo debatido en la misma, estas actas siempre se entregan con retraso por el gran cumulo de información que contienen lo que hace que el Activista de Funcionamiento no pueda hacer el cierre hasta que no tenga la última acta de cada comité de base de la facultad. Se dificulta el control de las sanciones de los militantes y a la hora de dar la evaluación del año al militante se hace difícil saber si tiene sanción o no porque habría que buscar el expediente de cada militante para saber si tiene sanción y este es un proceso bastante tedioso y por último no se cuenta con un acceso rápido al ID2 ya que toda la información está guardada en Excel lo que demora la búsqueda de información acerca de los militantes, como por ejemplo la Dirección Particular, la Fecha de Ingreso a la Juventud u otra información que se necesite obtener.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.2. MODELO DE NEGOCIO

La Unión de Jóvenes Comunistas (UJC) es la organización política a la que pertenece gran parte de los jóvenes de la Facultad 2 de la UCI. Esta organización se rige por un reglamento donde se encierran una serie de estatutos y deberes que debe cumplir cada joven que pertenezca a esta organización. Las reuniones ordinarias de cada mes es el proceso más importante, donde se rinde cuenta y se evalúa el trabajo hasta ese momento, todo esto es recogido en un acta que queda como constancia de todos los puntos analizados en la reunión, cada comité de base cuenta con un secretario de actas encargado de llevar este proceso. En la universidad existe la particularidad de contar con 10 facultades, las cuales deben realizar después de recibidas las actas de cada C/B un Cierre de Funcionamiento por mes, en el cual se recoge el por ciento de asistencia, promedio de acuerdos, visitas del organismo superior, cantidad de intervenciones en la VIP, observaciones, total de justificado, análisis de los ausentes injustificado (si se analizaron o no), cantidad de injustificados, entre otras cuestiones. Para luego entregarlo al Comité Primario de la Universidad.

Actualmente en cada comité primario de la Facultad 2 existe un Activista de Funcionamiento encargado de confeccionar el Cierre de Funcionamiento, proceso que se lleva de forma manual, aumentando el esfuerzo humano y la probabilidad de errores. En la Facultad 2 se han detectado los siguientes problemas:

- No se cuenta con un formato general a la hora de entregar cualquier tipo de información relacionada con este proceso.
- Se producen demoras a la hora de entrega de las actas por parte de los comités de base de la facultad.

Por tanto para resolver esta problemática se propone el presente proyecto que consiste en realizar una Aplicación Web que permita informatizar el proceso de confección de las actas en la Unión de Jóvenes



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Comunistas y automatizar la confección del Cierre de Funcionamiento. Esta aplicación mejorará y permitirá agilizar este proceso en la Facultad 2. El sistema debe ser capaz de:

- Permitir a cada secretario de acta, desde cualquier máquina en la red, elaborar el acta basado en la plantilla permitiendo estandarizar el modelo.
- Mostrar un listado con todos los militantes por comité de base.
- Mostrar un listado con todos los comités de base y las actas correspondientes a cada mes.
- Confeccionar el Cierre de Funcionamiento.

Nota: En todos los casos se debe posibilitar la impresión de estos listados.

2.2.1. Actores y Trabajadores del Negocio

Actores del negocio	Justificación
Organizador	Controla todo el proceso de: hacer el Cierre de Funcionamiento, realizar las reuniones ordinarias y extraordinarias de los comité de base, confección del ID2, imposición de sanciones a los militantes, entre otros procesos de negocio que también controla. Todos los procesos anteriores los controla para que el comité primario no incumpla con estas tareas.
Sec. Comité Base (Secretario del Comité de Base)	Controla todo el proceso de realizar las reuniones ordinarias y extraordinarias de su comité de base, que sean convocadas por el Comité Primario.

Tabla 1: Descripción de los actores del negocio.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Trabajadores del negocio	Justificación
Act. Funcionamiento (Activista de Funcionamiento)	Es el encargado de elaborar correctamente y entregar el Cierre de Funcionamiento, además tiene que ver con todo el proceso de entrega de las actas de las reuniones.
Sec. Acta (Secretario de Acta)	Es el encargado de confeccionar correctamente y entregar el Acta de la reunión al Activista de Funcionamiento.
Documentador	Es el encargado de confeccionar el ID2 correctamente.
Act. Proceso Político (Activista de Proceso Político)	Es el encargado de confeccionar y registrar las sanciones que se les imponen a los militantes.

Tabla 2: Descripción de los trabajadores del negocio.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.3.2 Diagrama de Casos de Uso del Negocio

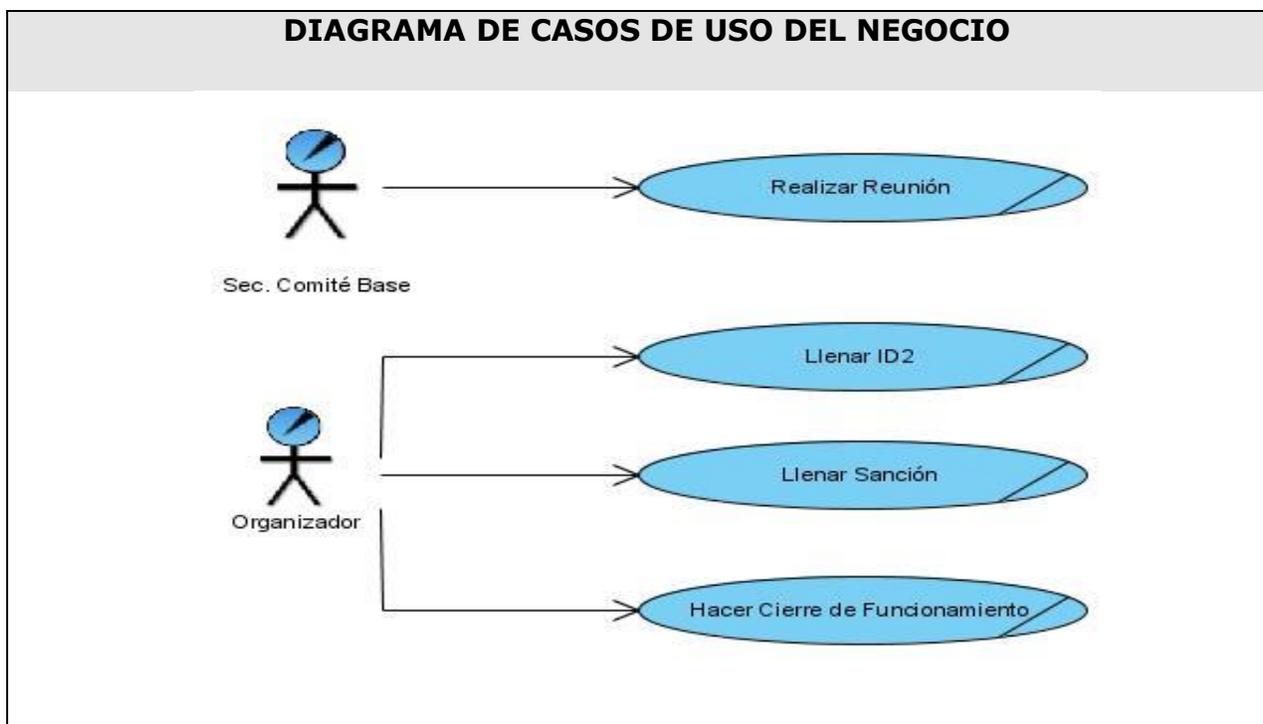


Figura 4: Diagrama de Casos de Uso del Negocio.

2.3. DESCRIPCIÓN TEXTUAL DE LOS CASOS DE USO DEL NEGOCIO

Describir textualmente los casos de uso del negocio tiene como objetivo principal mostrar cómo se realizan cada uno de ellos, es decir, como se llevan a cabo las actividades dentro del negocio y quiénes las realizan, desde que comienzan hasta que terminan y cómo interactúan con los actores.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Descripción Textual Casos de Uso del Negocio	
Caso de uso del Negocio:	Realizar Reunión.
Actor(es):	Sec. Comité Base (inicia).
Propósito: Llevar a cabo distintas reuniones ya sean ordinarias o extraordinarias.	
Resumen: El caso de uso se inicia cuando el Sec. Comité Base debe realizar las reuniones ordinarias o extraordinarias convocadas por el comité primario de la Facultad 2, quedando plasmado en un acta las cuestiones más importantes que allí se analizan, finalizando de esa forma el caso de uso.	
Flujo Normal de los Eventos:	
Acción del actor	Respuesta del negocio
1- Convoca a realizar la reunión.	2- El secretario de acta escribe en un documento llamado Acta todo lo debatido en la reunión. 3- El secretario de acta envía el acta de la reunión al Activista de Funcionamiento. 4- El Activista de Funcionamiento recibe el acta. 5- El Activista de Funcionamiento Revisa el acta, si la aprueba la registra, sino la envía de nuevo al Secretario de Acta señalándole los errores. 6- El Secretario de Acta corrige los errores y se la envía Activista de Funcionamiento, y se repite el proceso hasta que el acta este bien confeccionada. Finalizando de esa forma el caso de uso.
Prioridad: Responde al principal objetivo de automatización de nuestro problema, al resolver el	



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

problema principal del Comité Primario y de los Comité de Base de la facultad.
Mejoras: <ul style="list-style-type: none"> • Los procesos de las reuniones se automatizaran mediante una aplicación web. • El Activista de Funcionamiento podrá realizar consultas de todas las actas entregadas en la facultad 2.

Tabla 3: Descripción Textual del Caso de Uso Realizar Reunión.

Descripción Textual Casos de Uso del Negocio	
Caso de uso del Negocio:	Llenar ID2
Actor(es):	Organizador (inicia).
Propósito: Controlar el ID2 de los militantes de la Facultad 2	
Resumen: El caso de uso se inicia cuando el Organizador solicita al Documentador que es el encargado de elaborar el ID2, que le entregue el ID2 de cada comité de base de la Facultad 2, de esta forma el documentador confecciona y entrega el ID2. Finalizando de esa forma el caso de uso.	
Flujo Normal de los Eventos:	
Acción del actor	Respuesta del negocio
1- Pide al documentador que le entregue el ID2	2- El Documentador recolecta toda la información que lleva el ID2. 3- El Documentador confecciona el ID2. 4- El Documentador envía el ID2.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

5- Recibe el ID2.	
6- Registra el ID2.	
Prioridad: Responde a uno de los objetivos de automatización de nuestro problema al resolver uno de los problemas del Comité Primario de la facultad.	
Mejoras:	
<ul style="list-style-type: none"> • Los resúmenes (ID2) se harán de forma digital. • El Documentador podrá realizar consultas de todos los resúmenes ID2 de cada uno de los comités de base de la facultad. 	

Tabla 4: Descripción Textual del Caso de Uso Llenar ID2.

Descripción Textual Casos de Uso del Negocio	
Caso de uso del Negocio:	Hacer Cierre de Funcionamiento
Actor(es):	Organizador (inicia).
Propósito: Controlar la entrega del Cierre de Funcionamiento.	
Resumen: El caso de uso se inicia cuando el Organizador solicita al Activista de Funcionamiento que es el encargado de elaborar el Cierre de Funcionamiento, que le entregue el Cierre, este tiene que tener todas las actas bien confeccionadas de los distintos comités de base, de esta forma el Activista de Funcionamiento confecciona y entrega el Cierre de Funcionamiento. Finalizando de esa forma el caso de uso.	
Casos de Usos Asociados:	Realizar Reunión. (Después que se hace este caso de uso - Realizar Reunión -, en cada uno de los Comité de Base de la Facultad 2 y el Activista de Funcionamiento le llegan todas las actas bien confeccionadas, es que se ejecuta este caso de uso - Hacer Cierre de Funcionamiento -, es decir, después que el Activista de Funcionamiento tiene todas las actas recogidas es que puede hacer el cierre de funcionamiento, que se hace mensualmente).



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Flujo Normal de los Eventos:	
Acción del actor	Respuesta del negocio
1- Pide al Activista de Funcionamiento que le entregue el Cierre. 5- Recibe el Cierre. 6- Registra el Cierre.	2- El Activista de Funcionamiento consulta la información de las actas registradas. 3- El Activista de Funcionamiento confecciona el Cierre. 4- Envía el Cierre al Organizador.
Prioridad: Responde a uno de los objetivo de automatización de nuestro problema al resolver uno de los problemas del Comité Primario de la facultad.	
Mejoras: <ul style="list-style-type: none"> • El Cierre de Funcionamiento se hará de forma digital. • El Activista de Funcionamiento podrá realizar consultas de todas las actas entregadas en la facultad. 	

Tabla 5: Descripción Textual del Caso de Uso Hacer Cierre de Funcionamiento.

Descripción Textual Casos de Uso del Negocio	
Caso de uso del Negocio:	Llenar Sanción
Actor(es):	Organizador (inicia).



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Propósito: Controlar las sanciones que se le imponen en la Facultad 2 a los militantes.	
Resumen: El caso de uso se inicia cuando el Organizador solicita al Activista de Proceso Político que es el encargado de controlar las sanciones, que le entregue las sanciones de cada comité de base, de esta forma el Activista de Proceso Político confecciona y entrega las sanciones, finalizando de esa forma el caso de uso.	
Flujo Normal de los Eventos:	
Acción del actor	Respuesta del negocio
1- Pide al Activista de Proceso Político que le entregue las sanciones 5- Recibe el documento. 6- Registra el documento.	2- Recolecta todas las sanciones impuestas en la Facultad 2. 3- Confecciona un documento con todas ellas. 4- Envía el documento al Organizador.
Prioridad: Responde a uno de los objetivo de automatización de nuestro problema al resolver uno de los problemas del Comité Primario de la facultad.	
Mejoras:	
<ul style="list-style-type: none"> • Las sanciones se guardarán de forma digital. • El Activista de Proceso Político podrá realizar consultas de todas las sanciones de cada uno de los militantes de la Facultad 2. 	

Tabla 6: Descripción Textual del Caso de Uso Llenar Sanción.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.4. DIAGRAMAS DE ACTIVIDAD

Luego de realizar la descripción de los casos de uso del negocio se pueden hacer los diagramas de actividades que muestran el flujo de los procesos de manera gráfica. A través de las calles, se especifican las actividades que deben realizar los actores y trabajadores del negocio, y a través del envío de objetos cómo se utilizan las entidades del negocio.

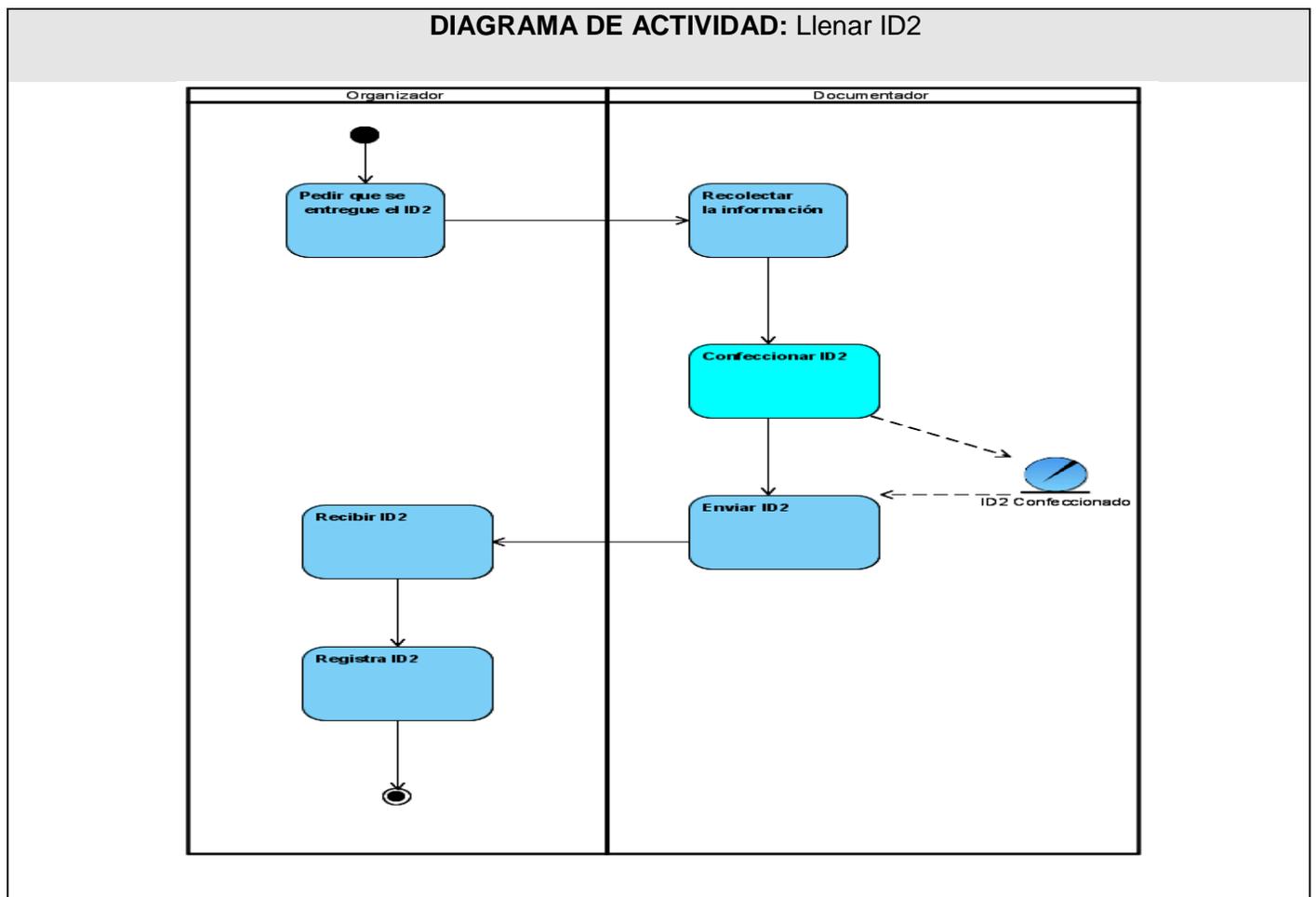


Figura 5: Diagrama de actividad del CUN Llenar ID2.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

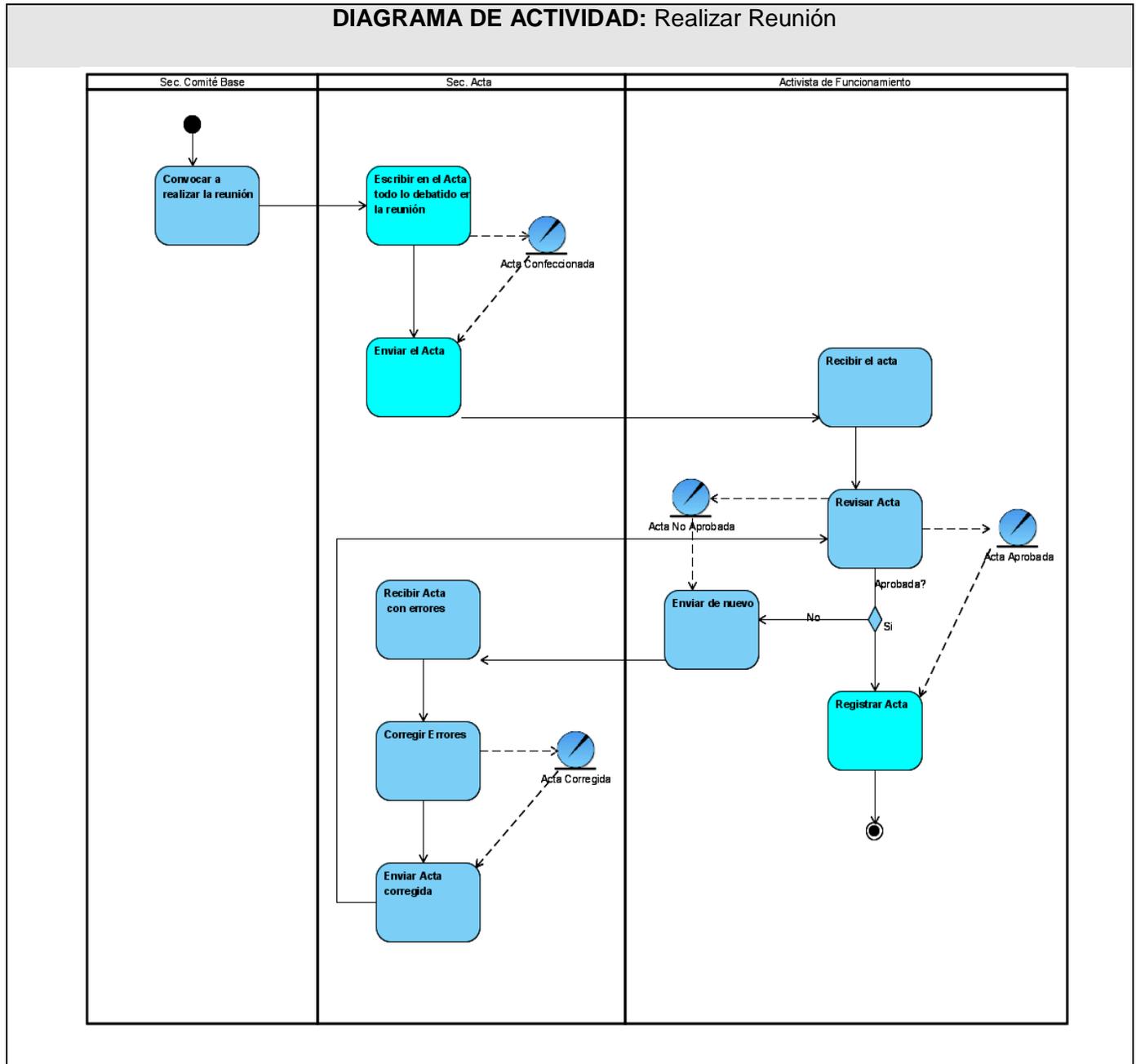


Figura 6: Diagrama de Actividad del CUN Realizar Reunión.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

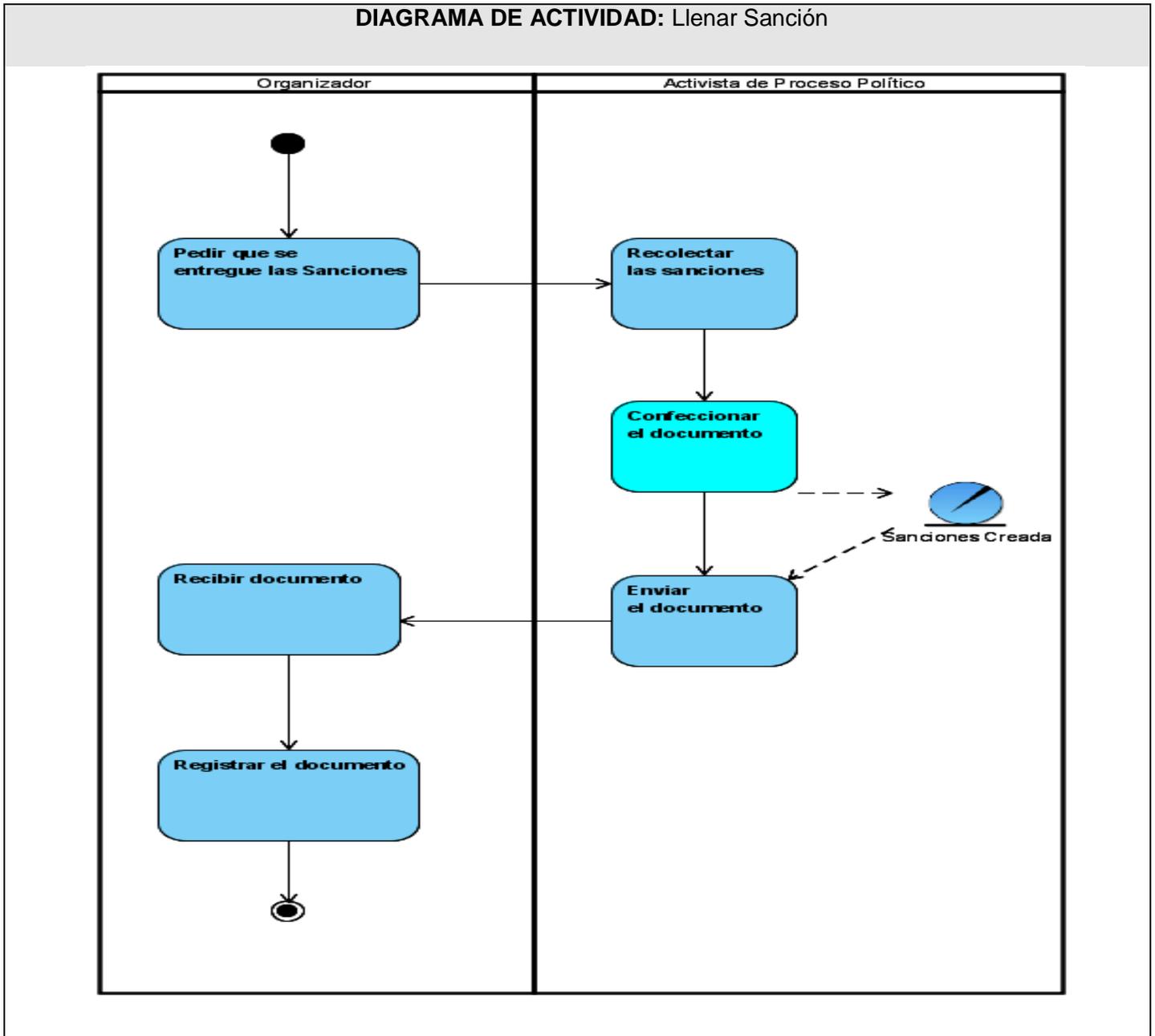


Figura 7: Diagrama de Actividad del CUN Llenar Sanción.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

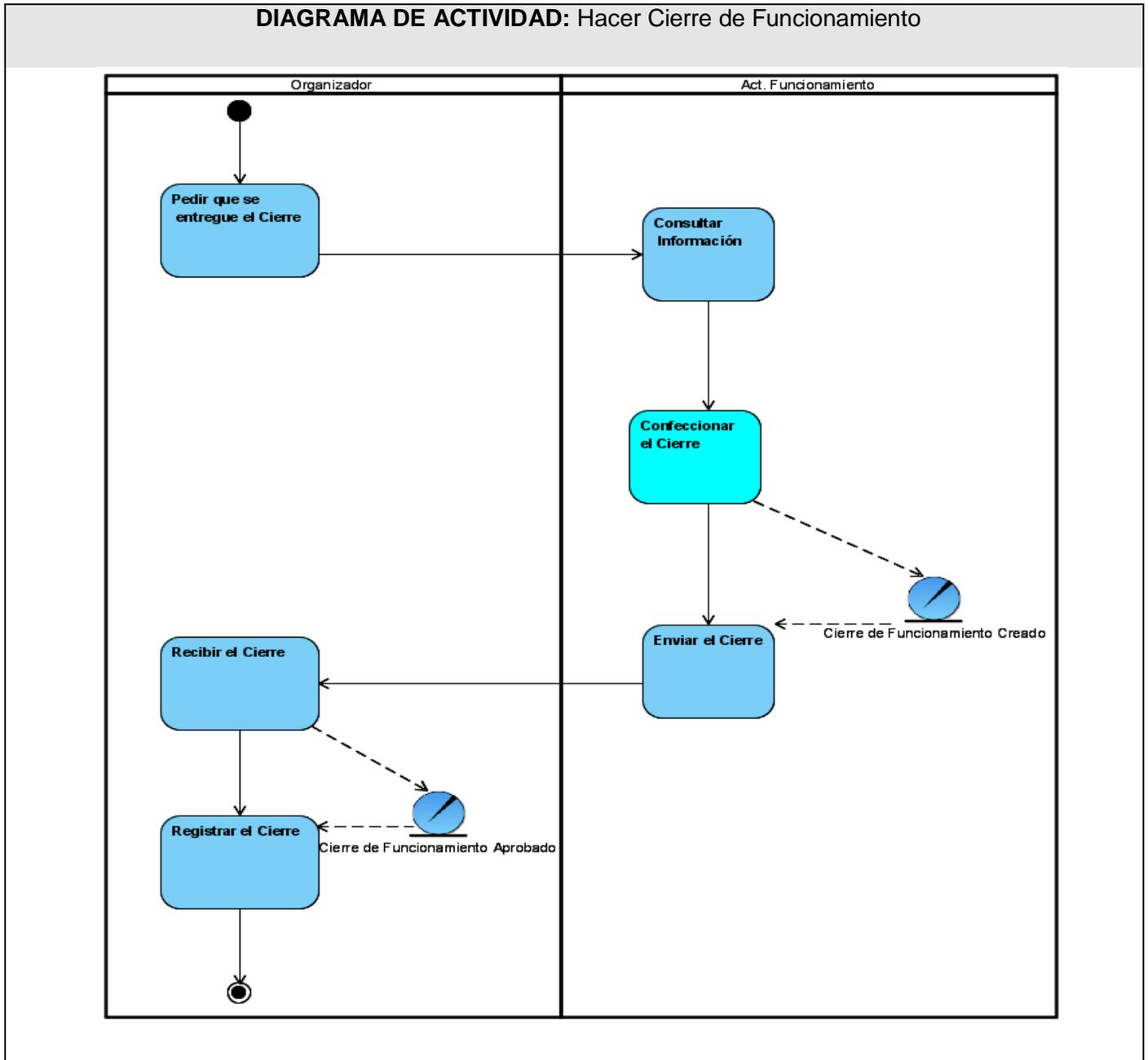


Figura 8: Diagrama de Actividad del CUN Hacer Cierre de Funcionamiento.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.5. MODELO DE OBJETOS

El modelo de objetos representa gráficamente con cuales entidades del negocio se relacionan los trabajadores del negocio.

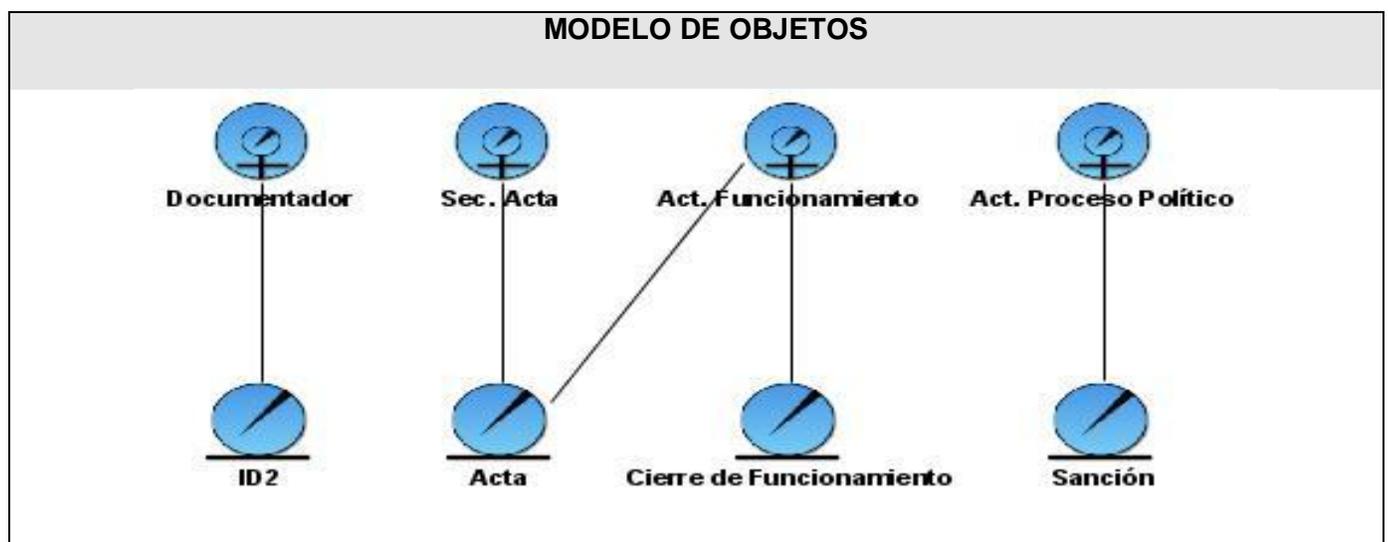


Figura 9: Modelo de objetos.

2.6. REGLAS DEL NEGOCIO

- Los miembros del comité de la juventud de la UCI deben convocar a los militantes a una reunión por brigada para estructurar el comité de base de cada una de ellas; después de este proceso queda constituido el Comité de Base, es decir cada miembro del C/B se la va a asignar un cargo.
- Para realizar una reunión del comité de base o del comité primario debe estar presente más del 96% de los convocados.
- Para Constituir un comité de base debe haber más de tres militantes matriculados en el.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

- Los secretarios de actas deberán revisar y entregar las actas al Comité Primario antes de cumplirse las 72 horas después de haber sido realizada la reunión por el Secretario General del Comité de Base.
- Las actas deberán utilizar el formato especificado por el organismo superior.
- El Documentador del comité primario debe confeccionar el ID2 correctamente.
- La información del ID2 solo debe de ser conocimiento del personal autorizado.

2.7. RELACIÓN DE LOS REQUERIMIENTOS DEL SISTEMA

Los requerimientos funcionales son aquellas capacidades o condiciones que el sistema debe cumplir. Ellos definen qué es lo que el sistema debe hacer y permiten identificar las funcionalidades requeridas. No alteran la funcionalidad del producto, es decir, los requerimientos funcionales se mantienen invariables sin importar con que propiedades o cualidades se relacionen.

2.7.1 Listado de los requerimientos funcionales enumerados

RF1- Permitir Autenticarse

1.1- Autenticar Usuario

RF2- Gestionar Usuario

2.1- Insertar usuario

2.2- Modificar usuario

2.3- Mostrar usuarios C/P

2.4- Mostrar usuarios de los C/B



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.5- Eliminar usuario

2.6- Cambiar Secretario del C/P

RF3- Gestionar Comité de Base

3.1- Insertar Comité de Base

3.2- Modificar Comité de Base

3.3- Mostrar Comité de Base

3.4- Eliminar Comité de Base

3.5- Ver Comité de Base

RF4- Gestionar Acta

4.1- Registrar asistencia

4.2- Insertar Acta

4.3- Modificar acta

4.4- Mostrar Acta

4.5- Eliminar Acta

4.6- Ver Acta

4.7- Ver Actas

RF5- Gestionar ID2

5.1- Insertar ID2



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

5.2-Modificar ID2

5.3-Mostrar ID2

RF6- Gestionar Sanción

6.1-Insertar Sanción

6.2-Modificar Sanción

6.3-Mostrar Sanción

6.4-Eliminar Sanción

RF7- Emitir Reporte

7.1- Listar militantes por cada comité de base

7.2- Listar actas registradas por cada mes en cada comité de base

7.3- Listar los ausentes a las reuniones mensualmente

7.4- Ordenar mensualmente los comités de base por porciento de asistencia

RF8-Permitir Generar Cierre de Funcionamiento

8.1- Generar Cierre de Funcionamiento



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.7.2 Definición de los requerimientos no funcionales:

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Los requerimientos no funcionales especifican propiedades del sistema como dependencias de plataformas, mantenibilidad y confiabilidad. Son esas características que posibilitan que el producto sea atractivo, usable, rápido, confiable, etc. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto.

- **Apariencia o interfaz externa.**

- Estará diseñado para resolución de 1024 por 768 píxeles, aunque se podrá ver en 800x600.
- La interfaz debe ser agradable para el usuario, que los iconos estén en correspondencia con lo que representan y que combine correctamente los colores, tipo de letra y tamaño.
- La navegabilidad debe ser sencilla.

- **Usabilidad.**

- Garantizar un acceso fácil y rápido a los usuarios. El sistema podrá ser usado por cualquier persona que posea conocimientos básicos de computación y trabajo en la Web.
- El sistema debe permitir el acceso concurrente de diferentes usuarios, en dependencia del nivel de usabilidad que este definida para cada uno.

- **Rendimiento.**

- El sistema deberá ser capaz de gestionar toda la información y dar respuesta a las solicitudes lo más rápido posible.
- Las páginas Web no deben tardar más de 15 segundos en cargarse.

- **Soporte.**

- El sistema debe ser de fácil instalación, mantenimiento y configuración.
- Se requiere de la instalación de un servidor Web con el servicio de interpretación de códigos con PHP 5.
- Contar con una etapa de pruebas para eliminar la mayor cantidad de errores.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

- **Portabilidad.**

El sistema funcionará en las plataformas Linux y Windows.

- **Seguridad**

- Se podrá acceder al sistema solamente después de autenticarse.
- Se garantizará que las funcionalidades de los distintos subsistemas 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 cargo autorizado para ello.
- Cumplir con la confidencialidad de los datos, permitiendo que cada miembro del Comité Primario de la UJC de la Facultad 2 solo pueda ver sus datos.
- Tiene que ser capaz de anular cualquier acción incorrecta que atente contra la integridad de los datos.

- **Políticos-culturales.**

- El sistema solo podrá ser utilizado en la Facultad 2 de la UCI.
- Se empleará en la aplicación 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 C/P de la Facultad 2.
- El sistema deberá respetar los términos empleados normalmente por los miembros de la UJC.

- **Legales.**

- La mayoría de las herramientas de desarrollo son libres y el resto, las licencias están avaladas.
- Documentación legal de uso como Declaración de Autoría.

- **Confiabilidad.**

- Validar la captación de datos para evitar entradas inadecuadas.
- Los reportes que se obtendrán deberán ser 100% reales y precisos.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

- Chequeo constante de la integridad y consistencia en los datos.

• Software.

- Navegador Mozilla Firefox v3.0.
- Servidor Web Apache v2.2 o superior con módulo PHP v5.2.3 disponible, este debe estar configurado con las extensiones “ldap”, “soap” y “pgsql”.
- Como SGBD: PostgreSQL preferiblemente v8.3 en adelante.

• Hardware para la PC Cliente.

- PC servidor: 20 GB disco duro.
- Microprocesador de 3.00 GHz.
- 256 MB mínimo de RAM.
- Tarjeta de red.

• Hardware para la PC Servidor.

- PC servidor: 30 GB disco duro.
- Microprocesador de 3.0 GHz..
- 2.5 GB mínimo de RAM.
- Tarjeta de red.

• Restricciones del diseño e implementación.

- Utilizar como lenguaje del lado del servidor al PHP 5 o superior y del lado del cliente: HTML y JavaScript.
- Para la programación en PHP se recomienda el IDE: Zend Studio for Eclipse -6.1.0
- Se recomienda el uso de la arquitectura MVC.
- El código deberá ser reutilizable.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.8. MODELO DE CASOS DE USO DEL SISTEMA.

2.8.1 Definición de los actores del sistema a automatizar:

Actores	Justificación
Sec. Comité Primario (Secretario del Comité Primario)	A partir de las actas entregadas por los Sec. de actas y una vez revisadas por el Activista de Funcionamiento que realiza el resumen de las mismas (Cierre de Funcionamiento), es el encargado de dar parte de como se comportó la facultad en el mes analizado a los organismo superiores, además de ser el encargado de administrar a los usuarios del sistema, es decir insertar, eliminar, y modificar un usuario, así como administrar el sistema.
Act. Funcionamiento (Activista de Funcionamiento)	A partir de las actas entregadas por los Secretarios de acta, después de revisadas se realiza un resumen de las mismas (Cierre de Funcionamiento).
Sec. Acta (Secretario de Acta)	Es el encargado de elaborar las actas de las reuniones del comité de base.
Documentador	Es el encargado de registrar en el sistema el ID2.
Act. Proceso Político (Activista de Proceso Político)	Es el encargado de registrar las sanciones en el sistema.
Militante	Puede ver las actas de su comité de base.
Usuario	Es un usuario que generaliza el rol de autenticación al sistema.

Tabla 7: Definición de los actores del sistema.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.8.2 PAQUETES Y RELACIÓN ENTRE ELLOS.

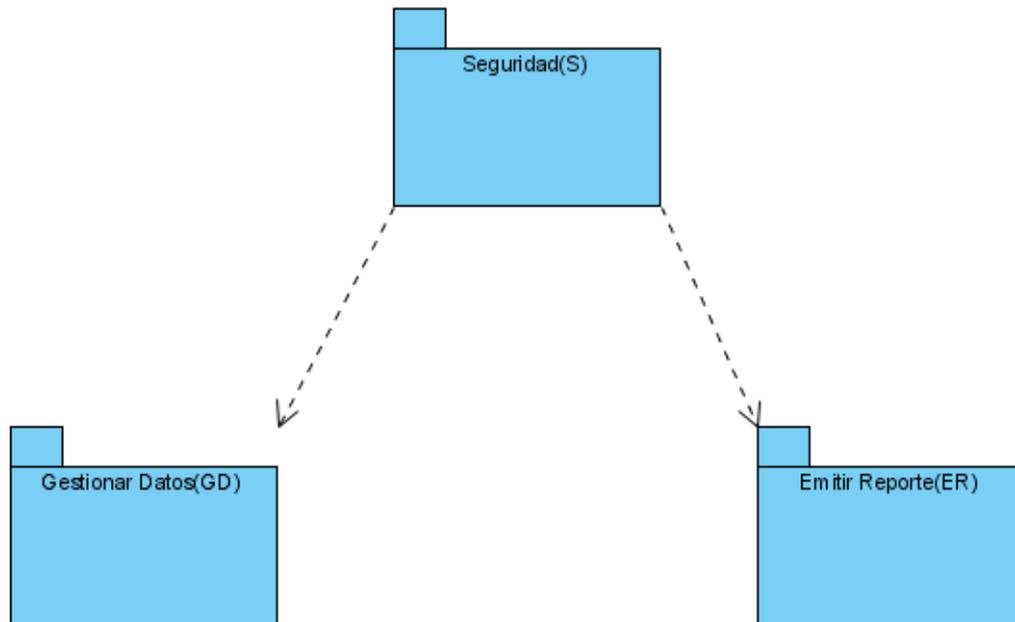


Figura 10: Diagrama de Paquete.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.8.3 DIAGRAMA DE CASOS DE USO DEL SISTEMA A AUTOMATIZAR.

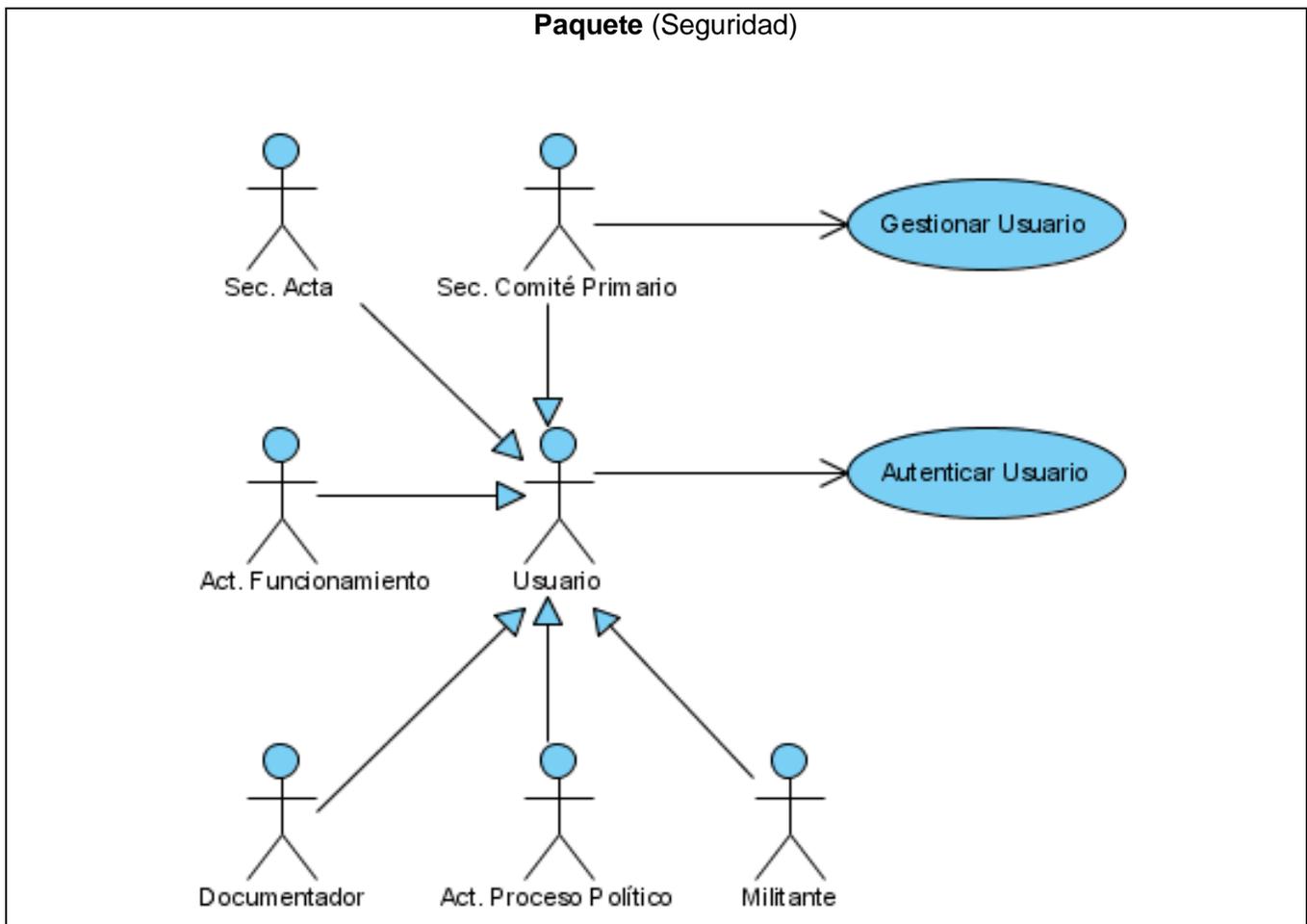


Figura 11: Diagrama de Casos de Uso del Sistema del paquete de Seguridad.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

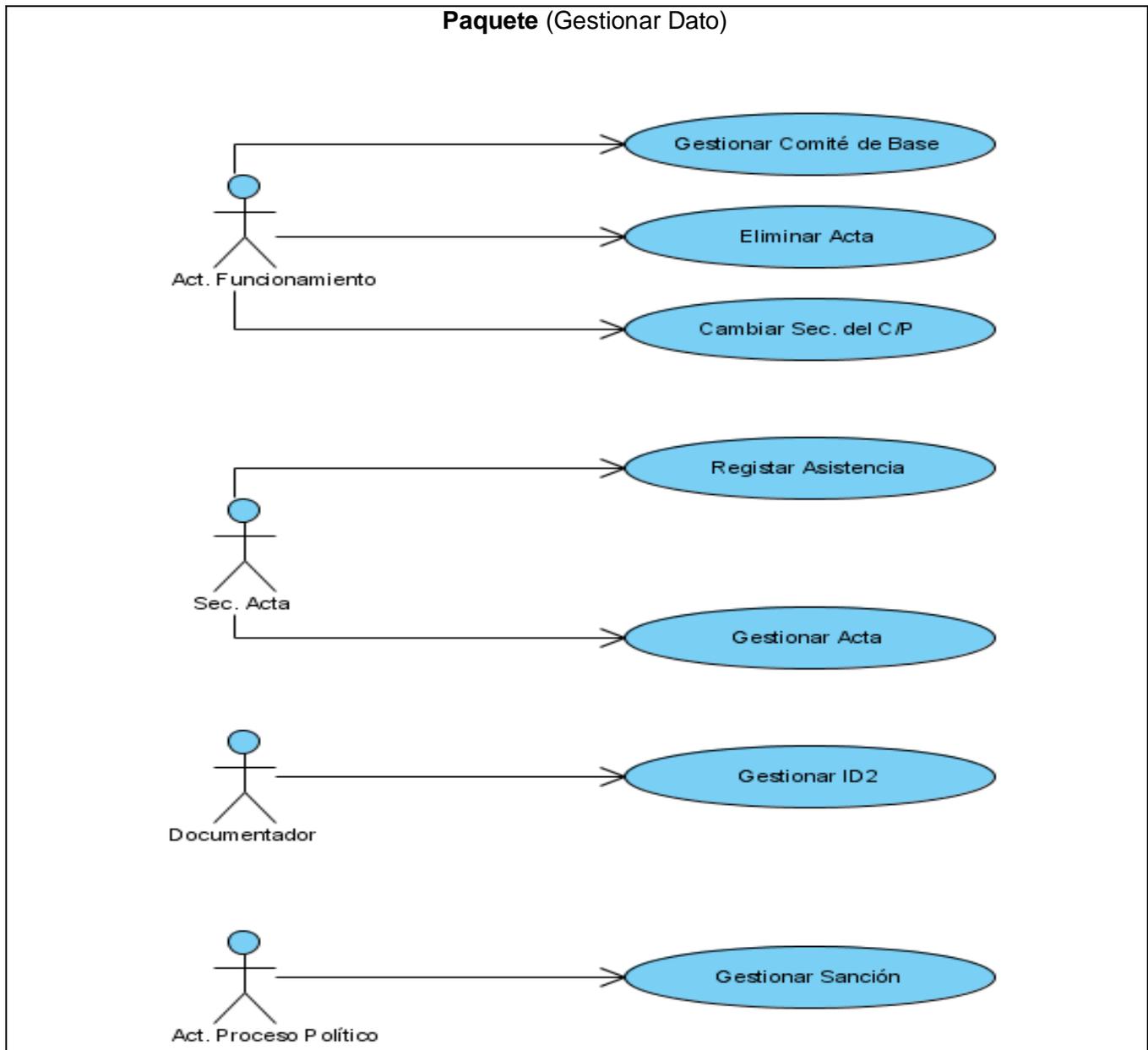


Figura 12: Diagrama de Casos de Uso del Sistema del paquete de Gestionar Dato.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

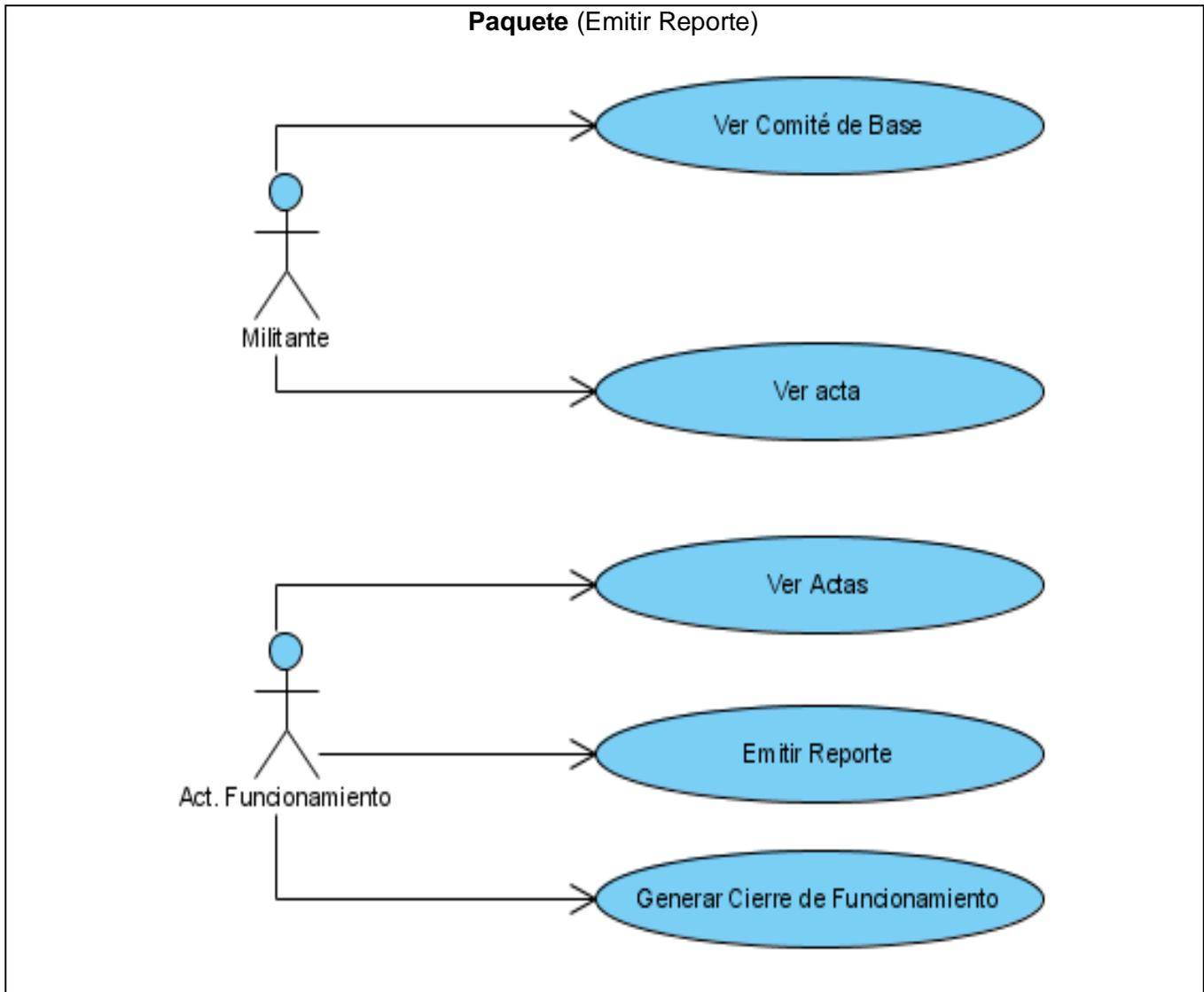


Figura 13: Diagrama de Casos de Uso del Sistema del paquete de Emitir Reporte.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.8.4 DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA.

La descripción de los casos de uso del sistema detalla las acciones que se llevan a cabo durante la interrelación actor-sistema, es decir, describe las actividades que realiza el actor al hacer uso del sistema y las correspondientes respuestas del mismo, entre otras cuestiones que se tienen en cuenta.

Nombre del caso de uso:	Autenticar Usuario
Actor(es):	Usuario (inicia)
Propósito:	Permitir el acceso de los usuarios al sistema.
Resumen:	El caso de uso inicia cuando el Usuario necesita entrar al sistema. Finaliza el caso de uso cuando el sistema permite el acceso.
Referencias:	R1.1
Precondiciones:	El Usuario debe estar registrado en la base de datos del sistema.
Poscondiciones:	El Usuario accede al sistema, entrando en los lugares permitidos.
Curso Normal de Eventos:	
Acciones del Actor	Respuesta del Sistema
1- El Usuario inicia el sistema.	1.1-El sistema muestra la interfaz de autenticación de usuario.
2-El Usuario introduce su identificador y contraseña.	2.1- El sistema verifica que el identificador y contraseña sean correctos. 2.2- El sistema permite el acceso.
Curso Alterno:	
	2.1.1 Si el campo de identificador y contraseña están vacíos. El sistema muestra un mensaje diciendo: "El campo usuario no puede estar vacío". Regresa al paso (3) del flujo normal de eventos. 2.1.2 Si el campo de identificador está vacío y el de la contraseña no está vacío. El sistema muestra un mensaje diciendo: "El campo usuario no puede estar vacío". Regresa al paso (3) del flujo normal de eventos. 2.1.3 Si el campo de identificador no está vacío y el de la contraseña está



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	<p>vacío. El sistema muestra un mensaje diciendo: “El campo de contraseña no puede estar vacío”. Regresa al paso (3) del flujo normal de eventos.</p> <p>2.1.4 Si el identificador o la contraseña están escritos mal. El sistema muestra un mensaje diciendo: “Verifique usuario o contraseña”. Regresa al paso (3) del flujo normal de eventos.</p> <p>2.1.5 Si el identificador está escrito mal o está escrito bien y no está registrado en la base de datos del sistema. El sistema muestra un mensaje diciendo: “Su usuario no está registrado en la base de datos, contacte con el Sec. C/P o con el Activista. de Funcionamiento de la Facultad 2”. Regresa al paso (3) del flujo normal de eventos.</p>
Prioridad:	Crítico

Tabla 8: Descripción del caso de uso del sistema Autenticar Usuario.

Nombre del caso de uso:	Gestionar Comité Base
Actor(es):	Act. Funcionamiento (inicia)
Propósito:	Garantizar el control de los comités de base.
Resumen:	El caso de uso se inicia cuando el Activista de Funcionamiento accede a la interfaz de Gestionar Comité de Base y selecciona la acción que desea realizar, ya sea insertar, modificar, mostrar o eliminar un comité de base y termina el caso de uso con la ejecución de dicha acción.
Referencias:	R3.1 R3.2 R3.3 R3.4
Precondiciones:	El Activista de Funcionamiento debe estar registrado en la base de datos del sistema con el Cargo de Activista de Funcionamiento, en el caso de que desee eliminar, modificar o mostrar un comité de base determinado, este debe encontrarse registrado.
Poscondiciones:	Queda actualizado el estado de los comités de base en el sistema, se registra un nuevo comité de base, se actualizan los datos de un comité de base, se muestra un comité de base o se elimina un comité de base.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Curso Normal de Eventos:	
Acciones del Actor	Respuesta del Sistema
1-El Activista de Funcionamiento accede a la interfaz de Gestión de Comité de Base. 2-El Activista de Funcionamiento elige la acción a realizar.	1.1- El sistema muestra una serie de acciones a realizar. 2.1- Si elige a) Crear un nuevo Comité Base ir a la sección “Crear Comité de Base” b) Modificar un Comité Base ir a la sección “Modificar Comité de Base” c) Mostrar un Comité Base ir a la sección “Mostrar Comité de Base” d) Eliminar un Comité Base ir a la sección “Eliminar Comité de Base”
Sección “Crear Comité de Base”	
2- El Activista de Funcionamiento escoge la brigada para crear el nuevo Comité de Base. 3- El Activista de Funcionamiento introduce los datos para crear al nuevo Comité de Base.	1- El sistema muestra la interfaz de crear Comité de Base, mostrando un formulario con las brigadas de la Facultad 2 que se deben introducir para crear el comité de base de la brigada. 2.1- El sistema verifica que se haya seleccionado una brigada. 2.2- El sistema muestra los datos de la brigada seleccionada. 3.1- El sistema verifica que los campos estén llenos. 3.2- El sistema verifica que no exista ese comité de base en el sistema. 3.3- El sistema crea el nuevo comité de base añadiéndolo a la base de datos del sistema, le informa al Activista de Funcionamiento que se ha efectuado el registro y finaliza el caso de uso.
Curso Alterno	
	2.1 El sistema envía un mensaje informativo se debe escoger una brigada. Regresa al paso (2) del flujo normal de eventos. 3.1 El sistema envía un mensaje solicitando llenar los campos. Regresa al paso (3) del flujo normal de eventos. 3.2 Si el comité de base está registrado se envía un mensaje informativo. Regresa al paso (3) del flujo normal de eventos.
Sección “Modificar Comité de Base”	
	1- El sistema muestra la interfaz de modificar Comité de Base, mostrando un formulario con los comités de base de la Facultad 2 que se deben



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

<p>2- El Activista de Funcionamiento selecciona el comité de base que desea modificar.</p> <p>3- El Activista de Funcionamiento realiza los cambios pertinentes.</p>	<p>introducir para modificarlos.</p> <p>2.1- El sistema verifica se haya seleccionado un comité de base. 2.2- El sistema muestra la información del comité de base seleccionado.</p> <p>3.1- El sistema verifica que los campos obligatorios estén llenos. 3.2- El sistema verifica que el cargo seleccionado sea uno, si no es militante. 3.3- El sistema guarda los datos modificados del comité de base en la base de datos, le informa al Activista de Funcionamiento que se ha modificado el comité de base y finaliza el caso de uso.</p>
Curso Alterno	
	<p>2.1 El sistema envía un mensaje solicitando escoger un comité de base. Regresa al paso (2) del flujo normal de eventos.</p> <p>3.1 El sistema envía un mensaje solicitando llenar todos los campos. Regresa al paso (2) del flujo normal de eventos. 3.2 El sistema envía un mensaje informativo que el cargo que es distinto a militantes debe ser uno solo. Regresa al paso (2) del flujo normal de eventos.</p>
Sección “Mostrar Comité de Base”	
<p>2- El Activista de Funcionamiento escoge el comité de base que desea mostrar.</p>	<p>1- El sistema muestra la interfaz de mostrar Comité de Base, mostrando un formulario con los comités de base de la Facultad 2 que se desean mostrar.</p> <p>2.1- El sistema verifica que haya seleccionado un comité de base. 2.2- El sistema muestra la información del comité de base seleccionado.</p>
Curso Alterno	
	<p>2.1 El sistema envía un mensaje solicitando escoger un comité de base. Regresa al paso (2) del flujo normal de eventos.</p>
Sección “Eliminar Comité de Base”	
	<p>1- El sistema muestra la interfaz de eliminar Comité de Base, mostrando un formulario con los comités de base de la Facultad 2 que se desean eliminar.</p>



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2- El Activista de Funcionamiento elige el comité de base a eliminar.	2.1- El sistema verifica que haya seleccionado un comité de base. 2.2- El sistema elimina el comité de base seleccionado, le informa al Activista de Funcionamiento que se ha eliminado el comité de base y finaliza el caso de uso.
Curso Alterno	
	2.1 El sistema envía un mensaje solicitando escoger un comité de base. Regresa al paso (2) del flujo normal de eventos.
Prioridad:	Crítico

Tabla 9: Descripción del caso de uso del sistema Gestionar Comité Base.

Nombre del caso de uso:	Registrar Asistencia
Actor(es):	Sec. Acta (inicia)
Propósito:	Registrar la asistencia a las reuniones ordinarias del comité de base.
Resumen:	El caso de uso se inicia cuando el Secretario de Acta accede al sistema y escoge la opción Asistencia. Finaliza cuando el sistema ha registrado la asistencia a la reunión.
Referencias:	R4.1
Precondiciones:	El usuario debe estar registrado como Secretario de Acta en la base de datos del sistema y estar creado el comité de base al cual él pertenece.
Poscondiciones:	Queda registrada mensualmente por el sistema la asistencia a la reunión ordinaria.
Curso Normal de Eventos:	
Acciones del Actor	Respuesta del Sistema
1-El Activista de Funcionamiento escoge la opción Asistencia.	1.1- El sistema muestra un listado con los militantes del comité de base del secretario de acta.
2- El Activista de Funcionamiento desmarca a los ausentes y escoge la opción aceptar.	2.1- El sistema verifica que el porcentaje de asistencia sea el correcto para realizar la reunión. 2.2- El sistema registra la asistencia. 2.2- El sistema crea el formulario para crear el acta y finaliza el caso de



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	USO.
Curso Alterno	
	2.1 El sistema envía un mensaje informativo el porcentaje de asistencia no es correcto para realizar la reunión. Regresa al paso (2) del flujo normal de eventos.
Prioridad:	Crítico

Tabla 10: Descripción del caso de uso del sistema Registrar Asistencia.

Nombre del caso de uso:	Gestionar Acta
Actor(es):	Sec. Acta(inicia)
Propósito:	Garantizar la gestión y el control de las actas que se almacenan en la base de datos del sistema.
Resumen:	El caso de uso se inicia cuando el Secretario de Acta accede a la interfaz de Gestionar Acta y selecciona la acción que desea realizar, ya sea crear, modificar o mostrar un acta y termina el caso de uso con la ejecución de dicha acción.
Referencias:	R4.2 R4.3 R4.4
Precondiciones:	El usuario debe estar registrado como Secretario de Acta en la base de datos del sistema, en el caso de que se desee crear un acta tiene que pasar la asistencia de dicha reunión y en el caso de modificar o mostrar un acta determinada ya debe encontrarse registrada en la base de datos del sistema.
Poscondiciones:	Queda actualizado el estado de las actas que se encuentran almacenadas en la base de datos del sistema, se registra una nueva acta, se actualizan los datos de la misma y se muestra.
Curso Normal de Eventos:	
Acciones del Actor	Respuesta del Sistema
1-El Secretario de Acta accede a la interfaz de Gestión de Acta. 2-El Secretario de Acta elige la acción a realizar.	1.1- El sistema muestra una serie de acciones a realizar. 2.1- Si elige a) Crear un Acta ir a la sección "Crear Acta".



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	<p>b) Modificar un Acta ir a la sección “Modificar Acta”.</p> <p>c) Mostrar un Acta ir a la sección “Mostrar Acta”.</p>
Sección “Crear Acta”	
<p>2- El Secretario de Acta introduce los datos para registrar el Acta y escoge la opción Crear Acta.</p>	<p>1- El sistema muestra el formulario de crear acta, mostrando los campos generales que se deben introducir en el transcurso de la reunión.</p> <p>2.1- El sistema pide confirmación de creación del acta.</p> <p>2.2- El sistema confirma que el usuario haya entrado los datos necesarios.</p> <p>2.3- El sistema introduce los datos correspondientes del acta en la base de datos y envía un mensaje informativo al usuario y finaliza el caso de uso.</p>
Curso Alterno	
	<p>2.1 Regresa al Paso (2) del flujo normal de eventos.</p> <p>2.2 El sistema muestra un mensaje señalando los campos que faltan por llenar. Regresa al paso (2) del flujo normal de eventos.</p>
Sección “Modificar Acta”	
<p>2- El Secretario de Acta realiza los cambios pertinentes.</p>	<p>1- El sistema muestra el formulario del acta del mes en curso con sus datos correspondientes y aclarando que solo tiene 72 horas para modificar el acta después de la fecha de realización de la reunión.</p> <p>2.1- El sistema pide confirmación de modificación del acta.</p> <p>2.2- El sistema guarda los cambios realizados en la base de datos del sistema y le envía un mensaje informativo al mismo y finaliza el caso de uso.</p>
Curso Alterno	
	<p>2.1 Regresa al Paso (2) del flujo normal de eventos.</p>
Sección “Mostrar Acta”	
<p>2- El Secretario de Acta selecciona el acta que desea mostrar.</p>	<p>1- El sistema muestra un listado que tiene la fecha de realización de las actas existentes en la base de datos que corresponden al comité de base del secretario de acta.</p> <p>2.1- El sistema muestra la información del acta seleccionada y así finaliza el caso de uso.</p>
Prioridad:	Crítico



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Tabla 11: Descripción del caso de uso del sistema Gestionar Acta.

Nombre del caso de uso:	Gestionar Sanción
Actor(es):	Act. Proceso Político (inicia)
Propósito:	Garantizar la gestión y el control de las sanciones que se almacenan en la base de datos del sistema.
Resumen:	El caso de uso se inicia cuando el Activista de Proceso Político accede a la interfaz de Gestionar Sanción y selecciona la acción que desea realizar, ya sea crear, modificar, mostrar o eliminar una sanción y termina el caso de uso con la ejecución de dicha acción.
Referencias:	R6.1 R6.2 R6.3 R6.4
Precondiciones:	El usuario debe estar autenticado como Activista de Proceso Político en la base de datos del sistema, en el caso de que se desee modificar, mostrar o eliminar una sanción determinada, esta debe encontrarse registrada en la base de datos del sistema.
Poscondiciones:	Queda actualizado el estado de las sanciones que se encuentran almacenadas en la base de datos del sistema, se crea una nueva sanción, se modifican y se muestran los datos de una sanción, y se elimina una sanción.
Curso Normal de Eventos:	
Acciones del Actor	Respuesta del Sistema
1-El Activista de Proceso Político accede a la interfaz de Gestión de Sanciones. 2-El Activista de Proceso Político elige la acción a realizar.	1.1- El sistema muestra una serie de acciones a realizar. 2.1- Si elige a) Crear una nueva sanción ir a la sección “Crear Sanción” b) Modificar una sanción ir a la sección “Modificar Sanción” c) Mostrar una sanción ir a la sección “Mostrar Sanción” d) Eliminar un sanción ir a la sección “Eliminar Sanción”



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Sección “Crear Sanción”	
<p>2- El Activista de Proceso Político escoge el comité de base del cual desea sancionar un militante.</p> <p>3- El Activista de Proceso Político escoge al militante que desea sancionar.</p> <p>4- El Activista de Proceso Político introduce los datos para registrar la nueva sanción.</p>	<p>1-El sistema muestra la interfaz de crear sanción, mostrando un formulario con los comités de base que están registrados en la base de datos.</p> <p>2.1- El sistema verifica que haya seleccionado un comité de base. 2.2-El sistema muestra un listado con los militantes del comité de base seleccionado.</p> <p>3.1 El sistema pide la confirmación de si desea sancionar a ese militante. 3.2- El sistema muestra un formulario con los campos necesarios para registrar la sanción.</p> <p>4.1- El sistema verifica la validez de los datos introducidos. 2.2- El sistema pide la confirmación de que si desea guardar la sanción. 4.3- Registra la nueva sanción en la base de datos del sistema, le informa al Activista de Proceso Político que se ha efectuado el registro y finaliza el caso de uso.</p>
Curso Alterno	
	<p>2.1- El sistema envía un mensaje solicitando escoger un comité de base. Regresa al paso (2) del flujo normal de eventos.</p> <p>4.1- Si los datos no son válidos se envía un mensaje informativo de que hay error en los datos introducidos. Regresa al paso (4) del flujo normal de eventos. 4.2 Regresa al Paso (2) del flujo normal de eventos.</p>
Sección “Modificar Sanción”	
<p>2- El Activista de Proceso Político escoge el comité de base del cual desea modificar la sanción de un militante.</p>	<p>1- El sistema muestra la interfaz de modificar sanción, mostrando un formulario con los comités de base que están registrados en la base de datos.</p> <p>2.1- El sistema verifica que haya seleccionado un comité de base. 2.2- El sistema muestra un listado con los militantes sancionados del comité de base seleccionado.</p>



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

<p>3- El Activista de Proceso Político selecciona al militante que le desea modificar la sanción.</p>	<p>3.1- El sistema pide la confirmación de que si es el estudiante deseado el que desea modificar su sanción. 3.2- El sistema muestra un listado con las sanciones del militante seleccionado.</p>
<p>4- El Activista de Proceso Político selecciona la sanción del militante seleccionado.</p>	<p>4.1- El sistema pide la confirmación de que si es la sanción seleccionada la que desea modificar. 4.2- El sistema muestra un formulario con los campos llenos, de la sanción del militante para que sean modificados.</p>
<p>5- El Activista de Proceso Político realiza los cambios deseados.</p>	<p>5.1- El sistema verifica que los campos obligatorios estén llenos. 5.2- El sistema pide la confirmación de que si desea guardar los cambios realizados en la sanción. 5.3- El sistema guarda los datos modificados de la sanción en la base de datos, le informa al Activista de Proceso Político que se ha efectuado la modificación y finaliza el caso de uso.</p>
<p>Curso Alterno</p>	
	<p>2.1- El sistema envía un mensaje solicitando escoger un comité de base. Regresa al paso (2) del flujo normal de eventos. 3.1- Regresa al Paso (3) del flujo normal de eventos. 4.1 – Regresa al Paso (4) del flujo normal de eventos. 5.1- El sistema envía un mensaje solicitando llenar todos los campos. Regresa al paso (5) del flujo normal de eventos. 5.2- Regresa al Paso (5) del flujo normal de eventos.</p>
<p>Sección “Mostrar Sanción”</p>	
<p>2- El Activista de Proceso Político escoge el comité de base del cual desea mostrar la sanción de un militante.</p>	<p>1- El sistema muestra la interfaz de mostrar sanción, mostrando un formulario con los comités de base que están registrados en la base de datos. 2.1- El sistema verifica que haya seleccionado un comité de base. 2.2- El sistema muestra un listado con los militantes sancionados del comité de base seleccionado.</p>
<p>3- El Activista de Proceso Político selecciona al militante que desea ver sus sanciones.</p>	<p>3.1- El sistema muestra un listado con las sanciones del militante seleccionado.</p>



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

4- El Activista de Proceso Político selecciona la sanción del militante seleccionado.	4.1- El sistema muestra una plantilla con los datos de la sanción del militante.
Curso Alterno	
	2.1- El sistema envía un mensaje solicitando escoger un comité de base. Regresa al paso (2) del flujo normal de eventos.
Sección “Eliminar Sanción”	
2- El Activista de Proceso Político escoge el comité de base del cual desea eliminarle la sanción a un militante.	1- El sistema muestra la interfaz de eliminar sanción, mostrando un formulario con los comités de base que están registrados en la base de datos. 2.1- El sistema verifica que haya seleccionado un comité de base. 2.2- El sistema muestra un listado con los militantes sancionados del comité de base seleccionado.
3- El Activista de Proceso Político selecciona al militante que desea quitarle la sanción.	3.1- El sistema muestra un listado con las sanciones del militante seleccionado.
4- El Activista de Proceso Político selecciona la sanción que desea quitarle al militante seleccionado.	4.1- El sistema pide la confirmación de si eliminar o no esta sanción. 4.2- El sistema elimina la sanción seleccionada, le informa al Activista de Proceso Político que se ha efectuado la eliminación y finaliza el caso de uso.
Curso Alterno	
	2.1 El sistema envía un mensaje solicitando escoger un comité de base. Regresa al paso (2) del flujo normal de eventos. 4.1 Regresa al paso (4) del flujo normal de eventos.
Prioridad:	Secundario

Tabla 12: Descripción del caso de uso del sistema Gestionar Sanción.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Nombre del caso de uso:	Emitir Reporte
Actor(es):	Act. Funcionamiento (inicia)
Propósito:	Mostrar los distintos reportes que desee el Activista de Funcionamiento.
Resumen:	El caso de uso se inicia cuando el Activista de Proceso Político accede a la interfaz de Emitir Reportes y selecciona la acción que desea realizar, ya sea Militantes por Comité de base, Actas registradas, Ausentes del mes y el porcentaje de asistencia de los Comité de base en el mes, y así finaliza el caso de uso con la ejecución de dicha acción.
Referencias:	R7.1 R7.2 R7.3 R7.4
Precondiciones:	Deben existir actas archivadas en el sistema.
Poscondiciones:	Reportes brindados por el sistema.
Curso Normal de Eventos:	
Acciones del Actor	Respuesta del Sistema
1-El Activista de Funcionamiento accede a la interfaz a la cual él tiene acceso. 3- El Activista de Funcionamiento escoge el tipo de reporte que desea consultar.	1.1-- El sistema muestra una serie de acciones a realizar. 2.1- Si elige a) Militantes por Comité de base ir a la sección "Militantes por Comité de base" b) Actas registradas ir a la sección "Actas registradas" c) Mostrar los ausentes del mes ir a la sección "Ausentes del mes" d) Ver el porcentaje de asistencia de los meses por Comité de base ir a la sección "Ver porcentaje de asistencia".
Sección "Militantes por Comité de base"	
2- El Activista de Funcionamiento escoge un Comité de Base para ver	1-El sistema muestra la interfaz de Militantes por Comité de Base, mostrando un formulario con los comités de base que están registrados en la base de datos. 2.1- El sistema verifica que haya seleccionado un Comité de Base. 2.2- El sistema muestra un listado de los militantes del Comité de



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

sus militantes.	Base seleccionado y así finaliza el caso de uso.
Curso Alterno	
	2.1 El sistema envía un mensaje solicitando escoger un comité de base. Regresa al paso (2) del flujo normal de eventos.
Sección “Actas Registradas”	
2- El Activista de Funcionamiento escoge un mes para ver sus actas.	1- El sistema muestra la interfaz de Actas Registradas, mostrando un formulario con los meses del curso. 2.1- El sistema verifica que se haya escogido un mes. 2.2- El sistema muestra un listado de los comités de base y la fecha de realización del acta del mes seleccionado.
3- El Activista de Funcionamiento escoge el acta del Comité de Base que desee.	3.1- El sistema muestra un formulario con los datos del acta seleccionada y así finaliza el caso de uso.
Curso Alterno	
	2.1 El sistema muestra un mensaje solicitando un mes. Regresa al paso (2) del flujo normal de eventos.
Sección “Ausentes del mes”	
	1- El sistema muestra un listado con los ausentes del mes el cual posee nombre de comité de base al que pertenece, nombre y apellidos del militante y causa de la ausencia, así finaliza el caso de uso.
Sección “Porciento de asistencia”	
2- El Activista de Funcionamiento escoge un mes para ver los porcentos de asistencia.	1- El sistema muestra la interfaz de Porciento de asistencia, mostrando un formulario con los meses del curso. 2.1- El sistema verifica que se haya escogido un mes. 2.2- El sistema muestra un listado con los comités de base y su porciento de asistencia ordenado descendentemente y así finaliza el caso de uso.
Curso Alterno	
	2.1 El sistema muestra un mensaje solicitando un mes. Regresa al paso (2) del flujo normal de eventos.
Prioridad:	Secundario

Tabla 13: Descripción del caso de uso del sistema Emitir Reporte.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Nombre del caso de uso:	Generar Cierre de Funcionamiento
Actor(es):	Act. Funcionamiento (inicia)
Propósito:	Mostrar el cierre de funcionamiento del mes actual.
Resumen:	El caso de uso se inicia cuando el Activista de Funcionamiento escoge la opción Generar Cierre. Finaliza cuando el sistema muestra el Cierre de Funcionamiento.
Referencias:	R8.1
Precondiciones:	Deben existir actas archivadas en el sistema del mes actual.
Poscondiciones:	Cierre de Funcionamiento brindado por el sistema.
Curso Normal de Eventos:	
Acciones del Actor	Respuesta del Sistema
1-El Activista de Funcionamiento accede a la interfaz a la cual él tiene acceso	1.1-- El sistema muestra una serie de acciones a realizar.
2- El Activista de Funcionamiento escoge la opción Generar Cierre de Funcionamiento.	2.1- El sistema muestra el cierre del mes actual y así finaliza el caso de uso.
Prioridad:	Secundario

Tabla 14: Descripción del caso de uso del sistema Gestionar Cierre de Funcionamiento.

Nota 1: El resto de las descripciones textuales, se encuentran en el **Anexo I del Trabajo.**



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

CONCLUSIONES

En este capítulo se pueden apreciar las características fundamentales del negocio, se definieron, describieron y comprendieron los principales procesos de negocio, centrándose fundamentalmente en “qué” hace el sistema y no en “cómo” lo hace. Se definieron las principales funcionalidades que debe tener el sistema a desarrollar, mostradas en los requisitos funcionales, estructurándose en casos de uso y describiéndose textualmente cada uno de ellos. Todo esto antes mencionado proporciona una visión general de lo que el sistema debe hacer.



CAPÍTULO 3: DISEÑO DEL SISTEMA

CAPÍTULO 3: DISEÑO DEL SISTEMA

INTRODUCCIÓN

En el desarrollo de este capítulo se muestran los resultados obtenidos durante el flujo de trabajo de Diseño del sistema, a través de los diagramas de clases del diseño, además de los diagramas de interacción y del diseño de la base de datos. En el flujo de trabajo de diseño se realiza un refinamiento del análisis, teniendo en cuenta los requisitos no funcionales, que no es más que ver cómo cumple el sistema sus objetivos y considerando además el entorno de implementación (Lenguaje de programación, Framework, plataforma, etc.). Tiene como propósitos fundamentales: Transformar los requerimientos en un diseño de cómo debe ser el sistema y adaptar el diseño para que se corresponda con el entorno de implementación, diseñando sus funcionalidades.

3.1. DISEÑO

3.1.1 DIAGRAMAS DE INTERACCIÓN.

Los diagramas de Interacción muestran una interacción concreta: un conjunto de objetos y sus relaciones, junto con los mensajes que se envían entre ellos, modelan el comportamiento dinámico del sistema; el flujo de control en una operación; describen la interacción entre objetos; los objetos interactúan a través de mensajes para cumplir ciertas tareas; las interacciones provee un “comportamiento” y típicamente implementan un Caso de Uso y existen dos tipo de diagramas de interacción en UML: Diagramas de Secuencia (dimensión temporal), Diagramas de Colaboración (dimensión estructural). Sirven para mostrar la relación entre los distintos objetos que participan en un escenario y esta relación se establece mediante el paso de mensajes.



CAPÍTULO 3: DISEÑO DEL SISTEMA

Los diagramas de secuencia son utilizados para modelar los aspectos dinámicos de un sistema y para construir sistemas ejecutables por medio de ingeniería directa e inversa. Un diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes y muestra interacciones basados en tiempo entre los objetos. Gráficamente, es una tabla que representa objetos, dispuestos a lo largo del eje X, y mensajes, ordenados según se suceden en el tiempo, a lo largo del eje Y. [2]

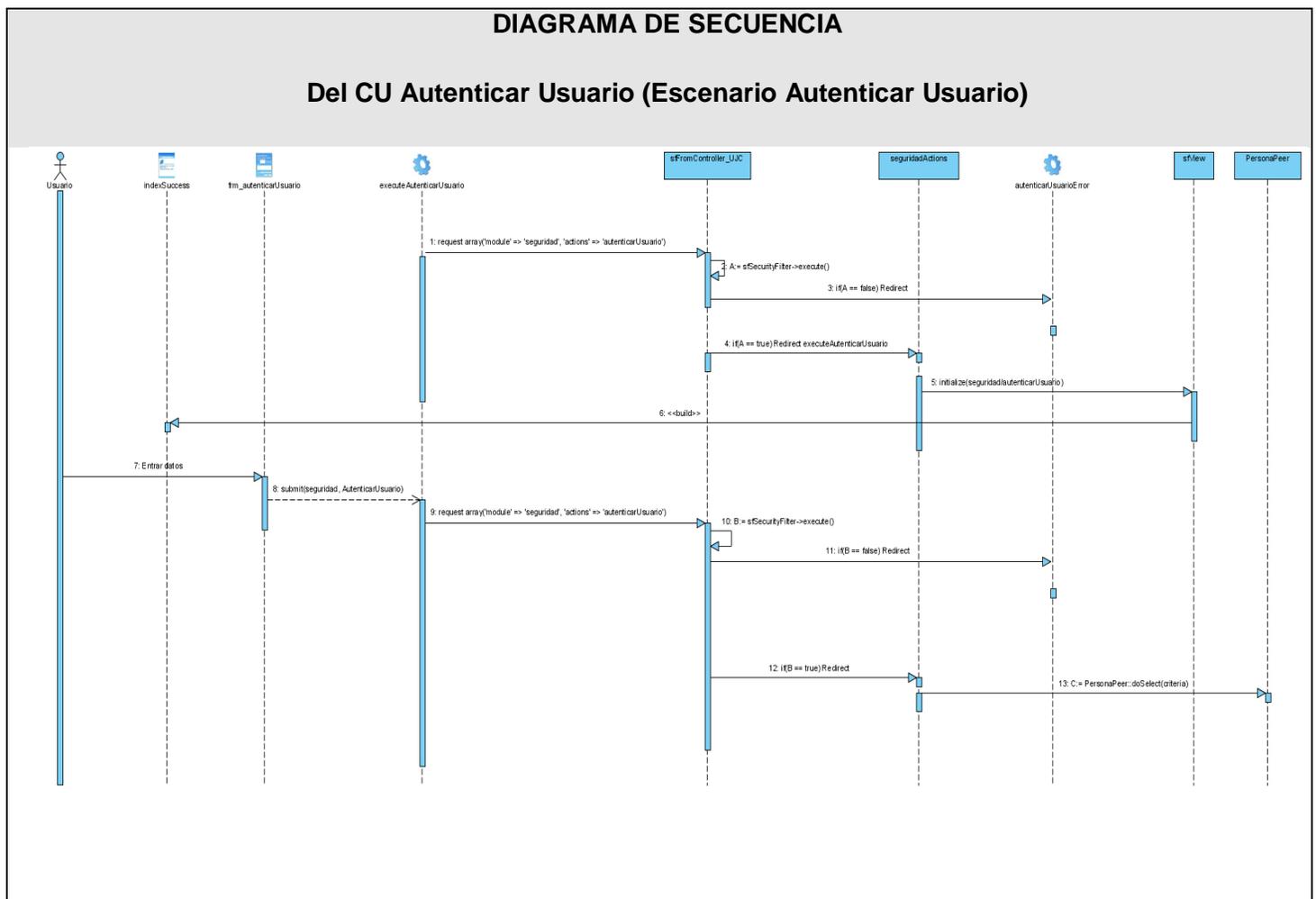


Figura 14: DS del Caso de Uso del Sistema Gestionar Usuario (Escenario Autenticar Usuario).



CAPÍTULO 3: DISEÑO DEL SISTEMA

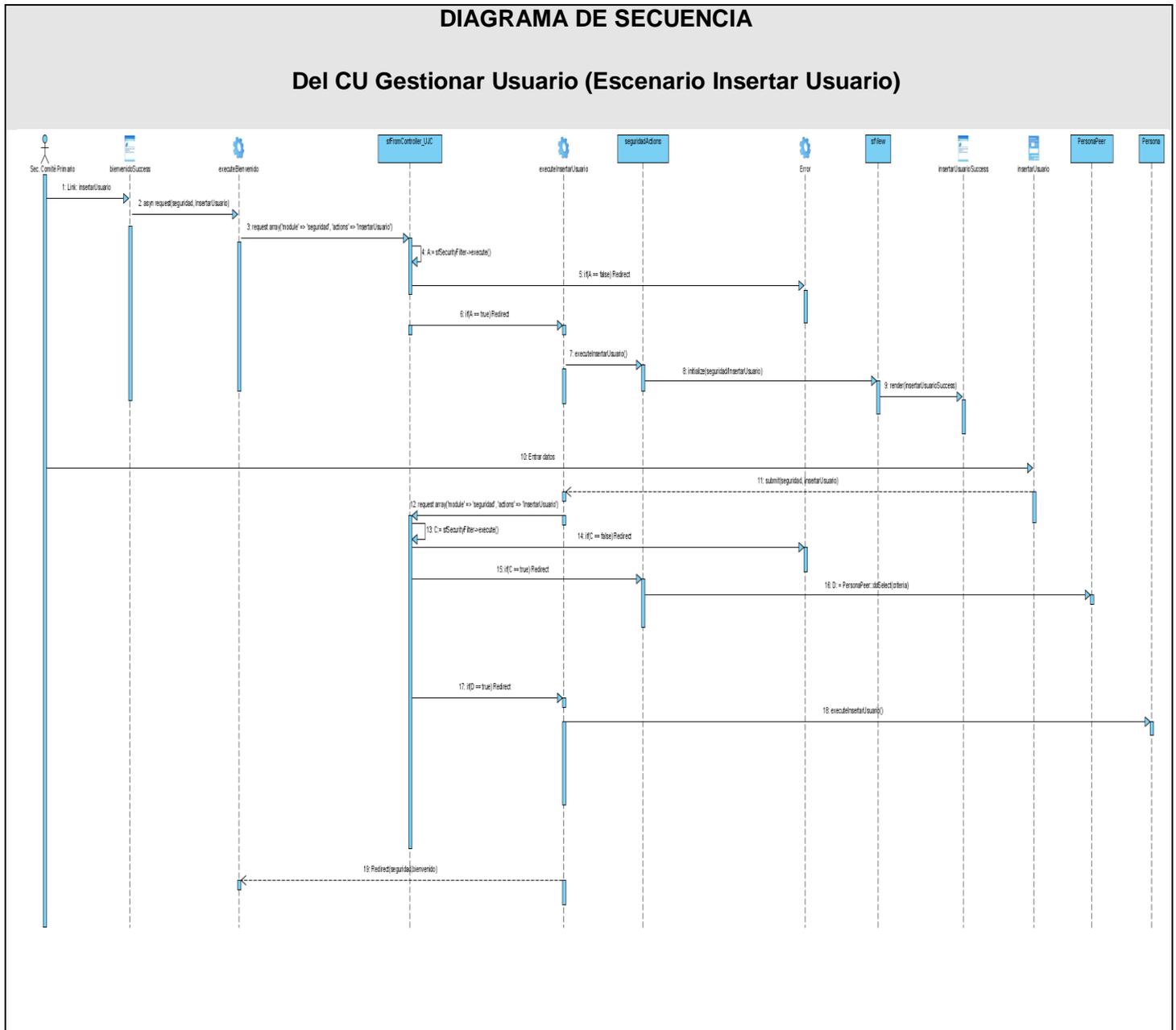


Figura 15: DS del Caso de Uso del Sistema Gestionar Usuario (Escenario Insertar Usuario).



CAPÍTULO 3: DISEÑO DEL SISTEMA

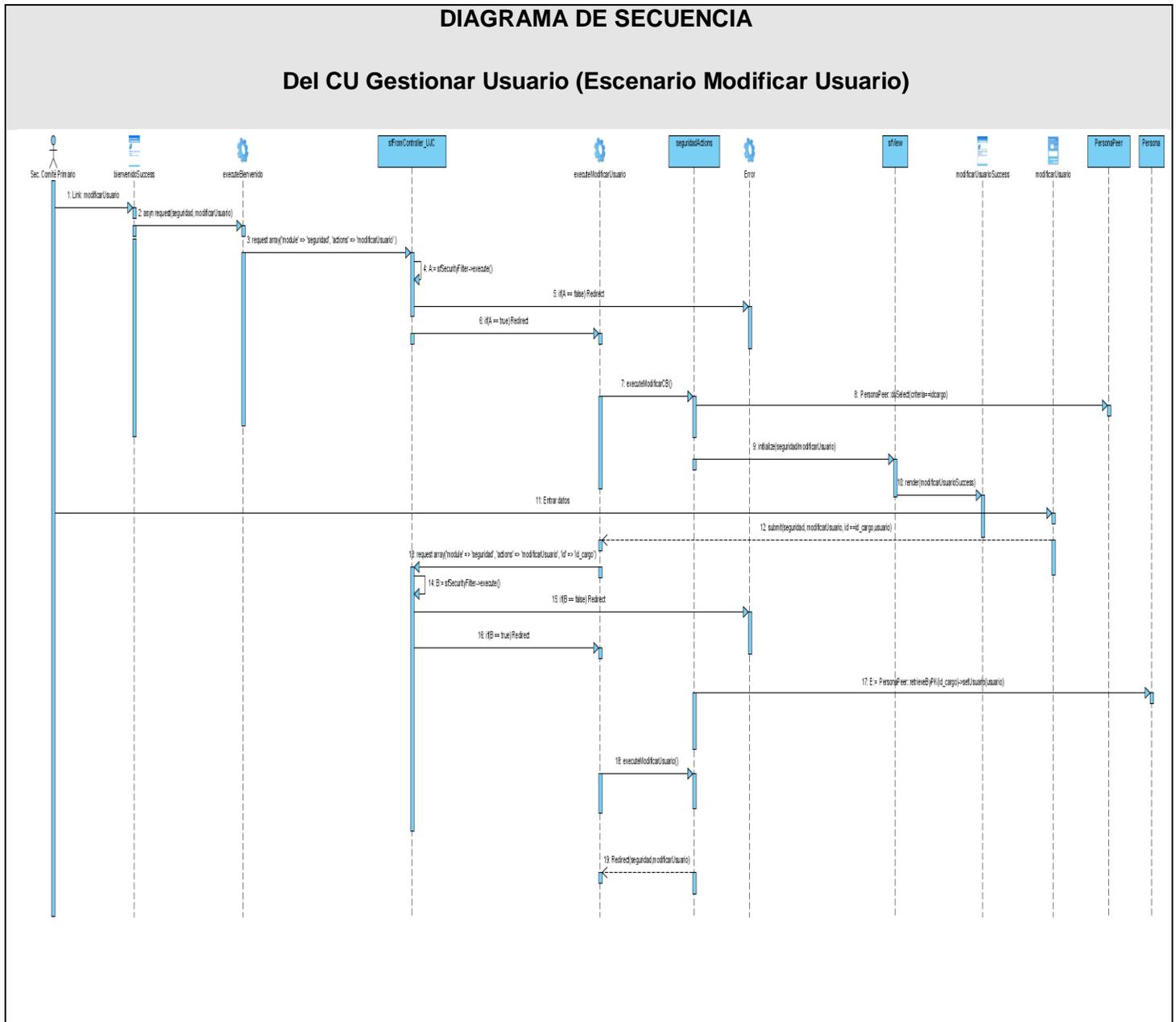


Figura 16: DS del Caso de Uso del Sistema Gestionar Usuario (Escenario Modificar Usuario).



CAPÍTULO 3: DISEÑO DEL SISTEMA

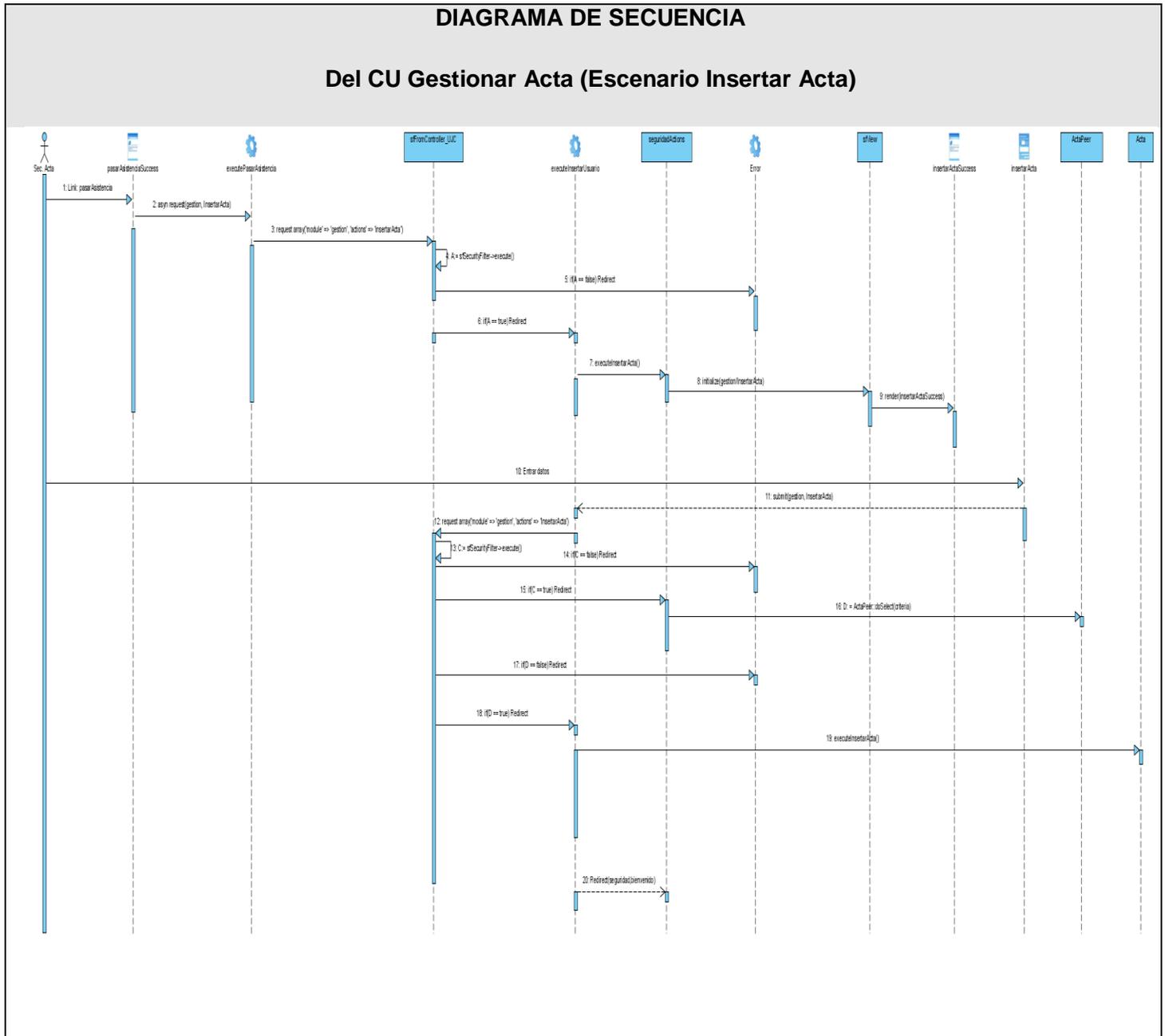


Figura 17: DS del Caso de Uso del Sistema Gestionar Acta (Escenario Insertar Acta).



CAPÍTULO 3: DISEÑO DEL SISTEMA

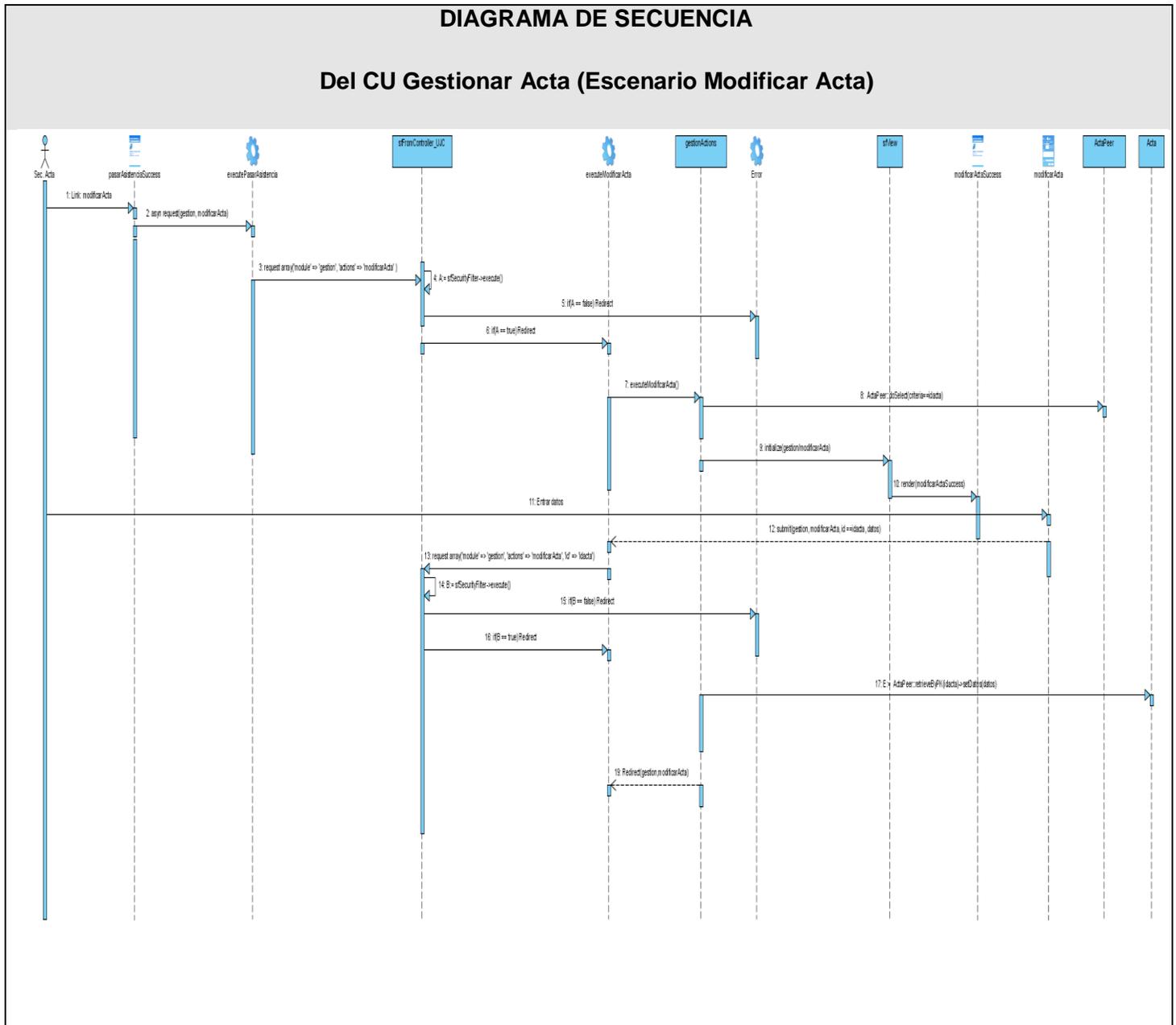


Figura 18: DS del Caso de Uso del Sistema Gestionar Acta (Escenario Modificar Acta).



CAPÍTULO 3: DISEÑO DEL SISTEMA

3.1.2 DIAGRAMA DE CLASES DEL DISEÑO.

Los diagramas de clases del diseño muestran un conjunto de clases, interfaces y colaboraciones con sus relaciones estructurales, de herencia o de composición; en el caso de los que son mostrados mediante estereotipos web, es decir que van hacer para mostrar el diseño de aplicaciones Web, representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase. En este tipo de aplicaciones son más importantes la modelación de la lógica y estado del negocio que los detalles de presentación.

Diseño con estereotipo Web Arquitectura Básica

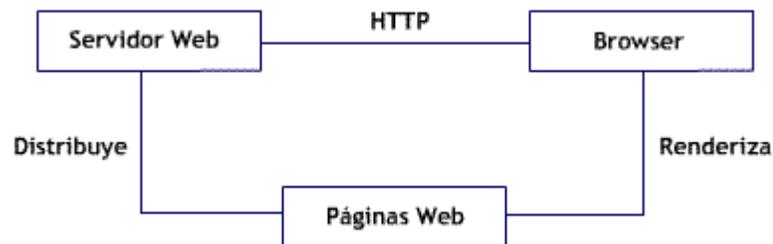


Figura 19: Arquitectura Básica.

Componentes básicos

- Navegador
- Servidor Web
- Servidor de aplicaciones
- Servidor de B.D



CAPÍTULO 3: DISEÑO DEL SISTEMA

Estereotipos de Clase:



<<Server Page>>

Representa la página Web con scripts que se ejecuta en el servidor. Los scripts interactúan con recursos del servidor, ya sean bases de datos, lógica del negocio, sistemas externos, etc. Las operaciones representan las funciones en el script y los atributos representan las variables visibles en el ámbito de la página, estas son accesibles por todas las funciones en la página. Y tiene como restricción que solo puede participar en relaciones con objetos del servidor.



<<Client Page>>

Una instancia de Página Cliente es una página Web con formato HTML, es decir, mezcla de datos, presentación y lógica. Es mostrada por un navegador o browser. Las operaciones representan funciones en el script en la página, los atributos representan variables del script, que son accesibles por toda función en la página. Puede participar en asociaciones con Páginas Cliente o con Páginas Servidoras. Y cada página cliente solo puede ser construida por una página servidor.



<<Form>>

Colección de campos de entrada de datos, es parte de una Página Cliente. No tiene operaciones. Sus atributos representan campos de entrada del form HTML (Text Field, Text Area, Button, Label, Radio Button, Radio Group, Select, Check Box y Hidden Fields). Toda operación que interactúa con el form es una propiedad de la página que contiene el form. (17)



CAPÍTULO 3: DISEÑO DEL SISTEMA

3.2. DISEÑO DE LA BASE DE DATOS

3.2.1 MODELO LÓGICO DE DATOS (DIAGRAMA DE CLASES PERSISTENTES).

Diagrama que muestra un conjunto de objetos capaces de almacenar su estado en un medio permanente.

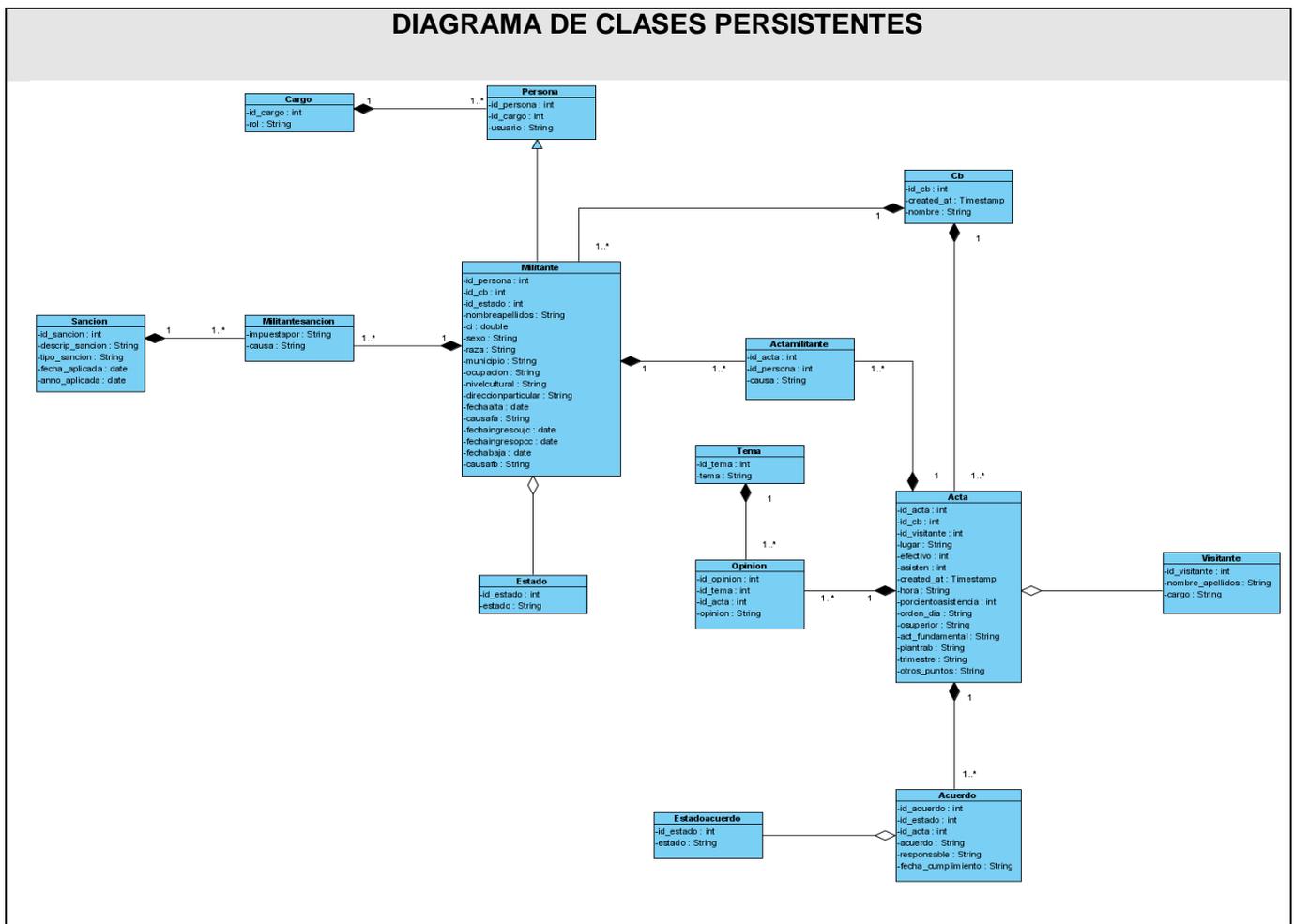


Figura 26: Diagrama de Clases Persistentes del Sistema.



CAPÍTULO 3: DISEÑO DEL SISTEMA

3.2.2 MODELO FÍSICO DE DATOS (MODELO DE DATOS).

El Modelo Entidad Relación (MER) se basa en la percepción del mundo real, los elementos esenciales del modelo son las entidades, los atributos y las relaciones entre las entidades. Su propósito es simplificar el diseño de bases de datos a partir de descripciones textuales de los requerimientos.

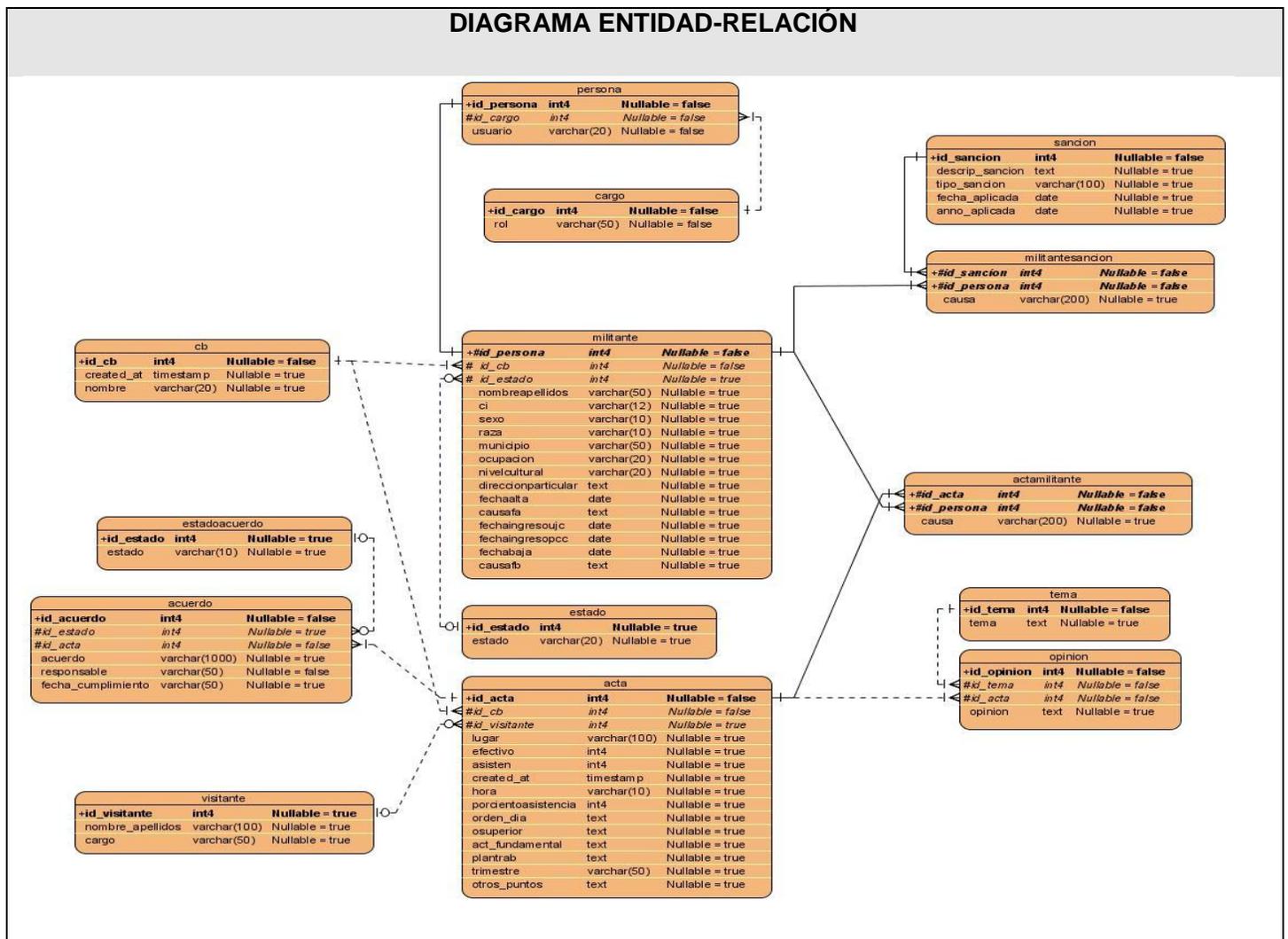


Figura 27: Modelo Entidad-Relación del Sistema.



CAPÍTULO 3: DISEÑO DEL SISTEMA

CONCLUSIONES

En este capítulo se pueden apreciar los diagramas de clases del Diseño que se hicieron usando estereotipos web de cada uno de los casos de uso, los diagramas de Interacción de cada uno de los escenarios que se pueden apreciar dentro de cada uno de los casos de uso, y el diseño de la Base de Datos.



CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

INTRODUCCIÓN

En el desarrollo de este capítulo se dará a conocer el flujo de trabajo de implementación, el cual se comienza con el resultado del diseño y describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue. Se muestran las dependencias entre las partes de código y la estructura del sistema en ejecución mediante los diagramas de componentes y despliegue respectivamente. Los diagramas de despliegue y componentes, que son artefactos generados en este flujo de trabajo conforman lo que se conoce como un modelo de implementación al describir los componentes a construir y su organización y dependencia entre nodos físicos en los que funcionará la aplicación. Se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. [2]

4.1. DIAGRAMA DE DESPLIEGUE

El diagrama de despliegue se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Es un grafo de nodos unidos por conexiones de comunicación que muestra las relaciones físicas entre los componentes hardware y software en el sistema final. Se compone por nodos, dispositivos y conectores; donde los nodos son elementos de procesamiento con al menos un procesador, memoria, etc.- Ej: Servidor, PC cliente-; los dispositivos son nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela –Ej: Impresora- y los conectores expresan el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo- Ej: USB, HTTP-. Con él se captura la configuración de los elementos de procesamiento y sus conexiones y se visualiza la distribución de los componentes de software en los nodos físicos. [2]



CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

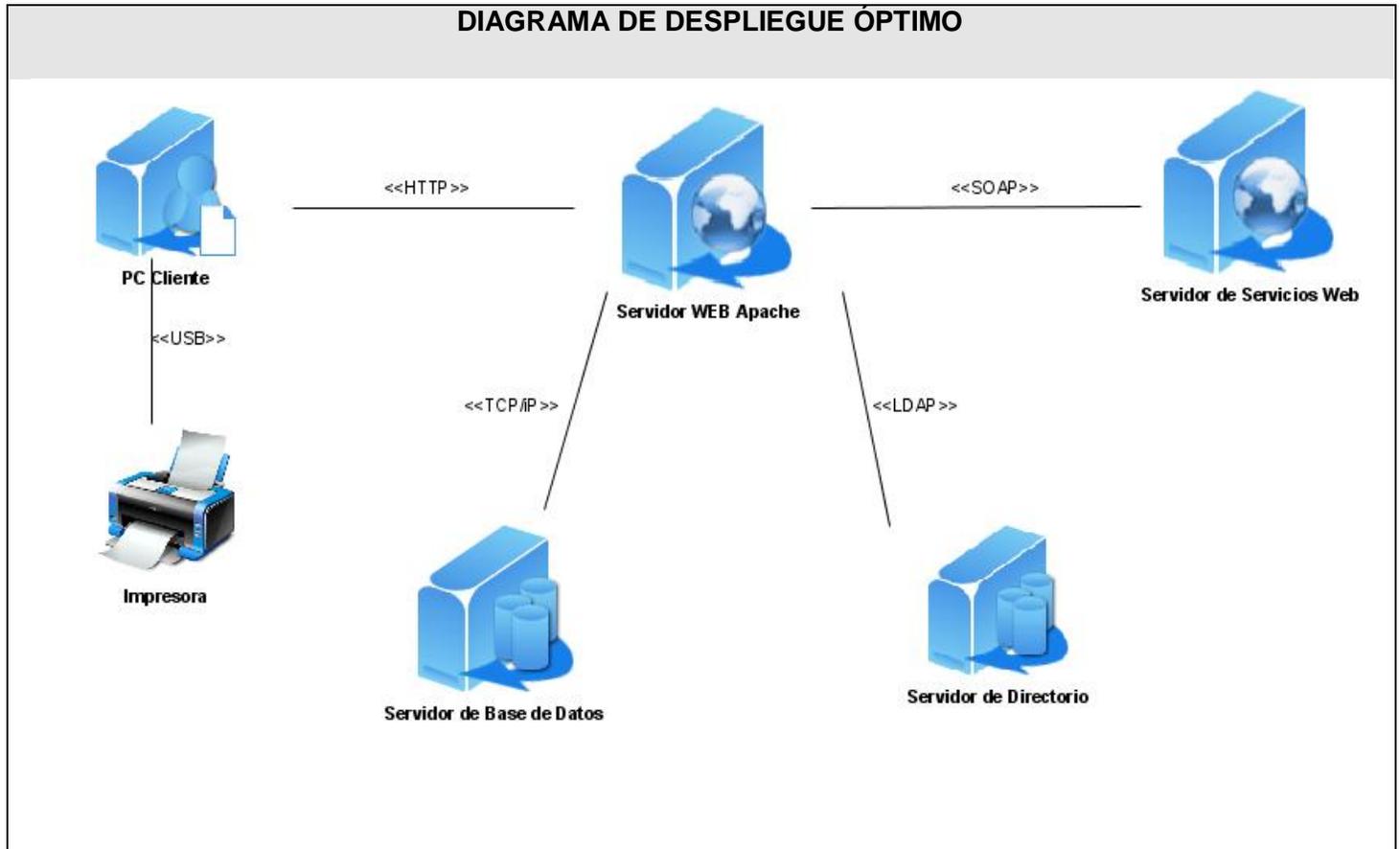


Figura 28: Diagrama de Despliegue Óptimo del F2-UJC.



CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

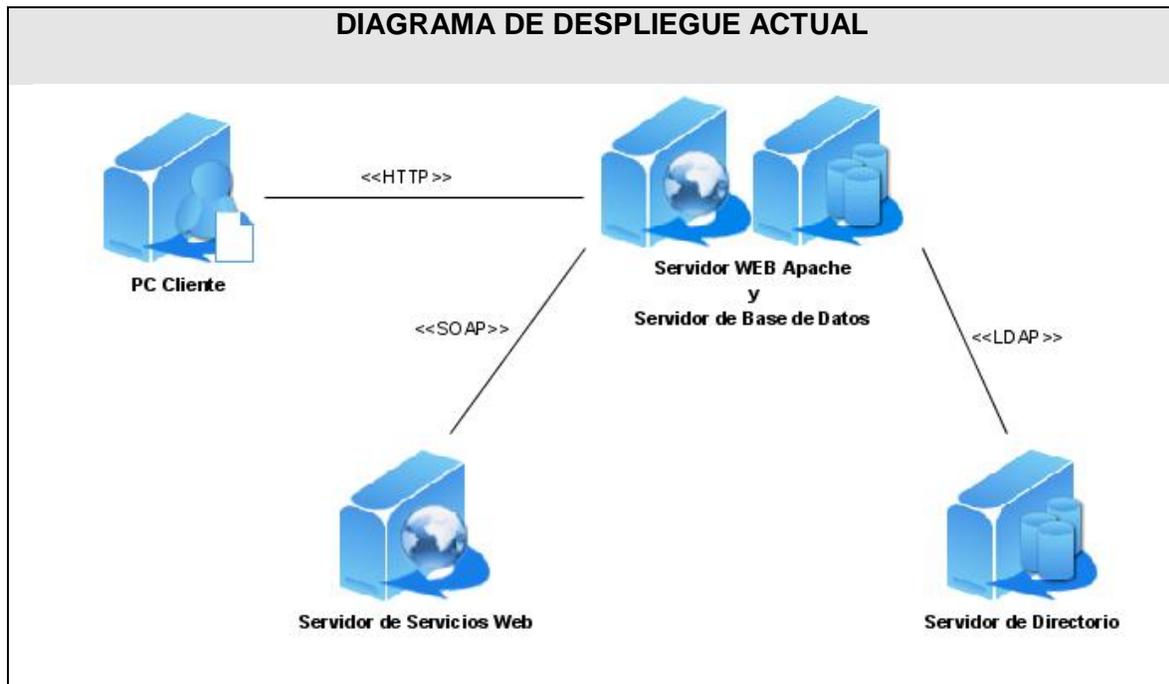


Figura 29 : Diagrama de Despliegue Actual del F2-UJC.

4.2. DIAGRAMA DE COMPONENTES

Un diagrama de componentes muestra las dependencias de compilación de los ficheros de código, relaciones de derivación entre ficheros de código fuente y ficheros que son resultados de la compilación, dependencias entre elementos de implementación y los correspondientes elementos de diseños que son implementados. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Se utilizan para modelar la vista estática de un sistema y estructurar el modelo de implementación en



CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. [2]

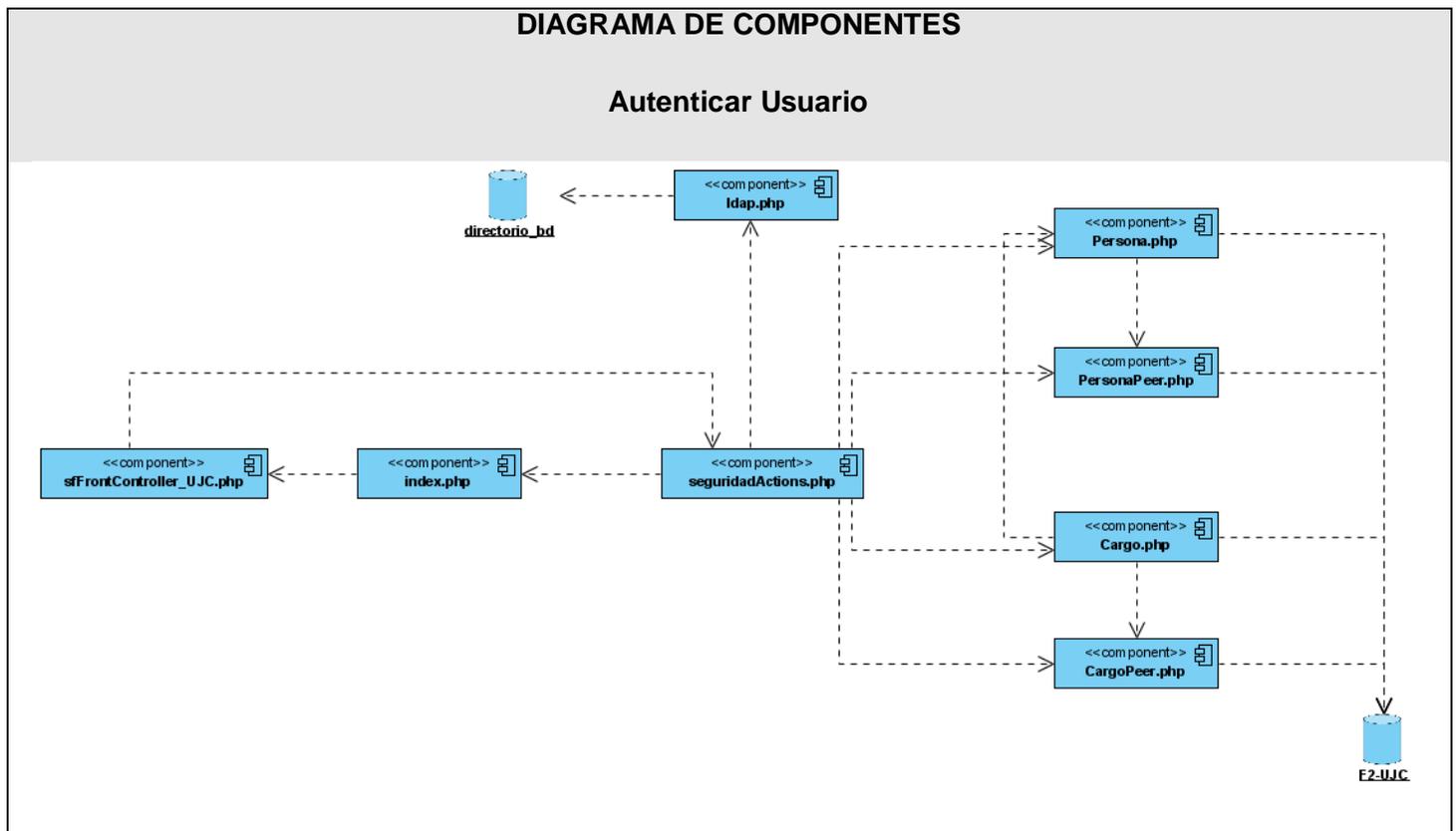


Figura 30: Diagrama de componentes del Caso de Uso del Sistema Autenticar Usuario.



CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

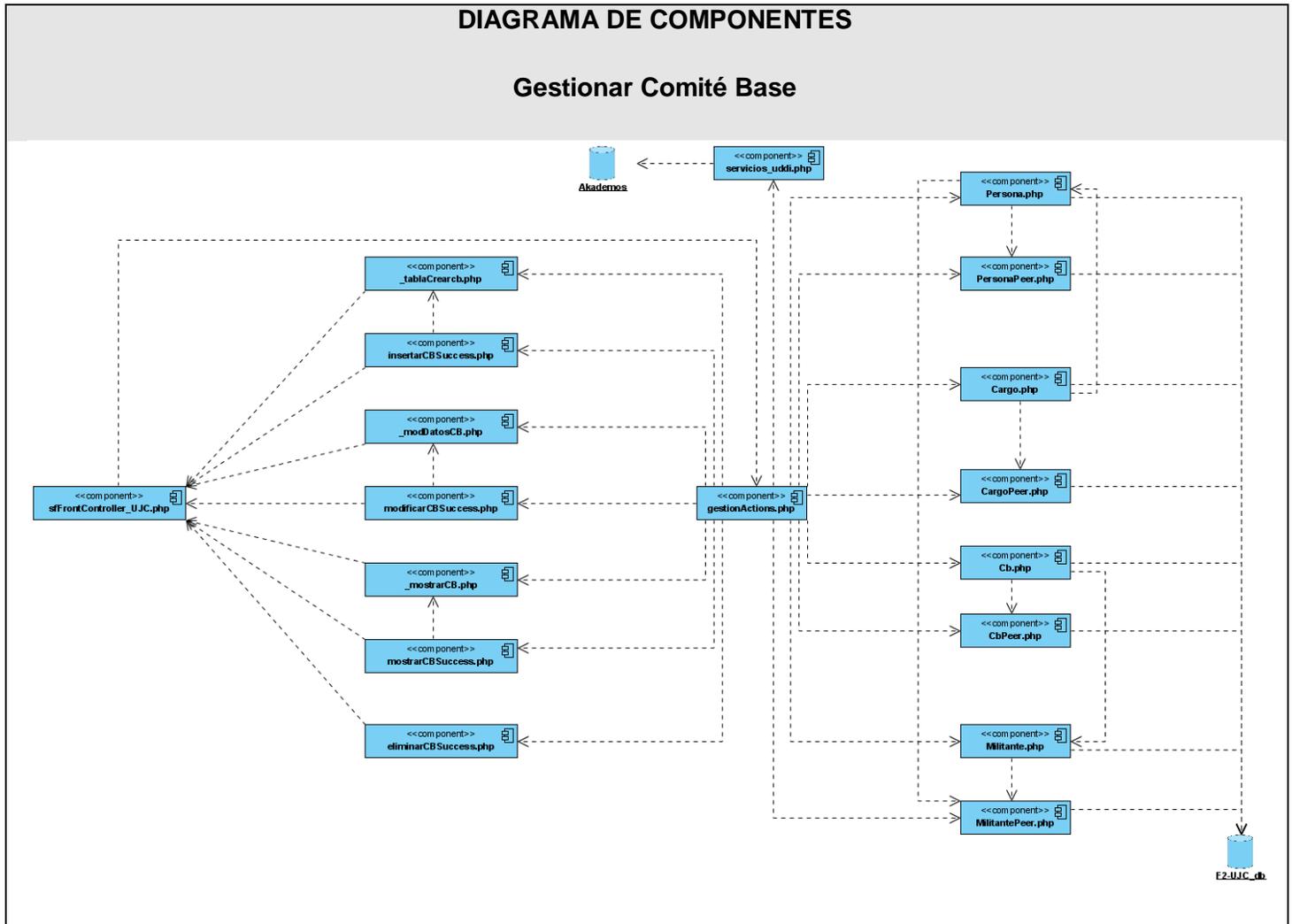


Figura 31: Diagrama de componentes del Caso de Uso del Sistema Gestionar Comité Base.



CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

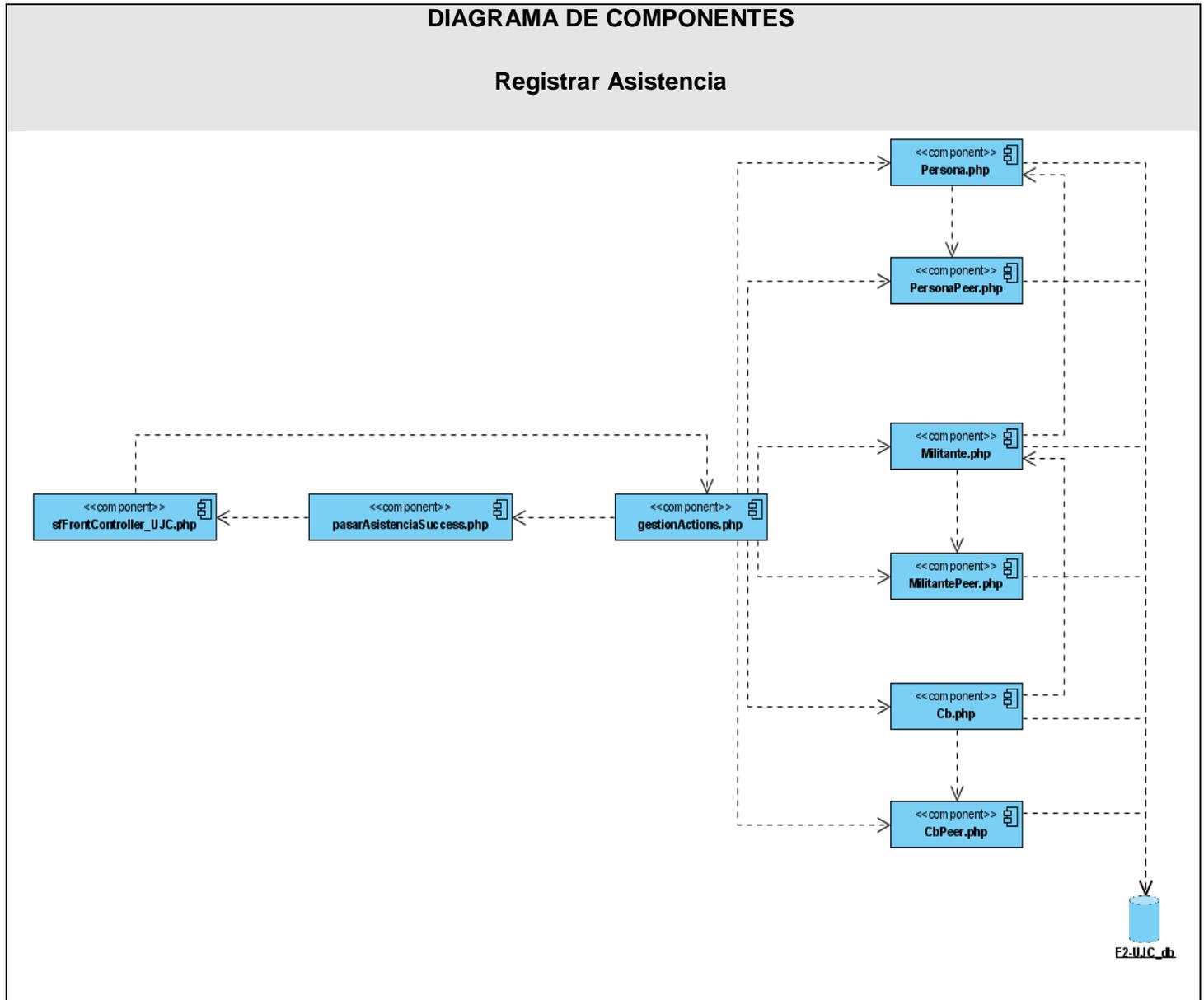


Figura 32: Diagrama de componentes del Caso de Uso del Sistema Registrar Asistencia.



CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

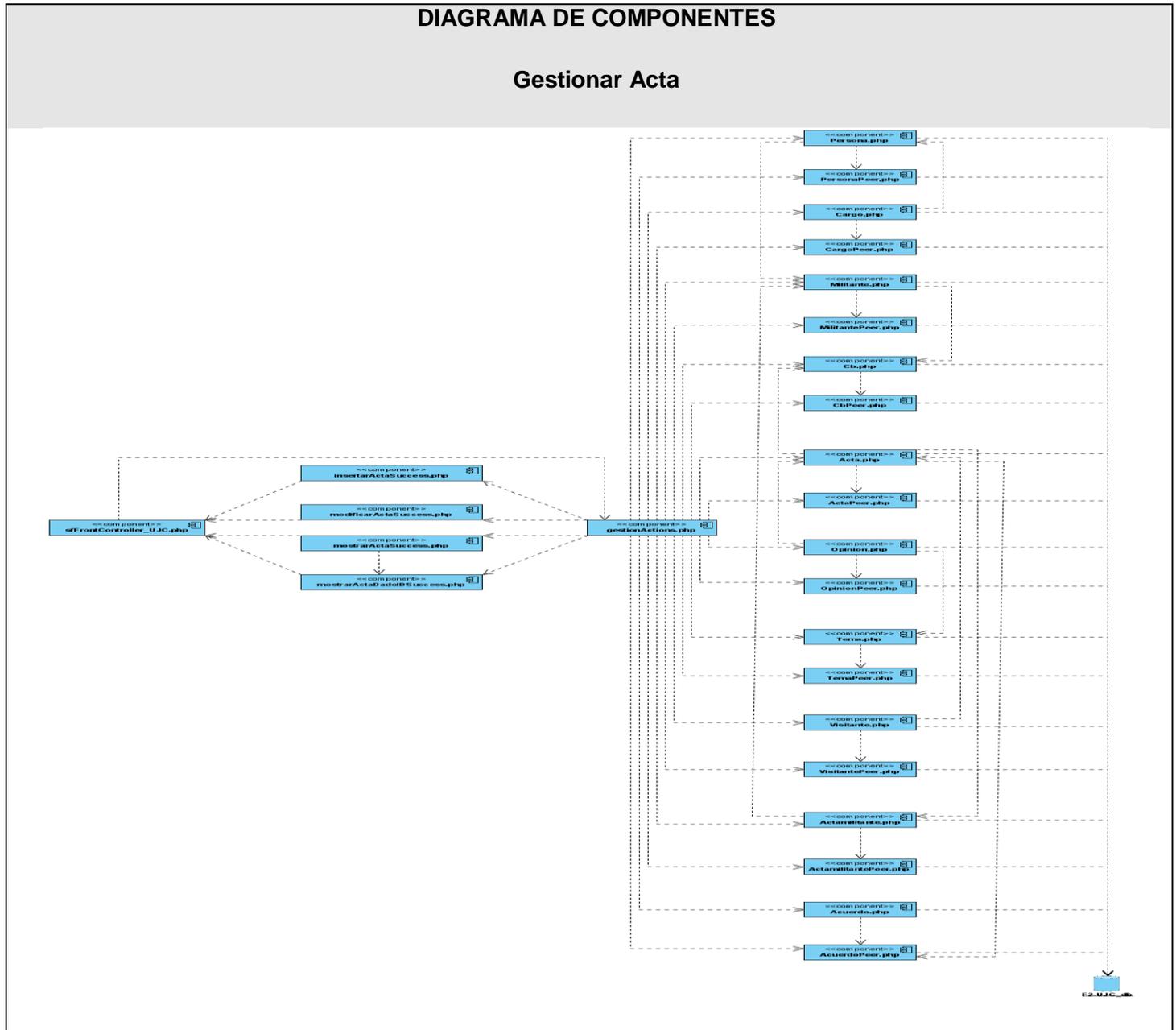


Figura 33: Diagrama de componentes del Caso de Uso del Sistema Gestionar Acta.



CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

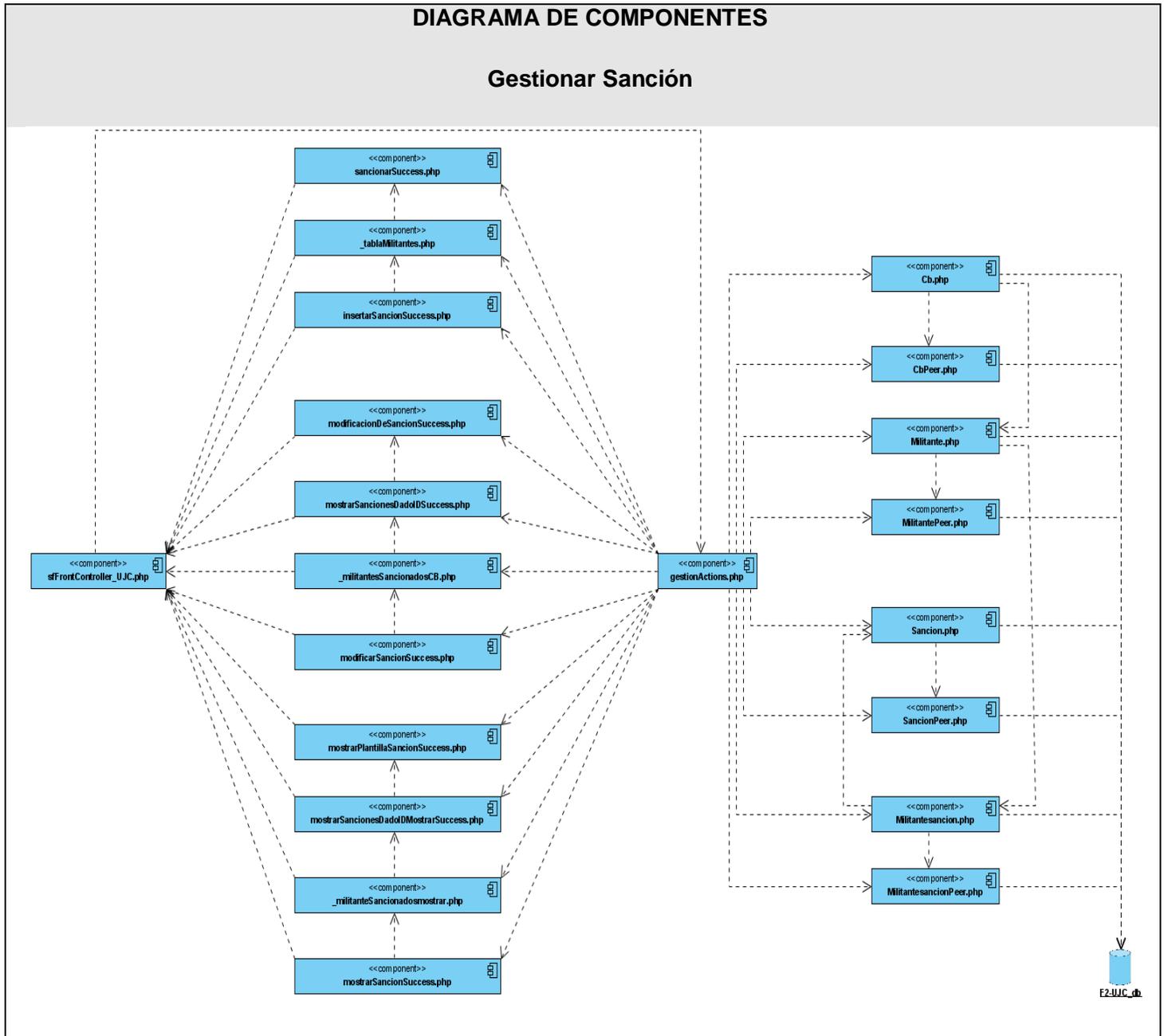


Figura 34: Diagrama de componentes del Caso de Uso del Sistema Gestionar Sanción.



CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

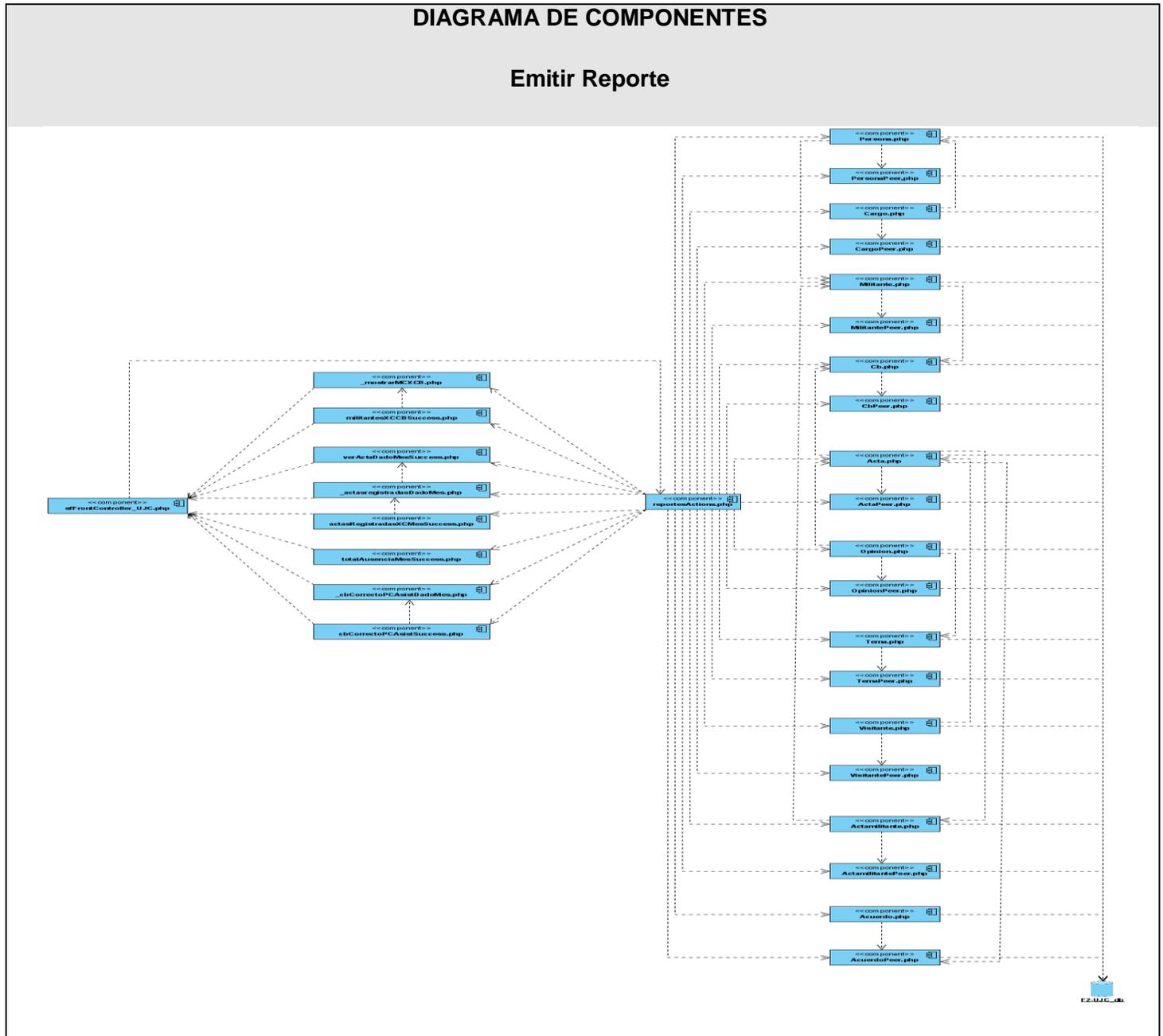


Figura 35: Diagrama de componentes del Caso de Uso del Sistema Emitir Reporte.



CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

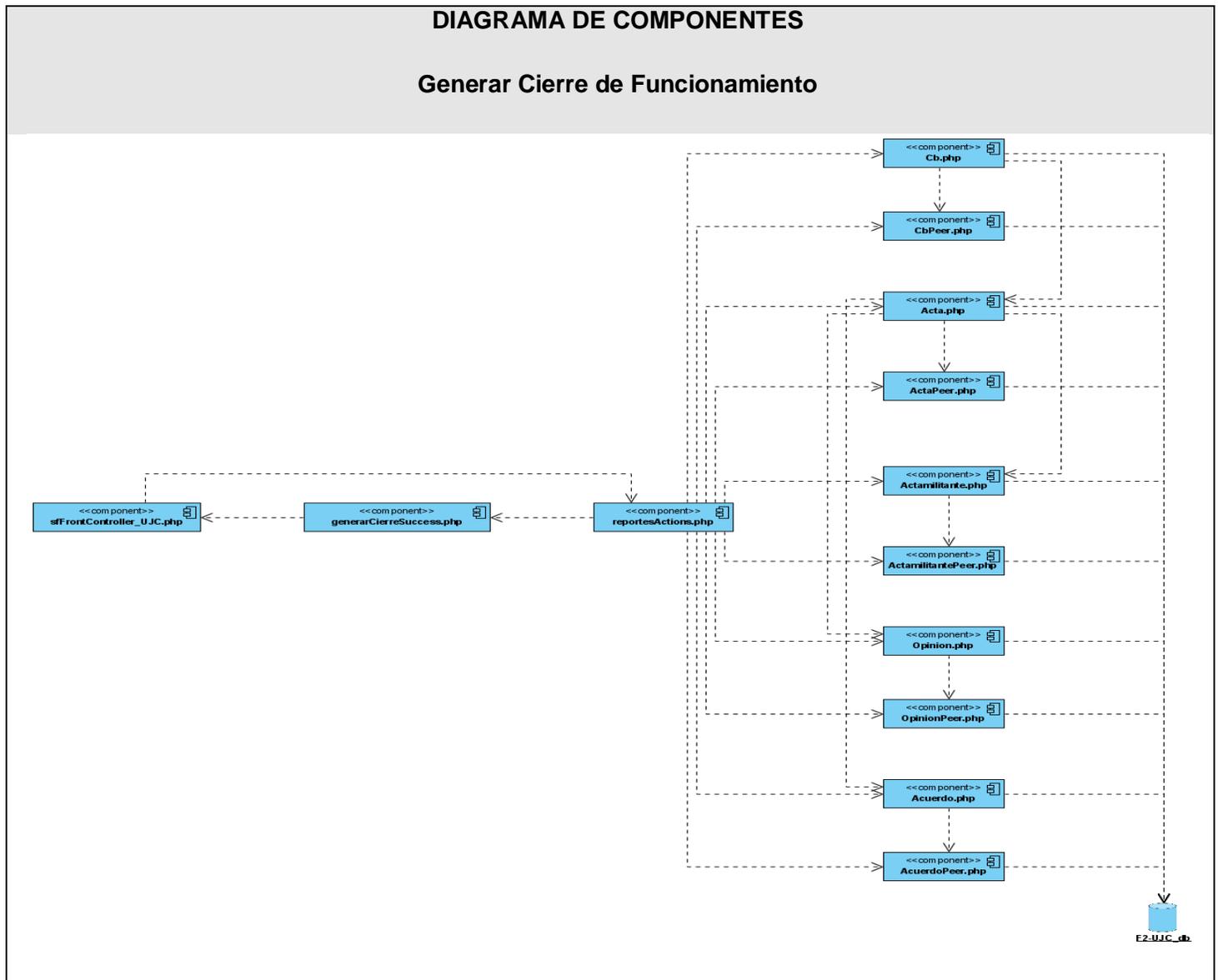


Figura 36: Diagrama de componentes del Caso de Uso del Sistema Generar Cierre de Funcionamiento.

Nota 1: El resto de los diagramas de componentes, se encuentran en el **Anexo III del Trabajo**.



CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4.3. PRUEBA

4.3.1 PRUEBAS DE CAJA NEGRA:

Tabla 15: Pruebas de Caja Negra para el caso de uso “Autenticar Usuario”

Condición de entrada	Casos válidos	Casos no válidos
Usuario	Cadena de caracteres	Dejar vacío el campo usuario
Contraseña	Cadena de caracteres	Dejar vacío el campo contraseña.

Caso de uso	Autenticar usuario
Caso de prueba	Permitir autenticarse al usuario que introduzca bien sus datos
Entrada:	El usuario introduce correctamente sus datos ejemplo: Usuario: “wlopez” contraseña: “wlopez1”
Resultado:	El sistema da acceso al usuario autenticado.
Condiciones:	El usuario y contraseña deben estar correctos.

Caso de uso	Autenticar usuario
Caso de prueba	Permitir autenticarse al usuario que introduzca bien sus datos
Entrada:	El usuario introduce incorrectamente sus datos ejemplo: Usuario: “wlopez” contraseña: “campo vacío”
Resultado:	El sistema muestra un mensaje solicitando llenar el campo de contraseña.
Condiciones:	El usuario o la contraseña deben estar incorrectos.



CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

Tabla 16: Pruebas de Caja Negra para el caso de uso “Gestionar sanción”: Sección: insertar sanción

Condición de entrada	Casos válidos	Casos no válidos
fecha_sancion	fecha del date asignado	Dejar vacío el campo fecha_sanción.
tipo_sancion	Cadena de caracteres	Dejar vacío el campo tipo_sanción.
anno_aplicada	Cadena de caracteres	Dejar vacío el campo anno_aplicada.
Sanción	Cadena de caracteres	Dejar vacío el campo sanción.
Causa	Cadena de caracteres	Dejar vacío el campo causa de la sanción.

Caso de uso	Insertar sanción.
Caso de prueba	Insertar una sanción introduciendo correctamente los datos.
Entrada:	El usuario introduce correctamente los datos para la sanción por ejemplo: Año: “2009”. Tipo Sanción: “LTD”. Causa:”Mala conducta ante las actividades patrióticas” Sanción: “El estudiante posee una limitación temporal de derechos por dos años”. Fecha: “28/05/2009”.
Resultado:	El sistema inserta la sanción en la base de datos.
Condiciones:	Los datos introducidos por el usuario deben estar correctos.

Caso de uso	Insertar sanción.
Caso de prueba	Insertar una sanción introduciendo incorrectamente los datos.
Entrada:	El usuario introduce incorrectamente los datos para la sanción por ejemplo: Año: “2009”. Tipo Sanción: “LTD”.



CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

	Causa: "Mala conducta ante las actividades patrióticas" Sanción: "El estudiante posee una limitación temporal de derechos por dos años". Fecha: " campo vacío ".
Resultado:	El sistema muestra el mensaje: "Debe ingresarle una fecha de sanción".
Condiciones:	Algún campo vacío.

Tabla 17: Pruebas de Caja Negra para el caso de uso "Gestionar sanción": Sección: modificar sanción

Condición de entrada	Casos válidos	Casos no válidos
fecha_sancion	fecha del date asignado	Dejar vacío el campo fecha_sancion.
tipo_sancion	Cadena de caracteres	Dejar vacío el campo tipo_sancion.
anno_aplicada	Cadena de caracteres	Dejar vacío el campo anno_aplicada.
Sanción	Cadena de caracteres	Dejar vacío el campo sanción.
Causa	Cadena de caracteres	Dejar vacío el campo causa de la sanción.

Caso de uso	Modificar sanción.
Caso de prueba	Modificar una sanción introduciendo correctamente los datos.
Entrada:	El usuario introduce correctamente los datos para la sanción por ejemplo: Año: "2009". Tipo Sanción: "LTD". Causa: "Mala conducta ante las actividades patrióticas" Sanción: "El estudiante posee una limitación temporal de derechos por dos años". Fecha: "28/05/2009".
Resultado:	El sistema modifica la sanción en la base de datos.
Condiciones:	Los datos introducidos por el usuario deben estar



CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

	correctos.
Caso de uso	Modificar sanción.
Caso de prueba	Modificar una sanción introduciendo incorrectamente los datos.
Entrada:	El usuario introduce incorrectamente los datos para la sanción por ejemplo: Año: "2009". Tipo Sanción: "LTD". Causa: "Mala conducta ante las actividades patrióticas" Sanción: "El estudiante posee una limitación temporal de derechos por dos años". Fecha: "campo vacío".
Resultado:	El sistema muestra el mensaje: "Debe ingresarle una fecha de sanción".
Condiciones:	Algún campo vacío.

CONCLUSIONES

En este capítulo se puede apreciar el Diagrama de Despliegue del sistema, así como los Diagramas de Componentes asociado a cada uno de los Diagramas de Clases del Diseño y algunas de las pruebas realizadas al sistema.



CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD

INTRODUCCIÓN

En el desarrollo de este capítulo se dará a conocer el Estudio de la Factibilidad, el cual tiene como objetivo fundamental determinar la posibilidad de llevar adelante el proyecto, con la estimación del mismo, la cual se llevará a cabo mediante el análisis de Puntos de Casos de Uso que es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan -estos factores son por ejemplo: el Factor de complejidad técnica(TCF) que dentro de estos factores están la Portabilidad, la Facilidad de uso, la Facilidad de instalación, entre otros, y el Factor de ambiente(EF) que dentro de estos factores se pueden encontrar la Motivación, la Capacidad del analista líder, la Estabilidad de requerimientos, entre otros-, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. Además en este capítulo se estima el esfuerzo, el costo, el tiempo que durará el desarrollo del sistema y se analizan los beneficios del sistema propuesto (F2-UJC).

5.1. APLICAR MÉTODO DE ESTIMACIÓN PUNTOS POR CASOS DE USO

5.1.1 CÁLCULO DE PUNTOS DE CASOS DE USO SIN AJUSTAR.

$$UUCP = UAW + UUCW$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar

UAW: Factor de Peso de los Actores sin ajustar

UUCW: Factor de Peso de los Casos de Uso sin ajustar

Para calcular el Factor de Peso de los Actores sin ajustar (UAW)



CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD

Tipo de Actor	Descripción	Peso	Actores	Cant actores * peso)
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface)	1	0	0*1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto	2	0	0*2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica	3	6	6*3
TOTAL				18

-Para calcular el Factor de Peso de los Casos de Uso sin ajustar (UUCW)

Tipo de CU	Descripción	Peso	Cantidad de CU	Cant CU * peso
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	4	20
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	4	40
Complejo	El caso de uso tiene más de 8 transacciones.	15	6	90
TOTAL				150



CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD

$$UUCP = UAW + UUCW$$

$$UUCP = 18 + 150$$

$$UUCP = 168$$

5.1.2 CÁLCULO DE PUNTOS DE CASOS DE USO AJUSTADOS.

$$UCP = UUCP * TCF * EF$$

Donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Factor de complejidad técnica (TCF)

$$TCF = 0.6 + 0.01 * \text{Ó} (\text{Pesoi} * \text{Valori}) \text{ (Donde Valor es un número del 0 al 5)}$$

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante, esto está reflejado en la siguiente tabla:

Significado de los valores

0: No presente o sin influencia

1: Influencia incidental o presencia incidental

2: Influencia moderada o presencia moderada

3: Influencia media o presencia media



CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD

4: Influencia significativa o presencia significativa

5: Fuerte influencia o fuerte presencia

Factor	Descripción	Peso	Valor asignado	Peso* valor asignado
T1	Sistema distribuido	2	4	8
T2	Tiempo de respuesta	1	2	2
T3	Eficiencia del usuario final	1	3	3
T4	Procesamiento interno complejo	1	3	3
T5	El código debe ser reutilizable	1	5	5
T6	Facilidad de instalación	0.5	5	2.5
T7	Facilidad de uso	0.5	5	2.5
T8	Portabilidad	2	4	8
T9	Facilidad de cambio	1	5	5
T10	Concurrencia	1	0	0
T11	Incluye objetivos especiales de seguridad	1	4	4
T12	Provee acceso directo a terceras partes	1	2	2
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	0	0



CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD

TOTAL	45
-------	----

$$TCF = 0.6 + 0.01 * 45$$

$$TCF=0.6+0.45$$

$$TCF = 1.05$$

Para Calcular Factor de ambiente (EF)

$$EF = 1.4 - 0.03 * \text{Ó (Pesoi * Valori)} \text{ (Donde Valor es un número del 0 al 5).}$$

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente.

- Para los factores del E1 al E4, un valor asignado de 0 significa sin experiencia, 3 experiencia media y 5 amplia experiencia (experto).
- Para el factor E5, 0 significa sin motivación para el proyecto, 3 motivación media y 5 alta motivación.
- Para el factor E6, 0 significa requerimientos extremadamente inestables, 3 estabilidad media y 5 requerimientos estables sin posibilidad de cambios.
- Para el factor E7, 0 significa que no hay personal part-time (es decir todos son full-time), 3 significa mitad y mitad, y 5 significa que todo el personal es part-time (nadie es full-time).
- Para el factor E8, 0 significa que el lenguaje de programación es fácil de usar, 3 medio y 5 que el lenguaje es extremadamente difícil.



CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD

Factor	Descripción	Peso	Valor asignado	Peso* valor asignado
E1	Familiaridad con el modelo de proyecto utilizado	1.5	5	7.5
E2	Experiencia en la aplicación	0.5	5	2.5
E3	Experiencia en la orientación a objetivos	1	5	5
E4	Capacidad del analista líder	0.5	3	1.5
E5	Motivación	1	5	5
E6	Estabilidad de requerimientos	2	3	6
E7	Personal Part-Time.	-1	0	0
E8	Dificultad del lenguaje de programación	-1	3	-3
TOTAL				24.5

$$EF = 1.4 - 0.03 * 24.5$$

$$EF = 1.4 - 0.735$$

$$\mathbf{EF = 0.665}$$

$$\text{Luego UCP} = \text{UUCP} * \text{TCF} * \text{EF}$$

$$\text{UCP} = 168 * 1.05 * 0.665$$

$$\mathbf{\text{UCP} = 117.306}$$



CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD

5.2. CÁLCULO DEL ESFUERZO DEL FT IMPLEMENTACIÓN

De los puntos de casos de uso a la estimación:

- Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores del E1 a E6.
- Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.
- Si el total -en este caso el total es 0- es 2 o menos, se utiliza el factor de conversión 20 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 20 horas-hombre.
- Si el total es 3 o 4, se utiliza el factor de conversión 28 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 28 horas-hombre.
- Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

Para calcular el Esfuerzo se utiliza la siguiente ecuación:

$$E = UCP * CF$$

Donde

E: esfuerzo estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: factor de conversión

-Para calcular CF

$$CF = 20 \text{ horas-hombre (si Total}_{EF} \leq 2)$$



CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD

CF = 28 horas-hombre (si Total EF = 3 ó Total EF = 4)

CF = abandonar o cambiar proyecto (si Total EF \geq 5)

Total EF = Cant EF < 3 (entre E1 –E6) + Cant EF > 3 (entre E7, E8)

Como **Total**_{EF} = 0 + 0

Total_{EF} = 0

CF = 20 horas-hombre (porque Total EF \leq 2)

Factor de Conversión (CF): 20 horas/hombres luego de haber realizado la comparación correspondiente en la tabla anterior.

El esfuerzo en horas-hombre viene dado por:

$$E = UCP * CF$$

$$E = 117.306 * 20$$

$$E = 2346.12 \sim 2346$$

Para una estimación más completa de la duración total del proyecto, hay que agregar a la obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software.

Para ello se puede tener en cuenta el siguiente criterio, que estadísticamente se considera aceptable. El criterio plantea la distribución del esfuerzo entre las diferentes actividades de un proyecto, según la siguiente aproximación:



CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD

5.3. DISTRIBUCIÓN DEL ESFUERZO ENTRE LAS DIFERENTES ACTIVIDADES DE UN PROYECTO

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10	586.5
Diseño	20	1173
Implementación	40	2346
Pruebas	15	879.75
Sobrecarga (otras actividades)	15	879.75
Total	100	5865

Obviamente, estos valores no son absolutos sino que pueden variar de acuerdo a las características de la organización y del proyecto.

Esfuerzo Total (Horas--Hombre)	ET	5865
Esfuerzo Total (Mes--Hombre)	ET	24(4262.5/240)
Salario	S	100
Cantidad de Hombres	CH	2
Costo Hombre--Mes	CHM	200
Costo Total	Costo	\$4800

Suponiendo que una persona trabaje 8 horas por día, y un mes tiene como promedio 30 días; la cantidad de horas que puede trabajar una persona en 1 mes es 240 horas

Si **ET = 5865 horas-hombre** y por cada 240 horas yo tengo 1 mes eso daría un **ET= .4 mes-hombre**



CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD

Esto quiere decir que 1 persona puede realizar el problema analizado en más o menos 24 meses y 12 días (24,4 meses).

En resumen:

Tiempo = ET / CH

Tiempo = $24/2$

Tiempo = 12 meses

De los resultados obtenidos se interpreta que con 2 hombres trabajando en el proyecto, el mismo se desarrolla en 12 meses y su costo total se estima sea de \$4800.

5.4. BENEFICIOS TANGIBLES E INTANGIBLES.

El Sistema Informático para la Gestión de Información de La Unión de Jóvenes Comunistas de la Facultad 2 tiene como objetivo fundamental recolectar, registrar y analizar una parte de la información relacionada con la UJC que se maneja en la Facultad 2. Por lo que contará con la constante interacción de los usuarios, de ahí su interfaz gráfica sencilla y amigable. Los beneficios más destacables brindados por este sistema son principalmente intangibles:

- Facilitó el trabajo de los Activistas de Funcionamiento, Proceso Político, el Documentador respectivamente y por ende el del Organizador y el Secretario del Comité Primario.
- Facilitó el acceso a la información que se necesite relacionada con la UJC de la Facultad 2 (Actas de reuniones, ID2, Militantes, Sanciones, Comités de Base).
- Hizo más eficiente el control de la información que se genera de la UJC de la Facultad 2 (Actas de reuniones, ID2, Militantes, Sanciones, Comités de Base).
- Ahorró el tiempo que se emplea en desarrollar los principales procesos que se desarrollan en la UJC de la Facultad 2, como por ejemplo: la gestión de las actas que se generan de las reuniones mensuales de los comités de base, llenar las sanciones de los militantes y obtener reportes de



CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD

ellas, gestionar la planilla de control de los militantes (ID2) y hacer el cierre de funcionamiento, procesos que se llevan mucho tiempo llevarlos a cabo sin la ayuda de un sistema automatizado.

- Posibilitó al Secretario del Comité Primario obtener importantes reportes que se le hacen imprescindibles para que la UJC en la Facultad 2 funcione de la mejor forma posible.

Como beneficio tangible se puede mencionar:

1- El Sistema **F2-UJC** (Sistema Informático para la Gestión de Información de La Unión de Jóvenes Comunistas de la Facultad 2).

Este sistema se hizo para que los usuarios que le interesen el tema de la información de la UJC de la Facultad 2 ingresen a la aplicación con los permisos que posean y que de esta misma tengan acceso a la misma en el tiempo requerido.

5.5. ANÁLISIS DE COSTO.

El Sistema Informático para la Gestión de Información de La Unión de Jóvenes Comunistas en la Facultad 2(F2-UJC) será desarrollado casi en su totalidad por software libre, lo que garantiza que los desarrolladores puedan entrar a su código fuente y hacerle los cambios que deseen realizar, para así obtener futuras versiones y acomodarlo a necesidades aún más específicas de los clientes. El desarrollo del sistema no requiere grandes gastos de recursos, ni de tiempo; los servidores que existen en la UCI y en particular los de la Facultad 2 son capaces de soportar el software y la base de datos que contiene la información. El entrenamiento a los usuarios del F2-UJC, no reportará gastos notables de tiempo pues será sencillo y de fácil manejo. Se les dará al Comité Primario de la UJC de la Facultad 2 una herramienta para la Gestión de la Información de la UJC que mejorara los problemas que dieron al traste con la realización de este trabajo, problemas que están expuestos en la Introducción de este trabajo. El Esfuerzo Total necesario para el desarrollo del F2-UJC es de 24(mes-hombre), trabajando en el proyecto 2 hombres tardaría en hacerse 12 meses, su costo se estima sea de \$4800 y por los beneficios antes mencionados se considera factible el desarrollo de este software.



CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD

CONCLUSIONES

En este capítulo se puede apreciar cómo se aplicó el método de estimación de Puntos por Casos de Uso al proyecto Sistema Informático para la Gestión de la Información de La Unión de Jóvenes Comunistas en la Facultad 2. Mediante los resultados que arrojó este método se pudo saber la estimación del tiempo que durará desarrollarlo, la estimación del costo, entre otras estimaciones. Se dieron a conocer los beneficios tangibles e intangibles que aportará el sistema al ser usado y se llegó a la conclusión de que es Factible su desarrollo por los resultados de la estimación.



CONCLUSIONES GENERALES

Utilizando la metodología *RUP* junto con el lenguaje de modelado *UML*, se logró realizar la Ingeniería de Software del sistema F2-UJC, aspecto fundamental para lograr una organización en el desarrollo del software. Para la implementación se utilizó el Framework *Symfony*, marco de trabajo que está hecho utilizando el lenguaje *PHP5*, con el mismo se logró agilizar el desarrollo del sistema, logrando realizarlo en menos tiempo que el estimado inicialmente. Para almacenar los datos relacionados con la aplicación se utilizó el Sistema Gestor de Base de Datos *PostgreSQL*, el cual permite gestionar los datos con una gran eficiencia. Llevar a cabo todo lo expuesto anteriormente permitió principalmente entregarle un sistema a la Facultad 2 capaz de llevar a cabo la gestión de la información relacionada con la Unión de Jóvenes Comunistas, logrando cumplir con el objeto del trabajo; aunque también se logró que los tesisistas logaran una mayor preparación en la rama de la informática.



RECOMENDACIONES

Una vez finalizado el desarrollo del sistema, dando cumplimiento a los objetivos del trabajo, y teniendo en cuenta las experiencias adquiridas durante el desarrollo del mismo, se recomienda:

- Que se profundice aún más en otros procesos importantes dentro de la organización que se puedan automatizar, y a partir de estos, obtener nuevas funcionalidades dentro del sistema.
- Ampliar el número de reportes brindados por la aplicación para proporcionar mayor información a los usuarios.
- Mejorar aún más la apariencia visual del sistema.
- Que mientras se está redactando el acta en el sistema, este le permita al secretario de acta ir guardando la información que el desee.
- Que las actas archivadas en la base de datos se puedan Exportar a PDF, para que estas puedan ser impresas y guardadas digitalmente.
- Extender la utilización del sistema hacia otras facultades ya que es para el beneficio de todos.



BIBLIOGRAFÍA

1. **LARMAN, C.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* Ciudad de La Habana : s.n., 2004. Vol. I.
2. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El Proceso Unificado de Desarrollo de Software.* La Habana : Félix Varela, 2004. Vol. I.
3. **LARMAN, C.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* Ciudad de La Habana : s.n., 2004. Vol. II.
4. **E., Currás.** *Caos y orden en las organizaciones del conocimiento.* 1996.
5. **Paradigm, Visual.** Increase productivity and enhance communication and collaboration efficiency by using UML. [En línea] [Citado el: 12 de 05 de 2009.] <http://www.visual-paradigm.com>.
6. Qué es HTML. [En línea] [Citado el: 07 de 05 de 2009.] <http://www.desarrolloweb.com/articulos/que-es-html.html>.
7. **Consortium, World Wide Web.** Guía Breve de CSS. [En línea] [Citado el: 12 de 05 de 2009.] <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>.
8. **Valdés, Damián Pérez.** ¿Qué es Javascript? *Maestros del Web.* [En línea] [Citado el: 20 de 05 de 2009.] <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.
9. **Group, The PHP.** PHP: Conceptos básicos. [En línea] [Citado el: 26 de 03 de 2009.] <http://www.php.net/manual/es/getting-started.php>.
10. **Fabien Potencier, François Zaninotto.** *Symfony 1.0, la guía definitiva.* [PDF] 2009.
11. **Alejandro Sena, Juan Pablo Sueiro.** *Zend Studio for Eclipse, desarrollo profesional en PHP.* [En línea] [Citado el: 15 de 04 de 2009.] <http://www.techtear.com/2008/01/22/zend-studio-for-eclipse-desarrollo-profesional-en-php/>.
12. **España, Diseño web.** Masadelante.com. *¿Qué es un servidor web (Web Servers)?* [En línea] [Citado el: 7 de 05 de 2009.] <http://www.masadelante.com/faqs/servidor-web>.
13. **Ciberaula.** Una Introducción a APACHE. [En línea] [Citado el: 05 de 04 de 2009.] http://linux.ciberaula.com/articulo/linux_apache_intro/.
14. **Consortium, World Wide Web.** Guía Breve de Servicios Web. [En línea] [Citado el: 12 de 05 de 2009.] <http://www.w3c.es/divulgacion/guiasbreves/serviciosWeb/>.



15. **L., Néstor A. Díaz.** TiendaLinux.com. *Ventajas de PostgreSQL*. [En línea] [Citado el: 20 de 03 de 2009.] http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html.
16. **Reynoso, Billy.** *Introducción a la Arquitectura de Software. V1.0*. Universidad de Buenos Aires : s.n., 2004.
17. **Colombia, Universidad Nacional de.** www.virtual.unal.edu.co. [En línea] [Citado el: 05 de 05 de 2009.] <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060038/lecciones/modulo%201/capitulo%204/conallen.htm>.



REFERENCIAS BIBLIOGRÁFICAS

1. **LARMAN, C.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* Ciudad de La Habana : s.n., 2004. Vol. I.
2. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El Proceso Unificado de Desarrollo de Software.* La Habana : Félix Varela, 2004. Vol. I.
3. **LARMAN, C.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* Ciudad de La Habana : s.n., 2004. Vol. II.
4. **E., Currás.** *Caos y orden en las organizaciones del conocimiento.* 1996.
5. **Paradigm, Visual.** Increase productivity and enhance communication and collaboration efficiency by using UML. [En línea] [Citado el: 12 de 05 de 2009.] <http://www.visual-paradigm.com>.
6. Qué es HTML. [En línea] [Citado el: 07 de 05 de 2009.] <http://www.desarrolloweb.com/articulos/que-es-html.html>.
7. **Consortium, World Wide Web.** Guía Breve de CSS. [En línea] [Citado el: 12 de 05 de 2009.] <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>.
8. **Valdés, Damián Pérez.** ¿Qué es Javascript? *Maestros del Web.* [En línea] [Citado el: 20 de 05 de 2009.] <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.
9. **Group, The PHP.** PHP: Conceptos básicos. [En línea] [Citado el: 26 de 03 de 2009.] <http://www.php.net/manual/es/getting-started.php>.
10. **Fabien Potencier, François Zaninotto.** *Symfony 1.0, la guía definitiva.* [PDF] 2009.
11. **Alejandro Sena, Juan Pablo Sueiro.** *Zend Studio for Eclipse, desarrollo profesional en PHP.* [En línea] [Citado el: 15 de 04 de 2009.] <http://www.techtear.com/2008/01/22/zend-studio-for-eclipse-desarrollo-profesional-en-php/>.
12. **España, Diseño web.** Masadelante.com. *¿Qué es un servidor web (Web Servers)?* [En línea] [Citado el: 7 de 05 de 2009.] <http://www.masadelante.com/faqs/servidor-web>.
13. **Ciberaula.** Una Introducción a APACHE. [En línea] [Citado el: 05 de 04 de 2009.] http://linux.ciberaula.com/articulo/linux_apache_intro/.
14. **Consortium, World Wide Web.** Guía Breve de Servicios Web. [En línea] [Citado el: 12 de 05 de 2009.] <http://www.w3c.es/divulgacion/guiasbreves/serviciosWeb/>.



15. **L., Néstor A. Díaz.** TiendaLinux.com. *Ventajas de PostgreSQL*. [En línea] [Citado el: 20 de 03 de 2009.] http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html.
16. **Reynoso, Billy.** *Introducción a la Arquitectura de Software. V1.0*. Universidad de Buenos Aires : s.n., 2004.
17. **Colombia, Universidad Nacional de.** www.virtual.unal.edu.co. [En línea] [Citado el: 05 de 05 de 2009.] <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060038/lecciones/modulo%201/capitulo%204/conallen.htm>.



ANEXOS

Anexo I: Descripción de los Casos de uso del Sistema

Nombre del caso de uso:	Gestionar Usuario
Actor(es):	Sec. Comité Primario (inicia)
Propósito:	Garantizar el control de los usuarios con acceso al sistema
Resumen:	El caso de uso se inicia cuando el Secretario del Comité Primario accede a la interfaz de Gestión de Usuario del sistema y selecciona la acción que desea realizar, ya sea insertar usuario, modificar usuario, mostrar usuarios del C/P, mostrar usuarios de los C/B o eliminar un usuario del C/P. Finaliza el caso de uso con la ejecución de dicha acción.
Referencias:	R2.1 R2.2 R2.3 R2.4 R2.5
Precondiciones:	El Secretario del Comité Primario debe estar registrado en la base de datos del sistema con el Cargo de Secretario del Comité Primario, en el caso de que desee eliminar, modificar o mostrar un usuario determinado, este debe encontrarse registrado.
Poscondiciones:	Queda actualizado el estado de los usuarios con acceso al sistema, se registra un nuevo usuario, se modifica un usuario, se muestran los usuarios del Comité Primario y los del Comité de Base se muestran por grupos y se elimina un usuario.
Curso Normal de Eventos:	
Acciones del Actor	Respuesta del Sistema
1-El Secretario del Comité Primario accede a la interfaz de Gestión de Usuario. 2-El Secretario del Comité Primario elige la acción a realizar.	1.1- El sistema muestra una serie de acciones a realizar. 2.1- Si elige a) Insertar un nuevo usuario ir a la sección "Insertar Usuario" b) Modificar usuario ir a la sección "Modificar Usuario" c) Mostrar Usuarios del C/P ir a la sección "Mostrar Usuarios del C/P"



ANEXOS

	<p>d)Mostrar Usuarios de los C/B ir a la sección “Mostrar Usuarios de los C/B”</p> <p>e) Eliminar un usuario ir a la sección “Eliminar Usuario”</p>
Sección “Insertar Usuario”	
<p>2-El Secretario del Comité Primario introduce los datos para registrar al nuevo usuario (usuario, cargo).</p>	<p>1-El sistema muestra la interfaz de insertar usuario, mostrando un formulario con los campos generales que se deben introducir.</p> <p>2.1- El sistema verifica que el campo de usuario esté lleno y que haya seleccionado un cargo.</p> <p>2.2- El sistema verifica que no exista ese usuario.</p> <p>2.3- El sistema verifica que el cargo seleccionado no esté asignado a ningún usuario.</p> <p>2.4- El sistema inserta el nuevo usuario en la base de datos del sistema, le informa al Secretario del Comité Primario que se ha efectuado el registro y finaliza el caso de uso.</p>
Curso Alterno	
	<p>2.1 El sistema envía un mensaje de que se debe llenar el campo de usuario y escoger un cargo. Regresa al paso (2) del flujo normal de eventos.</p> <p>2.2 Si el usuario esta registrado se envía un mensaje informativo que el ese usuario ya existe. Regresa al paso (2) del flujo normal de eventos.</p> <p>2.3 Si el cargo ya tiene usuario asignado se envía un mensaje. Regresa al paso (2) del flujo normal de eventos.</p>
Sección “Modificar Usuario”	
<p>2-El Secretario del Comité Primario selecciona el cargo que desea modificar.</p> <p>3-El Secretario del Comité Primario realiza los cambios pertinentes.</p>	<p>1- El sistema muestra la interfaz de modificar usuario, mostrando un formulario con los cargos existentes en la base de datos.</p> <p>2.1- El sistema verifica que haya seleccionado un cargo.</p> <p>2.2- El sistema verifica que este cargo tenga usuario asignado.</p> <p>2.3- El sistema muestra el usuario del cargo seleccionado.</p> <p>3.1- El sistema verifica que los campos obligatorios estén llenos.</p> <p>3.2- El sistema verifica que ese usuario no posea cargo.</p> <p>3.3- El sistema guarda los datos modificados, le informa al Secretario del Comité Primario que se ha efectuado la modificación y finaliza el caso de uso.</p>
Curso Alterno	



ANEXOS

	<p>2.1 El sistema envía un mensaje informativo que se debe seleccionar un cargo. Regresa al paso (2) del flujo normal de eventos.</p> <p>2.2 El sistema envía un mensaje informativo de que este cargo no tiene usuario asignado. Regresa al paso (2) del flujo normal de eventos.</p> <p>3.1 El sistema envía un mensaje solicitando llenar el campo usuario. Regresa al paso (2) del flujo normal de eventos.</p> <p>3.2 Si el usuario esta registrado se envía un mensaje informativo que el ese usuario ya tiene cargo asignado. Regresa al paso (2) del flujo normal de eventos.</p>
Sección “Mostrar Usuarios del C/P”	
	2.1- El sistema muestra un listado con los usuarios del comité primario.
Sección “Mostrar Usuarios de los C/B”	
2-El Secretario del Comité Primario selecciona el comité de base del que desea mostrar sus usuarios.	<p>1- El sistema muestra la interfaz de mostrar usuarios de los comités de base, mostrando un formulario con los comités de base existentes en la base de datos.</p> <p>2.1- El sistema verifica que haya seleccionado un comité de base.</p> <p>2.2- El sistema muestra el listado de los militantes de ese comité de base y finaliza el caso de uso.</p>
Curso Alterno	
	2.1 El sistema envía un mensaje informativo que se debe seleccionar un comité de base. Regresa al paso (2) del flujo normal de eventos.
Sección “Eliminar Usuario”	
2-El Secretario del Comité Primario escoge el usuario a eliminar.	<p>1- El sistema muestra la interfaz de eliminar usuario, mostrando un formulario con los usuarios del comité primario existentes en la base de datos.</p> <p>2.1- El sistema verifica que haya seleccionado un usuario.</p> <p>2.2- El sistema elimina el usuario seleccionado, le informa al Secretario Comité Primario que se ha eliminado el usuario y finaliza el caso de uso.</p>
Curso Alterno	
	2.1 El sistema envía un mensaje informativo que se debe seleccionar un usuario. Regresa al paso (2) del flujo normal de eventos.
Prioridad:	Crítico

Tabla 32: Descripción del caso de uso del sistema Gestionar Usuario.



ANEXOS

Nombre del caso de uso:	Eliminar Acta
Actor(es):	Act. Funcionamiento (inicia)
Propósito:	Eliminar el acta de la reunión, del comité de base que el Activista de Funcionamiento desee.
Resumen:	El caso de uso se inicia cuando el Activista de Funcionamiento accede al sistema y escoge la opción Eliminar Acta. Finaliza cuando el sistema ha eliminado el acta escogida.
Referencias:	R4.5
Precondiciones:	Deben existir actas archivadas en el sistema.
Poscondiciones:	Queda eliminada por el sistema el acta seleccionada.
Curso Normal de Eventos:	
Acciones del Actor	Respuesta del Sistema
1-El Activista de Funcionamiento escoge la opción Eliminar Acta.	1.1- El sistema muestra la interfaz de Eliminar Acta, mostrando un formulario con los comités de base de la Facultad 2.
2- El Activista de Funcionamiento escoge el comité de base del que quiere eliminar un acta.	2.1- El sistema verifica que el campo del comité de base esté lleno. 2.2- El sistema muestra un listado que tiene la fecha de realización de las actas.
3- El Activista de Funcionamiento escoge por la fecha, el acta que desea eliminar.	3.1- El sistema elimina el acta seleccionada, le informa al Activista de Funcionamiento que se ha eliminado el acta y finaliza el caso de uso.
Curso Alterno	
	2.1 El sistema envía un mensaje solicitando escoger un comité de base. Regresa al paso (2) del flujo normal de eventos.
Prioridad:	Secundario

Tabla 33: Descripción del caso de uso del sistema Eliminar Acta.



ANEXOS

Nombre del caso de uso:	Cambiar Sec. del C/P
Actor(es):	Act. Funcionamiento (inicia)
Propósito:	Sustituir al Secretario del Comité Primario de la Facultad 2.
Resumen:	El caso de uso se inicia cuando el Activista de Funcionamiento accede al sistema y escoge la opción Cambiar Secretario del C/P. Finaliza cuando el sistema ha cambiado el usuario del viejo secretario por el usuario del nuevo secretario.
Referencias:	R2.6
Precondiciones:	Debe existir un secretario del comité primario registrado en la base de datos del sistema.
Poscondiciones:	Queda modificado por el sistema el secretario del comité primario.
Curso Normal de Eventos:	
Acciones del Actor	Respuesta del Sistema
1-El Activista de Funcionamiento escoge la opción Cambiar Secretario del C/P. 2- El Activista de Funcionamiento introduce el usuario del nuevo secretario del comité primario.	1.1- El sistema muestra la interfaz de Cambiar Secretario del C/P, mostrando un formulario con el usuario del secretario del comité primario. 2.1- El sistema verifica que el campo de usuario esté lleno. 2.2- El sistema verifica que el usuario introducido no esté registrado en la base de datos. 2.3- El sistema modifica el usuario del secretario del comité primario, le informa al Activista de Funcionamiento que se ha modificado el secretario del comité primario y finaliza el caso de uso.
Curso Alterno	
	2.1 El sistema envía un mensaje solicitando llenar el campo de usuario. Regresa al paso (2) del flujo normal de eventos. 2.2 El sistema envía un mensaje informativo que el usuario ya está registrado en la base de datos. Regresa al paso (2) del flujo normal de eventos.
Prioridad:	Secundario

Tabla 34: Descripción del caso de uso del sistema Cambiar Sec. del C/P



ANEXOS

Nombre del caso de uso:	Gestionar ID2
Actor(es):	Documentador (inicia)
Propósito:	Garantizar el control de los militantes de la Facultad 2
Resumen:	El caso de uso se inicia cuando el Documentador accede a la interfaz de Gestionar ID2 y selecciona la acción que desea realizar, ya sea crear, modificar o mostrar un ID2 y termina el caso de uso con la ejecución de dicha acción.
Referencias:	R5.1 R5.2 R5.3
Precondiciones:	El usuario debe estar autenticado como Documentador en la base de datos del sistema, en el caso de que se desee modificar o mostrar un ID2 determinado, este debe encontrarse registrado en el sistema.
Poscondiciones:	Queda actualizado el estado de los ID2 que se encuentran almacenados en la base de datos del sistema, se crea un nuevo ID2, se modifican y se muestran los datos de un ID2.
Curso Normal de Eventos:	
Acciones del Actor	Respuesta del Sistema
1-El Documentador accede a la interfaz de Gestión de ID2. 2-El Documentador elige la acción a realizar.	1.1- El sistema muestra una serie de acciones a realizar. 2.1- Si elige a) Crear un nuevo ID2 ir a la sección "Crear ID2" b) Modificar ID2 ir a la sección "Modificar ID2" c) Mostrar ID2 ir a la sección "Mostrar ID2"
Sección "Crear ID2"	
2- El Documentador selecciona un comité de base para crear su ID2.	1-El sistema muestra la interfaz de crear ID2, mostrando un formulario con los comités de base que están registrados en la base de datos. 2.1- El sistema verifica que haya seleccionado un comité de base. 2.2- El sistema muestra un formulario con los campos que el usuario opcionalmente puede llenar.
3- El Documentador introduce los	3.1- El sistema registra los datos ingresados por el usuario en la base de



ANEXOS

datos en estos campos.	datos y muestra un mensaje informativo al usuario y finaliza el caso de uso.
Curso Alterno	
	<p>2.1 El sistema envía un mensaje solicitando escoger un comité de base. Regresa al Paso (2) del flujo normal de eventos.</p> <p>2.2 Si los datos no son válidos se envía un mensaje informativo de que hay error en los datos introducidos. Regresa al Paso (3) del flujo normal de eventos.</p>
Sección “Modificar ID2”	
<p>2- El Documentador selecciona un comité de base para modificarlo.</p> <p>3- El Documentador introduce los nuevos datos en estos campos.</p>	<p>1-El sistema muestra la interfaz de modificar ID2, mostrando un formulario con los comités de base que están registrados en la base de datos.</p> <p>2.1- El sistema verifica que haya seleccionado un comité de base.</p> <p>2.2- El sistema muestra un formulario donde aparecen los datos que una vez fueron insertados brindando la posibilidad de cambiar sus valores.</p> <p>3.1-El sistema verifica los nuevos datos introducidos.</p> <p>3.2-El sistema modifica los datos ingresados por el usuario en la base de datos y muestra un mensaje informativo al usuario y finaliza el caso de uso.</p>
Curso Alterno	
	<p>2.1 El sistema envía un mensaje solicitando escoger un comité de base. Regresa al Paso (2) del flujo normal de eventos.</p> <p>3.1 Si los datos no son válidos se envía un mensaje informativo de que hay error en los datos introducidos. Regresa al Paso (3) del flujo normal de eventos.</p>
Sección “Mostrar ID2”	
<p>2- El Documentador selecciona un comité de base para ver su ID2.</p>	<p>1-El sistema muestra la interfaz de mostrar ID2, mostrando un formulario con los comités de base que están registrados en la base de datos.</p> <p>2.1- El sistema verifica que haya seleccionado un comité de base.</p> <p>2.2- El sistema muestra la información del ID2 del comité de base seleccionado.</p>
Curso Alterno	
	<p>2.1 El sistema envía un mensaje solicitando escoger un comité de base. Regresa al paso (2) del flujo normal de eventos.</p>



ANEXOS

Prioridad:	Secundario
-------------------	------------

Tabla 35: Descripción del caso de uso del sistema Gestionar ID2.

Nombre del caso de uso:	Ver Comité de Base
Actor(es):	Militante (inicia)
Propósito:	Mostrar la estructura del comité de base al cual pertenece el militante.
Resumen:	El caso de uso se inicia cuando el Militante accede al sistema y escoge la opción Ver Comité de Base. Finaliza cuando el sistema muestra un listado con los nombres, apellidos y el cargo de cada uno de los integrantes del comité de base al cual pertenece el militante.
Referencias:	R3.5
Precondiciones:	El usuario debe estar registrado en la base de datos del sistema con el cargo de militante y debe estar creado su comité de base.
Poscondiciones:	Se muestra un listado con la estructura del comité de base del cual el militante es miembro.
Curso Normal de Eventos:	
Acciones del Actor	Respuesta del Sistema
1-El Militante accede a la interfaz a la cual él tiene acceso.	1.1- El sistema muestra una serie de acciones a realizar.
2- El Militante escoge la opción Ver C/B.	2.1- El sistema muestra un listado con los integrantes del comité de base del militante, en el cual aparecen el nombre, apellidos y cargo de cada uno y así finaliza el caso de uso.
Prioridad:	Secundario

Tabla 36: Descripción del caso de uso del sistema Ver Comité de Base.

Nombre del caso de uso:	Ver Acta
Actor(es):	Militante (inicia)



ANEXOS

Propósito:	Mostrarle las actas que se realizan durante las reuniones ordinarias mensuales al militante.
Resumen:	El caso de uso se inicia cuando el Militante accede al sistema y escoge la opción Ver Acta. Finaliza cuando el sistema muestra el acta escogida por el militante.
Referencias:	R4.6
Precondiciones:	El usuario debe estar registrado en la base de datos del sistema con el cargo de militante, debe estar creado su comité de base y debe tener archivada al menos un acta.
Poscondiciones:	Se muestra el acta seleccionada por el militante.
Curso Normal de Eventos:	
Acciones del Actor	Respuesta del Sistema
1-El Militante accede a la interfaz a la cual él tiene acceso.	1.1- El sistema muestra una serie de acciones a realizar.
2- El Militante escoge la opción Ver Acta.	2.1- El sistema muestra un listado con la fecha de realización de cada una de las actas archivadas en la base de datos del sistema.
3- El militante escoge por la fecha de realización el acta que desea ver.	3.1- El sistema muestra una plantilla con los datos del acta seleccionada y así finaliza el caso de uso.
Prioridad:	Secundario

Tabla 37: Descripción del caso de uso del sistema Ver Acta.

Nombre del caso de uso:	Ver Actas
Actor(es):	Act. Funcionamiento (inicia)
Propósito:	Mostrarle las actas de los comités de base de la Facultad 2 que se encuentran archivadas en la base de datos del sistema al Activista de Funcionamiento para que las revise.
Resumen:	El caso de uso se inicia cuando el Activista de Funcionamiento accede al sistema y escoge la opción Ver Actas. Finaliza cuando el sistema muestra el acta escogida por el Activista de Funcionamiento.



ANEXOS

Referencias:	R4.7
Precondiciones:	El usuario debe estar registrado en la base de datos del sistema con el cargo de Activista de Funcionamiento, deben estar creados los comités de base de la Facultad 2 y debe tener archivadas al menos un acta por comité de base.
Poscondiciones:	Se muestra el acta seleccionada por el Activista de Funcionamiento.
Curso Normal de Eventos:	
Acciones del Actor	Respuesta del Sistema
<p>1-El Activista de Funcionamiento accede a la interfaz a la cual él tiene acceso.</p> <p>2- El Activista de Funcionamiento escoge la opción Ver Actas.</p> <p>3- El Activista de Funcionamiento escoge un comité de base del cual quiere ver sus actas.</p> <p>4- El Activista de Funcionamiento escoge por la fecha de realización el acta que desea ver.</p>	<p>1.1- El sistema muestra una serie de acciones a realizar.</p> <p>2.1- El sistema muestra un formulario con los comités de base existentes en la base de datos del sistema.</p> <p>3.1- El sistema verifica que se halla seleccionado un comité de base 3.2- El sistema muestra un listado con la fecha de realización de cada una de las actas archivadas en la base de datos del sistema de el comité de base seleccionado.</p> <p>4.1- El sistema muestra una plantilla con los datos del acta seleccionada y así finaliza el caso de uso.</p>
Curso Alterno	
	3.1 El sistema envía un mensaje solicitando escoger un comité de base. Regresa al paso (3) del flujo normal de eventos.
Prioridad:	Secundario

Tabla 38: Descripción del caso de uso del sistema Ver Actas.



ANEXOS

DIAGRAMA DE CLASES DEL DISEÑO USANDO ESTEREO TIPOS WEB

Cambiar Sec. del C/P

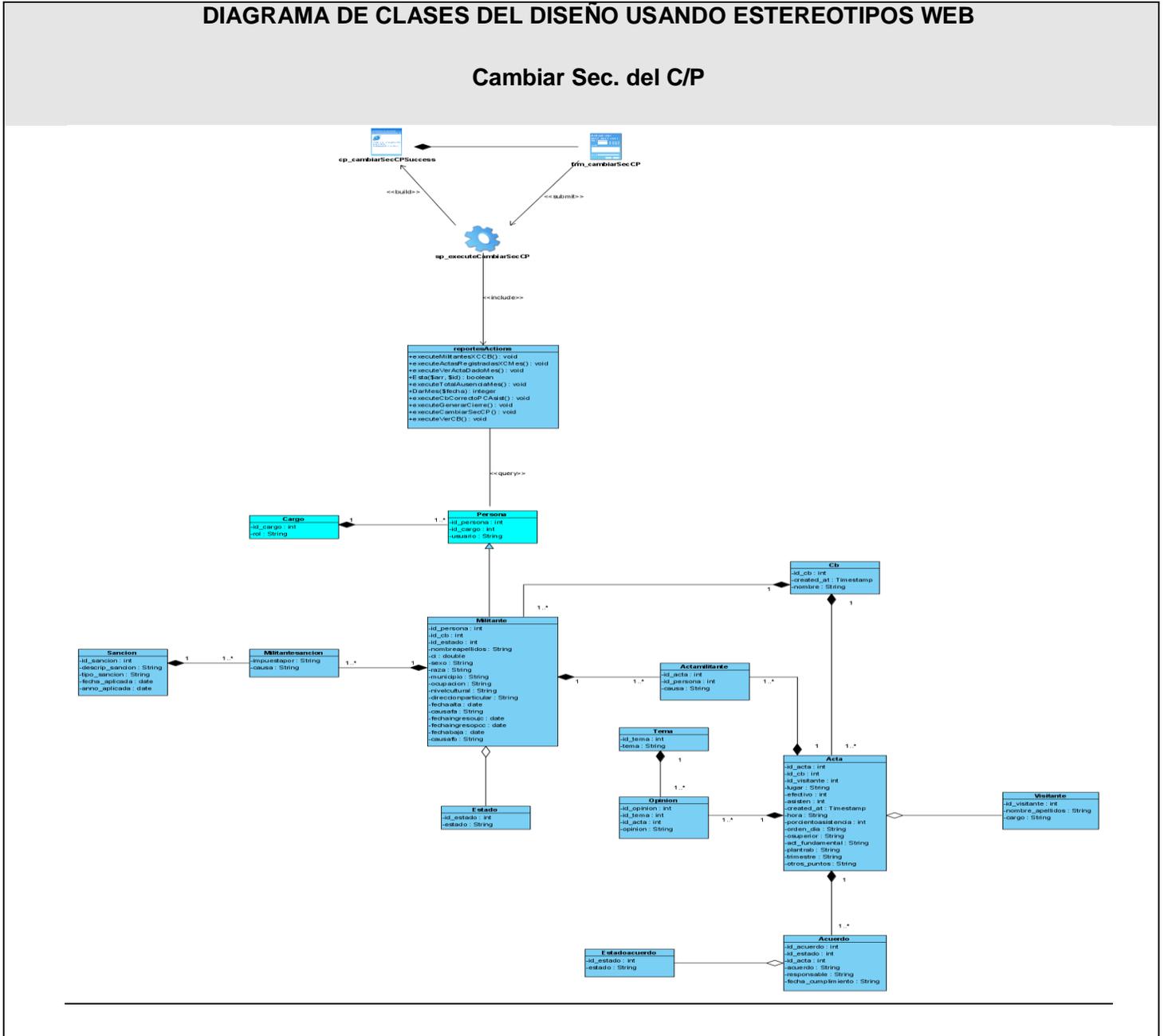


Figura 38: DCD usando estereotipos web del Caso de Uso del Sistema Cambiar Sec. del C/P.



ANEXOS

DIAGRAMA DE CLASES DEL DISEÑO USANDO ESTEREOTIPOS WEB

Ver Comité de Base

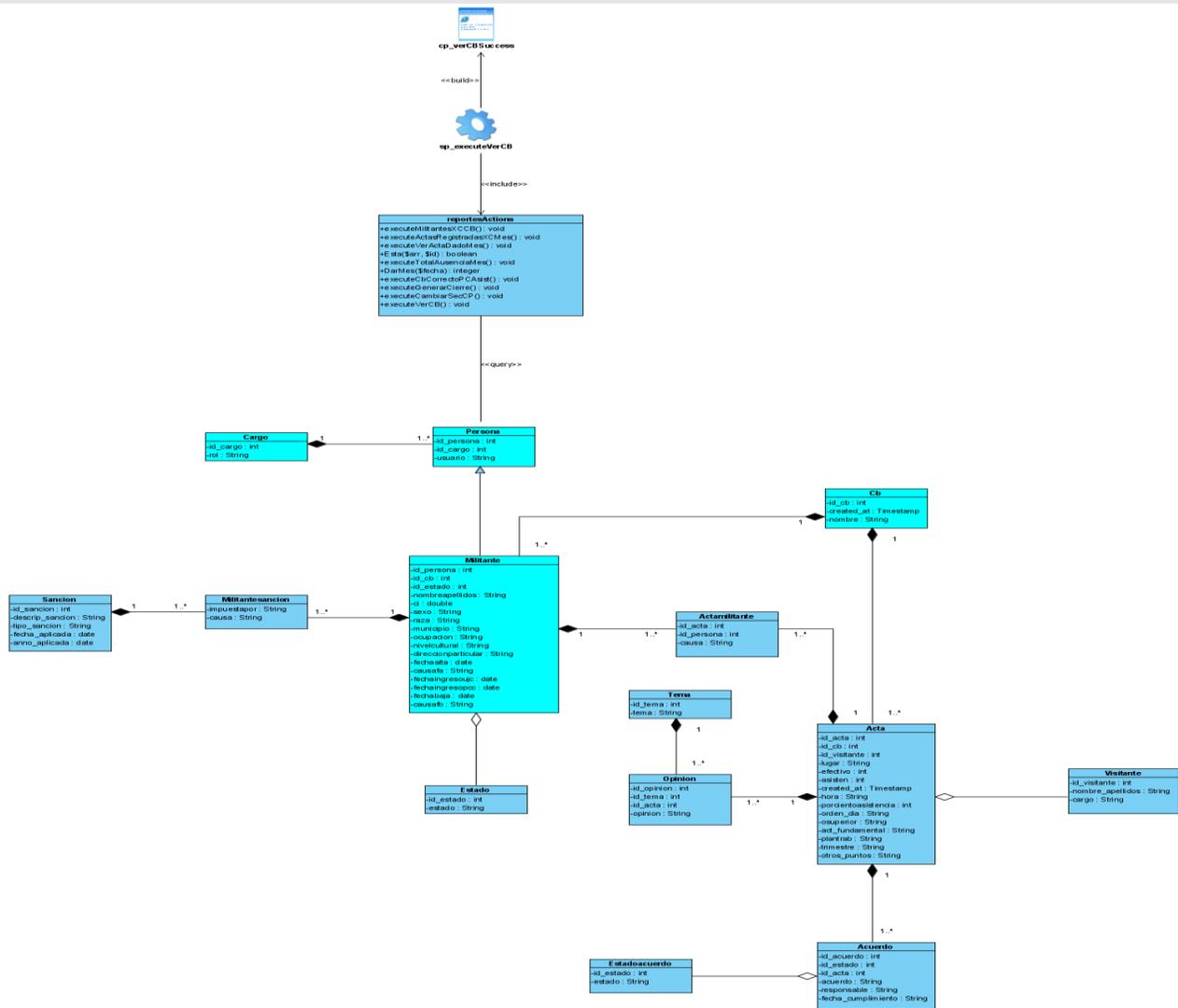


Figura 40: DCD usando estereotipos web del Caso de Uso del Sistema Ver Comité de Base.



ANEXOS

DIAGRAMA DE SECUENCIA

Del CU Gestionar Comité Base (Escenario Insertar Comité de Base)

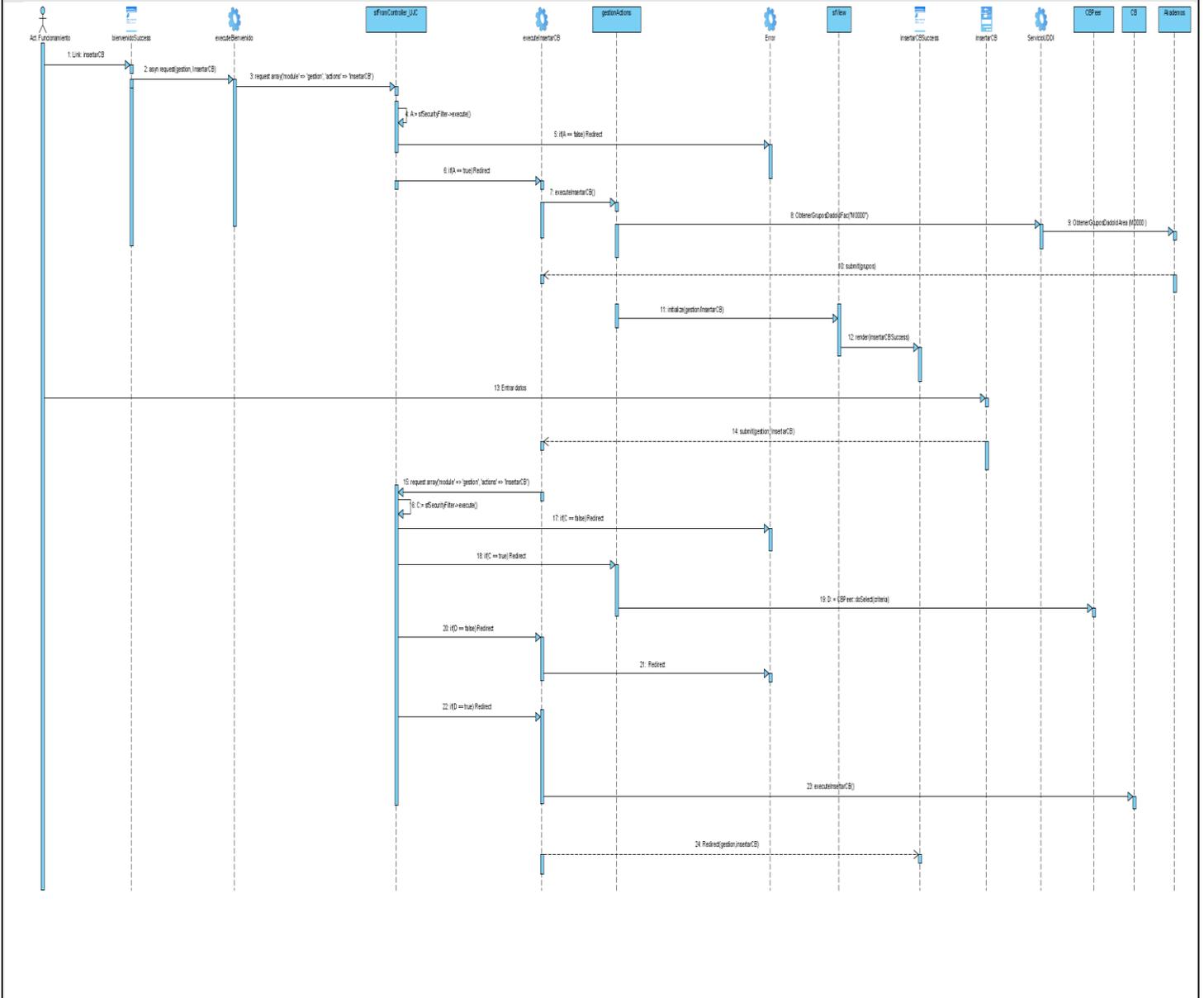


Figura 43: DS del Caso de Uso del Sistema Gestionar Comité Base (Escenario Insertar Comité de Base).



ANEXOS

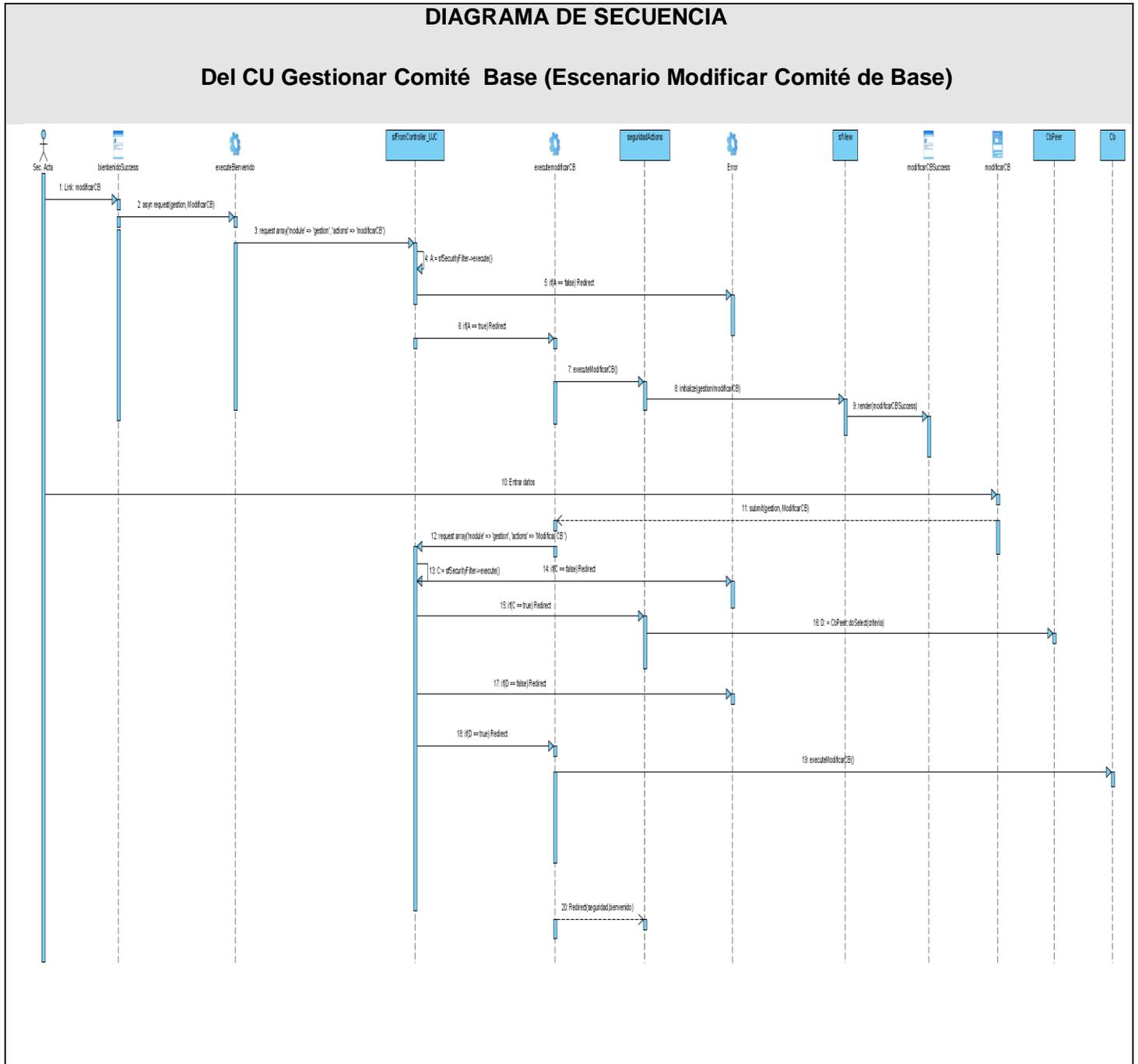


Figura 44: DS del Caso de Uso del Sistema Gestionar Comité Base (Escenario Modificar Comité de Base).



ANEXOS

DIAGRAMA DE SECUENCIA

Del CU Gestionar Sanción (Escenario Insertar Sanción)

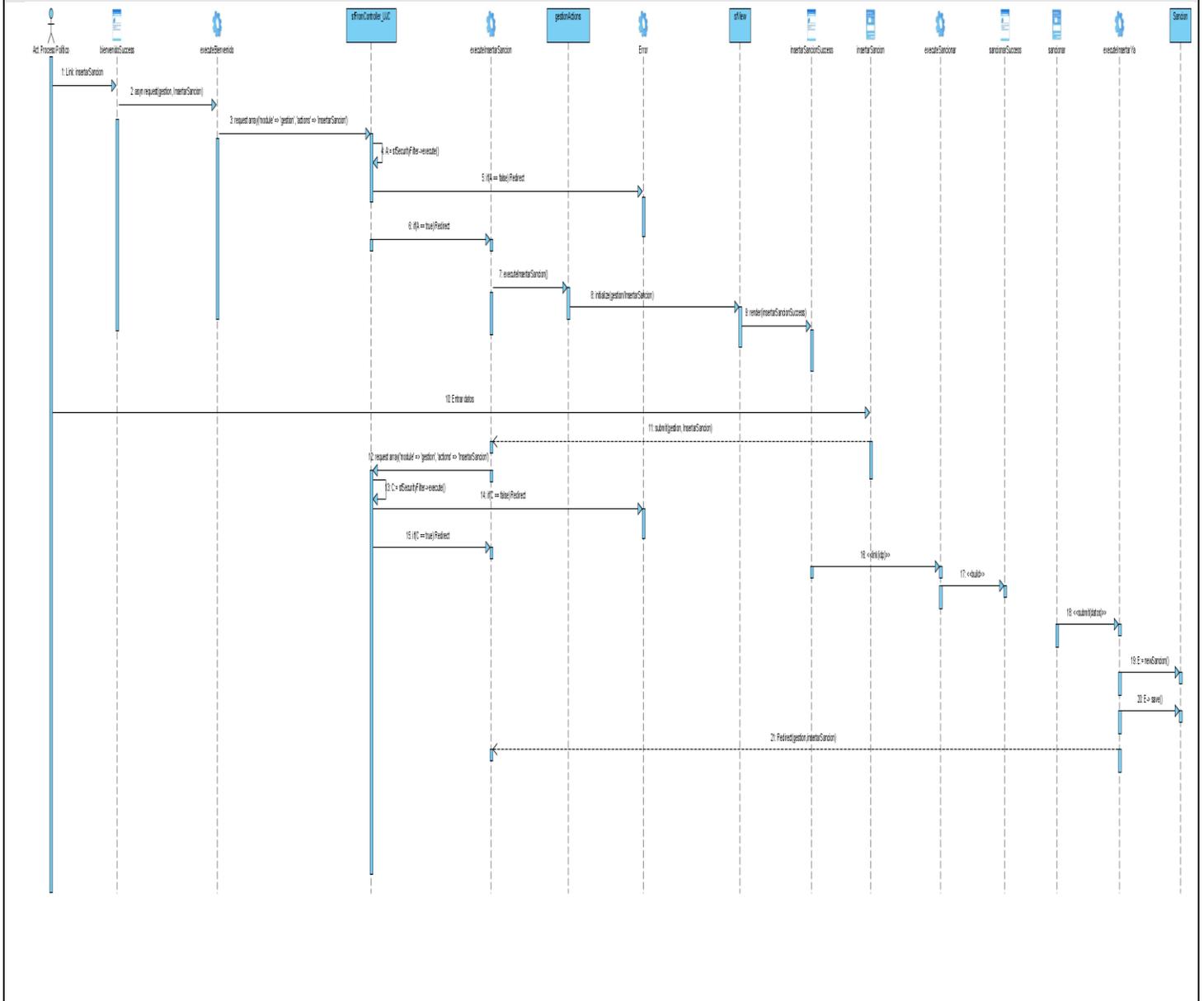


Figura 45: DS del Caso de Uso del Sistema Gestionar Sanción (Escenario Insertar Sanción).



ANEXOS

Anexo III: Diagrama de componentes del Sistema

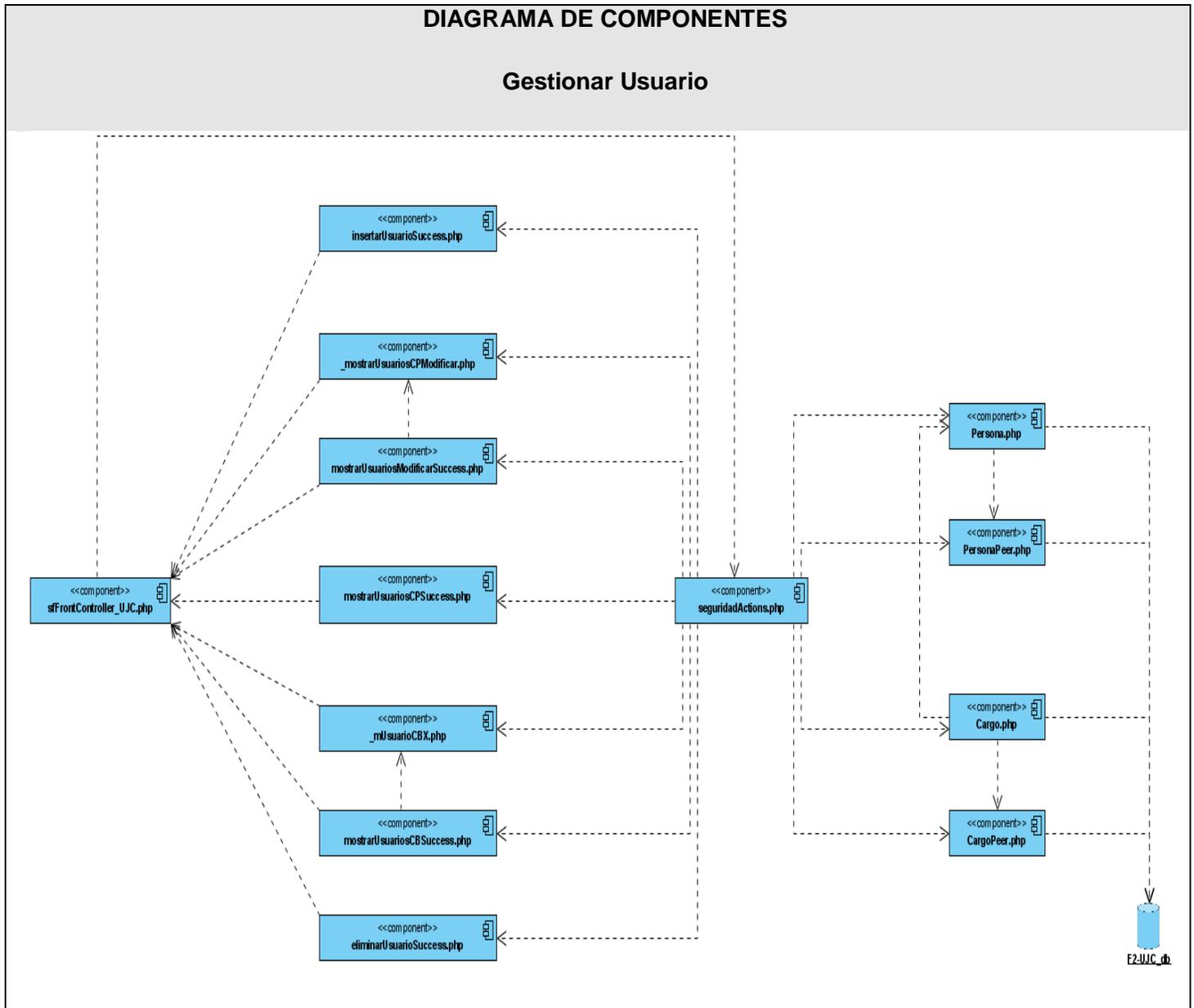


Figura 47: Diagrama de componentes del Caso de Uso del Sistema Gestionar Usuario.



ANEXOS

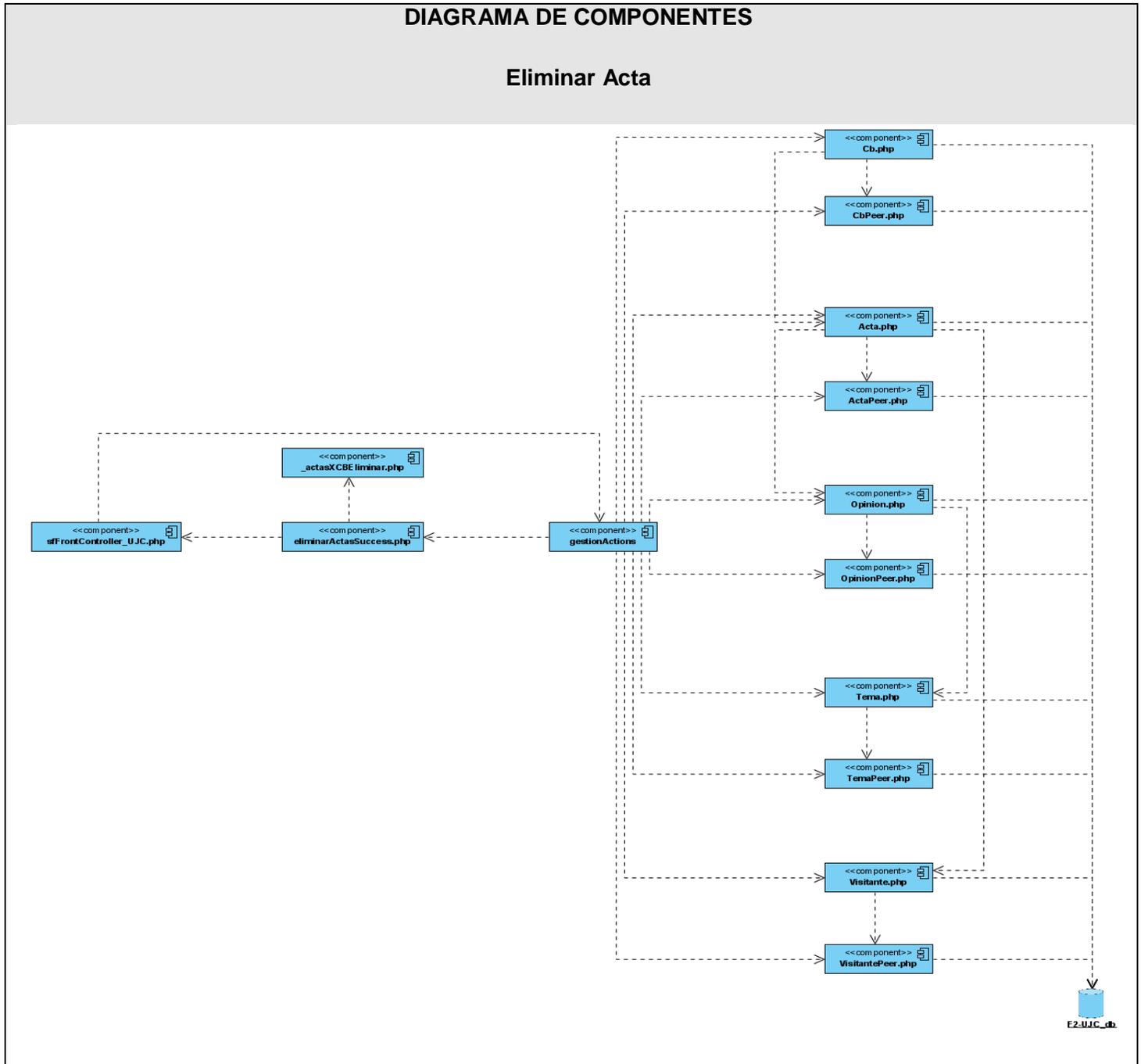


Figura 48: Diagrama de componentes del Caso de Uso del Sistema Eliminar Acta.



ANEXOS

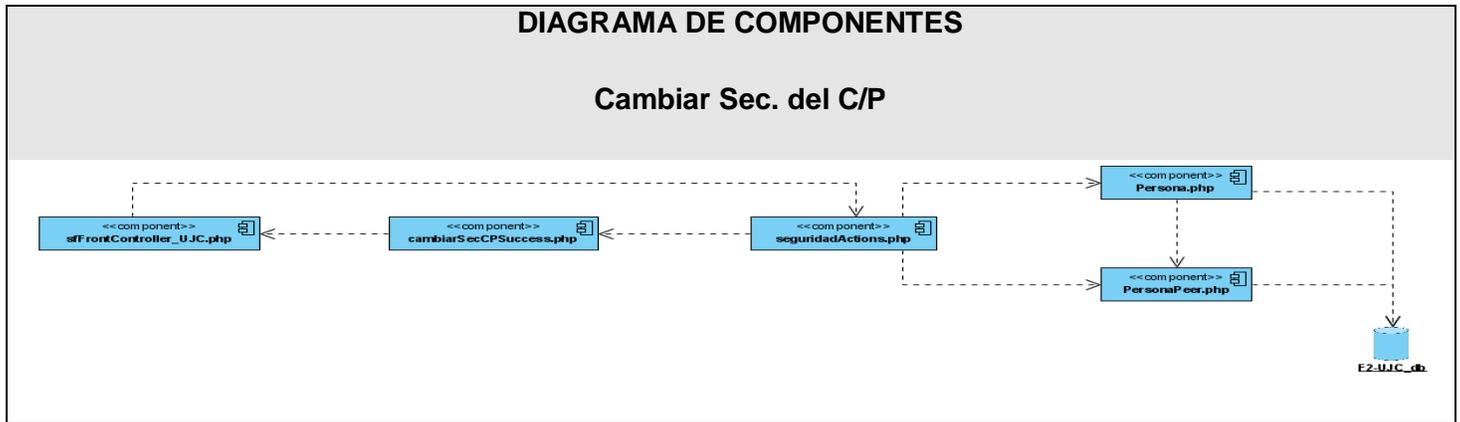


Figura 49: Diagrama de componentes del Caso de Uso del Sistema Cambiar Sec. del C/P.

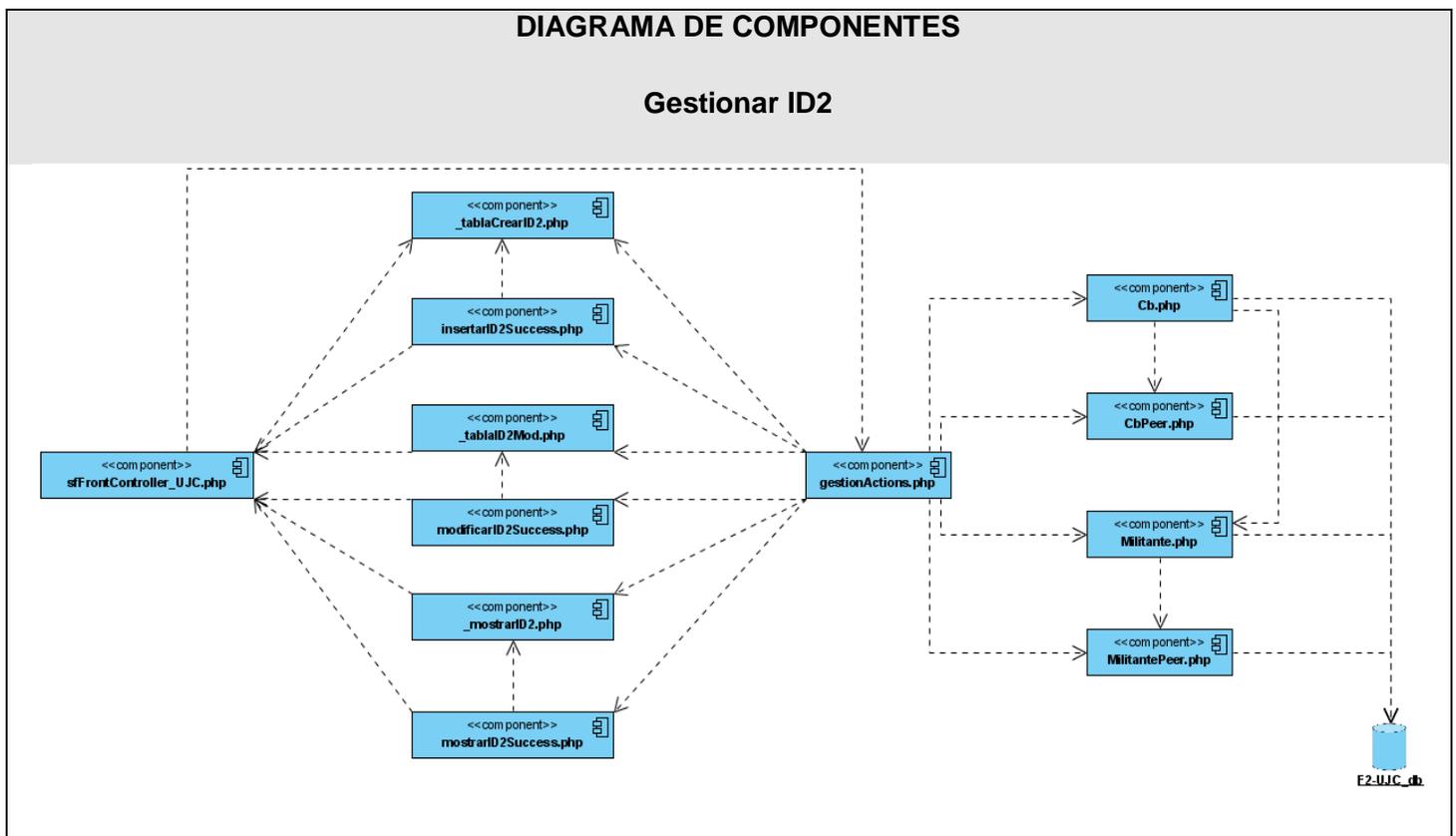


Figura 50: Diagrama de componentes del Caso de Uso del Sistema Gestionar ID2.



ANEXOS

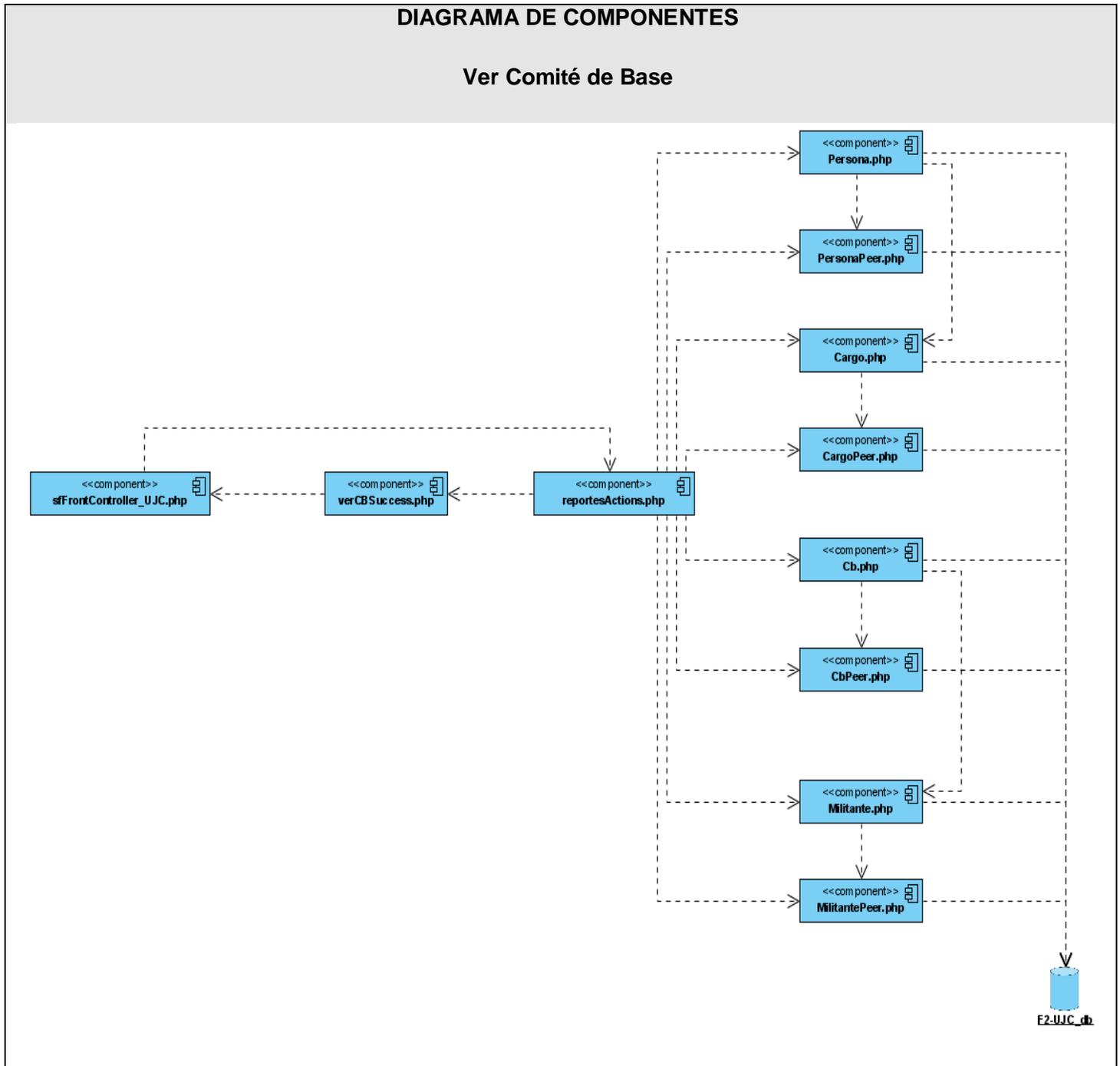


Figura 51: Diagrama de componentes del Caso de Uso del Sistema Ver Comité de Base.



ANEXOS

DIAGRAMA DE COMPONENTES

Ver Acta

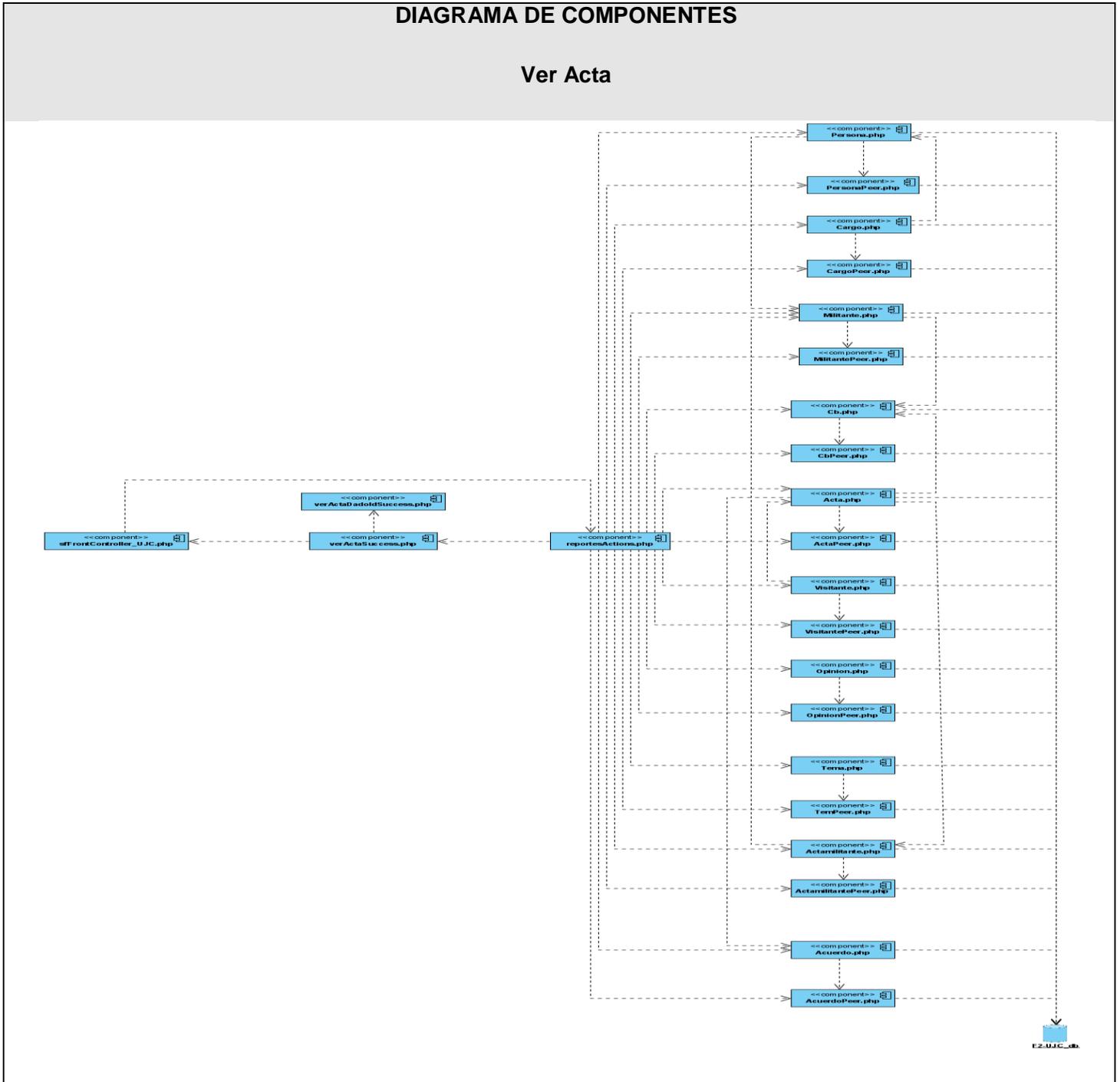


Figura 52: Diagrama de componentes del Caso de Uso del Sistema Ver Acta.



ANEXOS

DIAGRAMA DE COMPONENTES

Ver Actas

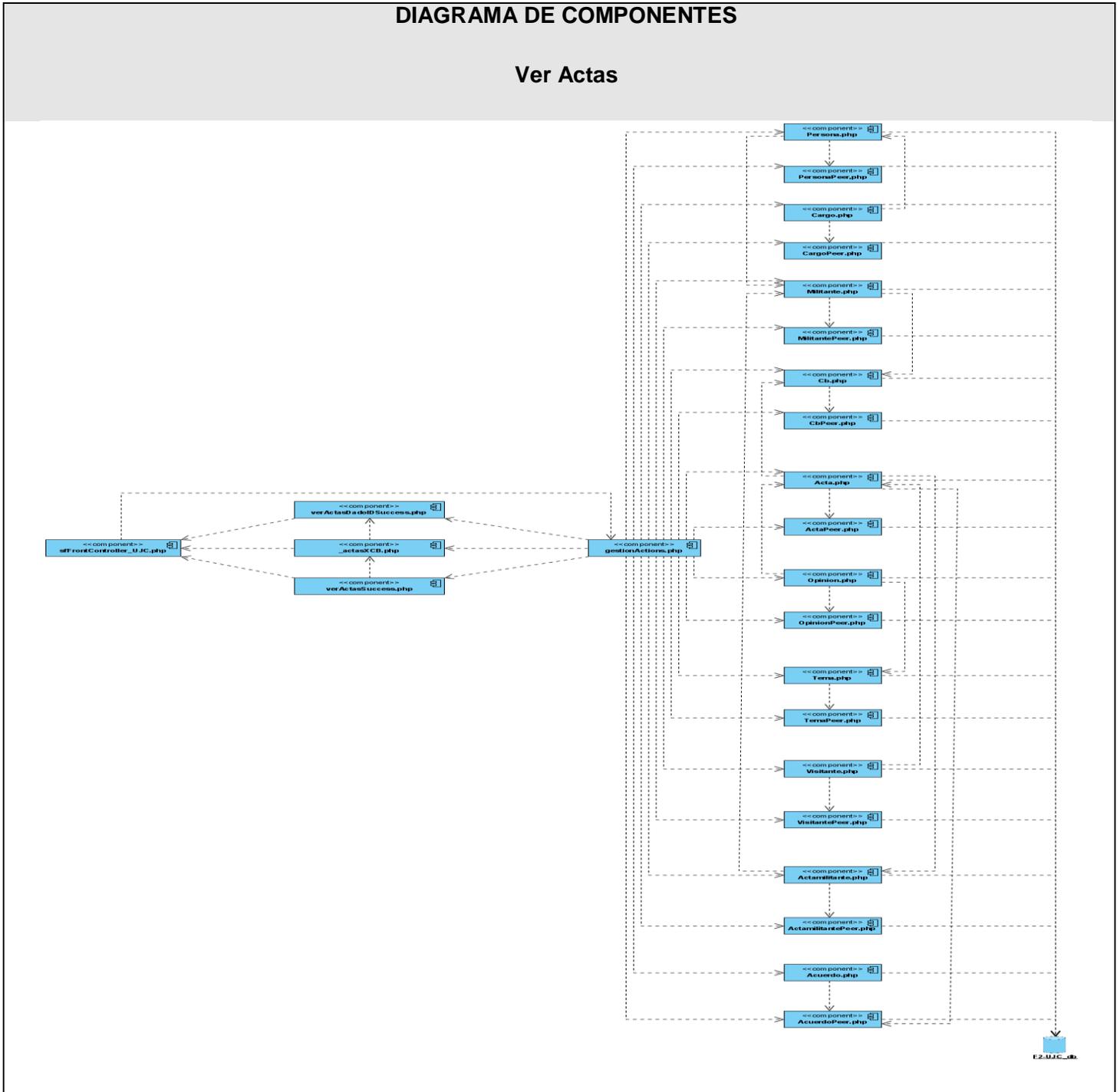


Figura 53: Diagrama de componentes del Caso de Uso del Sistema Ver Actas.



GLOSARIO DE TÉRMINOS

UCI: Universidad de las Ciencias Informáticas.

UJC: Unión de Jóvenes Comunistas.

C/B: Comité de Base

C/P: Comité Primario

CUN: Casos de Uso del Negocio.

CUS: Casos de uso del Sistema.

DCD: Diagrama de Clases del Diseño.

DS: Diagrama de Secuencia

CASE: (*Computer Aided Software*) en español: Ingeniería de Software Asistida por Computadora.

UAW: (*Unadjusted Actor Weights*) en español: Factor de peso de los actores sin ajustar.

EF: (*Environment Factor*) en español: Factor de Ambiente.

GPL: (*General Public License*) Licencia Pública General. Orientada principalmente a proteger la libre distribución, modificación y uso de software.

HTTP: (*Hypertext Transfer Protocol*) en español: Protocolo de Transferencia de Hipertexto. Modo de comunicación para solicitar páginas Web.

HTTPS: (*Hypertext Transfer Protocol Secure*) en español: Protocolo seguro de transferencia de hipertexto. Es un protocolo de red basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto, es decir, es la versión segura de HTTP. El sistema HTTPS utiliza un cifrado basado en las SSL para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP.



GLOSARIO DE TÉRMINOS

USB: (*Universal Serial Bus*) en español: Conductor Universal en Serie. Es un puerto que sirve para conectar periféricos a una computadora.

LDAP: (*Lightweight Directory Access Protocol*) en español: Es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. Habitualmente, almacena la información de login (usuario y contraseña) y es utilizado para autenticarse aunque es posible almacenar otra información.

Microsoft: Empresa multinacional estadounidense dedicada al sector de la informática.

PDF: (*Portable Document Format*) en español: Formato de Documento Portátil.

RUP: (*Rational Unified Process*) en español: Proceso Unificado de Desarrollo. Metodología para el desarrollo de software.

SOAP: (*Simple Object Access Protocol*) en español: Protocolo Simple de Acceso a Objetos. Basado en XML, que permite la interacción entre varios dispositivos y que tiene la capacidad de transmitir información compleja.

SSL: (*Secure Sockets Layer*) en español: Protocolo de Capa de Conexión Segura. Protocolo de seguridad que proporciona un método para transferir datos entre el cliente y el servidor de forma segura.

WSDL: (*Web Services Description Language*) en español: Lenguaje de Descripción de Servicios Web. Permite que un servicio y un cliente establezcan un acuerdo en lo que se refiere a los detalles de transporte de mensajes y su contenido, a través de un documento procesable por dispositivos.

XML: (*Extensible Markup Language*) en español: Lenguaje de Marcas Extensible. Es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Su función principal es describir datos y no mostrarlos. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones. Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información.



GLOSARIO DE TÉRMINOS

Frameworks: Marco de Trabajo. Los Frameworks ayudan en el desarrollo de software, proporcionan una estructura definida la cual ayuda a crear aplicaciones con mayor rapidez. Ayuda a la hora de realizar el mantenimiento del sitio gracias a la organización durante el desarrollo de la aplicación.

BSD: (*Berkeley Software Distribution*) en español: Distribución de Software Berkeley. Se utiliza para identificar un sistema operativo derivado del sistema Unix nacido a partir de los aportes realizados a ese sistema por la Universidad de California en Berkeley. Así se llamó a las distribuciones de código fuente que se hicieron en la Universidad de Berkeley en California y que en origen eran extensiones del sistema operativo UNIX.

IDE: (*Integrated Development Environment*) en español: Entorno de Desarrollo Integrado. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación.

ORM: (*Object-Relational mapping*) en español: mapeo objeto-relacional. Es un componente de software que me permite trabajar con los datos persistidos como si ellos fueran parte de una base de datos orientada a objetos (en este caso virtual), es decir, sirve para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional. En la práctica esto crea una base de datos orientada a objetos virtual, sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo). Debido a que lo *standard* es trabajar con BD relacionales, se deben realizar operaciones que permitan transformar un registro en objeto y viceversa.