

Universidad de las Ciencias Informáticas



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

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

Autores: Yaisel Avilés Ramirez

Alexis Molina González

Tutor: Ing. Antonio Marrero Palomino

Ciudad de La Habana, Cuba

Junio 2009

"Si salgo llego, si llego entro, si entro triunfo".

Fidel Castro Ruz

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado: Sistema Integral de Gestión de la UJC Nacional: Módulo Militante y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Yaisel Avilés Ramirez

Firma del Autor

Alexis Molina González

Firma del Autor

Ing. Antonio Marrero Palomino

Firma del Tutor

DEDICATORIA

A mi madre y mis abuelos que son el centro de mi vida.

A mi hermanito por ser la persona que más quiero en la vida.

A mis tíos por todo su apoyo y cariño.

A mi novio lindo por ser fuente de inspiración.

A los que pensaron que no podía.

Yaisel

Quiero dedicarle este Trabajo de Diploma a mi madre que siempre a estado presente en los buenos y malos tiempos, a mi familia, amigos y a todas esas personas que se preocuparon por mí y estuvieron pendientes de mis preocupaciones, mis sentimientos y mis problemas...

Alexis

AGRADECIMIENTOS

Hoy, cuando ya estamos a punto de graduarnos, tenemos la oportunidad de mirar atrás y ver lo difícil que ha sido el camino para llegar hasta este momento, nos damos cuenta que no caminamos este sendero a solas, siempre estuvieron familiares y amigos que sueñan a la par de nosotros desde nuestros inicios escolares en vernos convertidos en profesionales. Al término de esta etapa de nuestras vidas, quisiéramos expresar un profundo agradecimiento a quienes con su ayuda, apoyo y comprensión aunque no estuviesen cerca nos alentaron a lograr esta hermosa realidad.

Quiero agradecer a mi mamá Inia y mi papá Amado, por estar siempre en cada paso que he dado de mi vida, ser la principal fuente de inspiración para seguir adelante y contribuir con su amor a formar la profesional en la que me he convertido, por enseñarme lo bello que es tener abuelos, soy muy afortunada de tenerlos a ustedes, los quiero con todo mi corazón.

A mi bello y querido hermanito Arielito, eres la persona que más quiero en este mundo, gracias por cuidar de mi mamá en mi ausencia, trataré siempre de ser tu ejemplo y si algún día no lo lograra mira a mi corazón que el amor que siento por ti te guiará, Ari te adoro por encima de todo.

Mis agradecimientos a mis queridos Buru, Chichi y Niño por ser los mejores tíos, no piensen que los quiero, porque en realidad, los quiero más de lo que piensan.

A Leidita, Chacho y Yorcleiber por ser unos primos tan lindos.

A ti Ari que a pesar de las dificultades has contribuido mucho en mi formación, has cuidado de las personas a las que más quiero y me diste el privilegio de tener un hermano tan lindo, además te quiero como el padre que nunca tuve, no lo dudes nunca.

Quiero agradecer de forma muy especial a mi querida, bella e idolatrada madre, mi amiga, mi hermana y mi cómplice, a quien le debo todo lo que soy, por su amor, dedicación y ayuda desinteresada, por caminar conmigo este duro sendero de la vida y por dar siempre lo mejor de sí, por enseñarme a levantarme en los

momentos en que me vencía el cansancio que no fueron pocos y por brindarme siempre su mano, por enseñarme a querer y a perdonar, por brindarme su hombro para llorar, por confiar en mí y siempre estar presente, mamá Emis, éstas 140 páginas no alcanzarían para decirte lo mucho que te quiero y te estoy agradecida, me siento muy orgullosa de que seas mi madre, te querré siempre.

A mi Nene, por apoyarme en las buenas y las malas, por ser el completamiento de mi vida, por brindarme todo su amor y estar pendiente de mis cosas, por ser el hombre, compañero, amante, y el amigo ideal, agradecerte por haberme hecho una mujer mejor, por todo esto y más: "Te Quiero Mucho"

Tengo la dicha de tener dos suegras maravillosas y un suegro muy bueno, a todos gracias, a mis cuñados tan lindos y a toda esa familia por parte de novio que llegó para quedarse.

Para todos aquellos que por sobre todas las pruebas de la vida, estuvieron siempre presentes, gracias a Yoita, Yanina, Totica, y Abito, siempre serán mis amigos.

A todos mis colegas del proyecto en especial a Ale, Tony, Eraldo y Mairelys creo que hicimos un buen equipo, y si no fue así, por lo menos salimos de esta.

A mis amigas de las buenas y las malas Marlen y Adiary por estar siempre dispuestas, por quererme y soportarme, de ustedes me llevo sus enseñanzas y les dejo parte de mi corazón.

Aunque tuvimos poco tiempo para compartir no puedo dejar de mencionar a Liu, Yule, Yusma y Sissy, a la tumba me llevo este 5to año muchachitas jajajaja. Son súper OK.

La universidad me dio la dicha de conocer a los amigos que jamás voy a olvidar: Yaniel, Evelin, Ricardo, Pons y Arodys, son muy especiales para mí.

A todos aquellos que desde lejos siempre estuvieron pendientes de mis resultados, a Eliset por confiar en mí, a la India por quererme tanto, a Carmen por cocer tan rápido y a todos esos buenos vecinos que por problema de espacio no puedo poner aquí.

A ese grupo del barrio que hizo que mis pases fueran tan divertidos: a Dairina, Nela, Vivi, Pinki, Tico...bueno recuerden que ustedes son muchos....gracias a todos.

A la virgen milagrosa y a la vida que me ha dado la posibilidad de crecer y llegar hasta aquí. Gracias a la Vida que me ha dado tanto!!!!

Para todos los amigos de 5 años de estudio, de fiestas y trabajo, de lluvias y sol, de amaneceres y marchas, a todos, gracias. Y a todo aquel que luego del abrazo y del saludo, preguntó ¿Cómo va la Tesis?, muchas, muchas gracias.

Yaisel

Quiero agradecer ante todo a mi familia en especial a mi madre, mis abuelos, mi tío, mis hermanas, a Elena, mi segunda madre, a Fredy, a todos los amigos que estuvieron presentes cuando los necesité, a los que me ayudaron a llevar a cabo esta tarea entre ellos Luis, Guille, Yane, también a los que de una forma u otra me dieron su apoyo: Yulién, Víctor, Aidelís, Yasnahi, Lili, Savier, Juani, Ernesto Daniel, Jesús, Sonja, Cristian, Víctor Armando, Héctor, Jorge, el Migue, Jorge Antonio, Tony, Manuel, Yoandy, Clara, Albis...

A mis colegas del proyecto, en especial a Eraldo, a Yanisbel, Mailén, Alieski, Jose y Guillermo.

A mi compañera de tesis Yaisel jeje sin importar esos pequeños momentos de tensiones, pues fue una buena colega y juntos pudimos salir adelante.

También a esas personas que contribuyeron a mi formación como ser humano y como profesional.

Muchas gracias a todas las personas que me quieren y que no han dejado de hacerlo durante estos 5 años, a pesar de la posible lejanía que pudo haber causado esta etapa de mi vida.

Alexis

RESUMEN

La Unión de Jóvenes Comunistas (UJC) es la organización líder de la juventud cubana. Su objetivo principal está en la continua preparación cultural, política e ideológica de nuestros jóvenes en función de seguir consolidando las conquistas de nuestra Revolución, así como la formación integral y multifacética de las nuevas generaciones.

Actualmente la UJC tiene dificultades para llevar el control de las informaciones y estadísticas de sus integrantes y de la propia organización; así como del óptimo manejo de datos referentes al funcionamiento y dirección de la misma. Los trabajos se realizan utilizando métodos obsoletos, como lápiz y papel o con herramientas que no son las más idóneas para la gestión de grandes volúmenes de información. Esta situación provoca que las cifras que se gestionan sean inexactas, teniendo un margen de error elevado; además que se dificulte buscar información referente a un Comité de Base o de un miembro de la organización. Por este motivo la UJC Nacional en coordinación con la Universidad de las Ciencias Informáticas (UCI) decidió desarrollar una aplicación, enfocada a dar una solución informática a los problemas de gestión y control de las estadísticas y el funcionamiento de la organización en todo el país.

El objetivo general de este trabajo es implementar el Módulo Militante como parte del Sistema Integral para la UJC Nacional, que permita mejorar la gestión de la información de los militantes y el universo juvenil. En el presente documento se exponen las principales características de las herramientas, tecnologías y metodología empleadas.

Palabras Claves: Gestión, Información.

Contenido

INTRODUCCIÓN.....	1
1. CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1 UN POCO DE HISTORIA DE LA UJC	6
1.2 ESTADO ACTUAL DEL FUNCIONAMIENTO DE LA UJC	7
1.3 LA ESTRUCTURA DE LA UJC	8
1.4 ¿QUÉ ES UN SISTEMA DE GESTIÓN?.....	8
1.4.1 ¿POR QUÉ LOS SISTEMAS DE GESTIÓN SON NECESARIOS?.....	ERROR! BOOKMARK NOT DEFINED.
1.5 SISTEMAS DE INFORMACIÓN	9
1.6 SISTEMAS AUTOMATIZADOS EXISTENTES VINCULADOS AL CAMPO DE ACCIÓN	10
1.7 FUNDAMENTACIÓN DE LAS TECNOLOGÍAS Y HERRAMIENTAS A UTILIZAR	12
1.7.1 METODOLOGÍA DE DESARROLLO DE SOFTWARE	12
1.7.2 HERRAMIENTA CASE	13
1.7.3 LENGUAJE DE MODELADO: UML	14
1.7.4 LENGUAJE DE PROGRAMACIÓN	15
1.7.5 ENTORNO DE DESARROLLO: NETBEANS	16
1.7.6 SISTEMA DE BASE DE DATOS: DB4O.....	17
1.7.7 SISTEMA DE CONTROL DE VERSIONES: SUBVERSIÓN	19
1.8 ROLES Y ARTEFACTOS	20
1.8.1 ROL ANALISTA	20
1.8.2 ROL DISEÑADOR.....	21
1.8.3 ROL IMPLEMENTADOR.....	21
1.9 CONCLUSIONES	22
2. CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	23
2.1 MODELO DEL NEGOCIO.....	23
2.1.1 OBJETIVOS DEL MODELADO DEL NEGOCIO	23
2.1.2 ANÁLISIS DE LOS PROCESOS DE NEGOCIO	24
2.2 REGLAS DEL NEGOCIO A CONSIDERAR.....	27
2.3 ACTORES DEL NEGOCIO.....	27
2.4 TRABAJADORES DEL NEGOCIO	28

2.5	DIAGRAMAS DE CASOS DE USO DEL NEGOCIO	29
2.6	DESCRIPCIÓN DE LOS CUN Y SUS CORRESPONDIENTES DIAGRAMAS DE ACTIVIDADES	30
2.6.1	CASO DE USO GESTIONAR UJ:	31
2.6.2	DIAGRAMA DE ACTIVIDADES DEL CASO DE USO GESTIONAR UJ:	32
2.7	LEVANTAMIENTO DE REQUISITOS	32
2.7.1	OBJETIVOS DEL LEVANTAMIENTO DE REQUISITOS	33
2.8	REQUERIMIENTOS FUNCIONALES (LISTADO DE LAS CARACTERÍSTICAS BÁSICAS DEL SISTEMA)	33
2.9	REQUERIMIENTOS NO FUNCIONALES	43
2.10	ACTORES DEL SISTEMA	45
2.11	LISTADO DE CASOS DE USO	46
2.12	DIAGRAMA DE CASOS DE USO DEL SISTEMA	52
2.13	DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA	54
2.14	CONCLUSIONES	59
3.	CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	60
3.1	ESTILO ARQUITECTÓNICO: ARQUITECTURA EN TRES CAPAS	60
3.2	PATRONES DE DISEÑO	61
3.2.1	PATRONES GRASP BÁSICOS	61
3.2.2	PATRONES GOF	63
3.2.3	PATRÓN ESTRUCTURAL	64
3.2.4	PATRÓN DE ACCESO A DATOS.....	64
3.3	ANÁLISIS	65
3.3.1	DIAGRAMA DE CLASES DEL ANÁLISIS	65
3.4	DISEÑO	67
3.4.1	DIAGRAMAS DE CLASES DEL DISEÑO	67
3.4.2	DESCRIPCIÓN DE LAS CLASES DEL DISEÑO.....	69
3.4.3	DIAGRAMAS DE INTERACCIÓN	82
3.5	DISEÑO DE LA BD	83
3.5.1	DESCRIPCIÓN DE LAS CLASES PERSISTENTES.....	86
3.6	DIAGRAMA DE DESPLIEGUE	90
3.7	CONCLUSIONES	91
4.	CAPÍTULO 4: IMPLEMENTACIÓN DEL SISTEMA	92
4.1	ESTÁNDAR DE CODIFICACIÓN	92

4.2 ESTÁNDAR DE CODIFICACIÓN UTILIZADO	93
4.2.1 CONVENCIONES DE NOMBRES UTILIZADOS.....	96
4.3 DIAGRAMA DE COMPONENTES	96
4.4 MODELO DE PRUEBA.....	99
4.4.1 PRUEBAS DE CAJA NEGRA	99
4.5 ANÁLISIS DE COSTO Y BENEFICIOS.....	116
4.5.1 BENEFICIOS	122
4.6 CONCLUSIONES	123
CONCLUSIONES GENERALES.....	124
RECOMENDACIONES	125
REFERENCIAS BIBLIOGRÁFICAS	126
BIBLIOGRAFÍAS.....	128

ÍNDICE DE FIGURAS

Figura 1: Diagrama de CUN	30
Figura 2: Diagrama de Actividades del Caso de uso Gestionar UJ.....	32
Figura 3: Diagrama de caso de uso del sistema	53
Figura 4: DCA CU_ Gestionar Datos UJ.....	66
Figura 5: DCD CU_ Gestionar Datos UJ.....	68
Figura 6: Modelo de clases persistentes (Parte 1)	84
Figura 7: Modelo de clases persistentes (Parte 2)	85
Figura 8: Diagrama de Despliegue.....	90
Figura: 9 Diagrama de componente del CU_ Gestionar Datos UJ	98

ÍNDICE DE TABLAS

Tabla 2.1: Actores del Negocio.....	28
Tabla 2.2: Trabajadores del Negocio	29
Tabla 2.3: Descripción del CUN Gestionar UJ	31
Tabla 2.4: Actores del sistema	46
Tabla 2.5: Resumen del caso de uso Gestionar Usuario	46
Tabla 2.6: Resumen del caso de uso Autenticar Usuario	46
Tabla 2.7: Resumen del caso de uso Gestionar Estructura	47
Tabla 2.8: Resumen del caso de uso Buscar Militante	47
Tabla 2.9: Resumen del caso de uso Gestionar Datos UJ.....	48
Tabla 2.10: Resumen del caso de uso Gestionar Crecimiento	48
Tabla 2.11: Resumen del caso de uso Crear Expediente.....	48
Tabla 2.12: Resumen del caso de uso Gestionar Expediente.....	49
Tabla 2.13: Resumen del caso de uso Realizar Alta por Crecimiento	49
Tabla 2.14: Resumen del caso de uso Realizar Alta por Ajuste.....	49
Tabla 2.15: Resumen del caso de uso Realizar Traslado Interno	49
Tabla 2.16: Resumen del caso de uso Realizar Alta por Traslado Externo	50
Tabla 2.17: Resumen del caso de uso Realizar Baja por Traslado Externo	50
Tabla 2.18: Resumen del caso de uso Realizar Baja por Ajuste	50
Tabla 2.19: Resumen del caso de uso Realizar Baja Natural.....	50
Tabla 2.20: Resumen del caso de uso Realizar Baja por Defunción	51
Tabla 2.21: Resumen del caso de uso Gestionar Centro.....	51
Tabla 2.22: Resumen del caso de uso Buscar UJ.....	51
Tabla 2.23: Resumen del caso de uso Buscar Centro.....	52
Tabla 2.24: Resumen del caso de uso Buscar Comité	52
Tabla 2.25: Descripción ampliada del caso de uso Gestionar Datos UJ.....	58
Tabla 3.1: Descripción de la clase Persona	69
Tabla 3.2: Descripción de la clase Militante	71
Tabla 3.3: Descripción de la clase MilitanteDao.....	71
Tabla 3.4: Descripción de la clase MilitanteDaoImpl.....	72
Tabla 3.5: Descripción de la clase UniversoJuvenil.....	73
Tabla 3.6: Descripción de la clase UniversoJuvenilDao	73
Tabla 3.7: Descripción de la clase UniversoJuvenilDaoImpl	74
Tabla 3.8: Descripción de la clase Crecimiento.....	75
Tabla 3.9: Descripción de la clase CrecimientoDao	75
Tabla 3.10: Descripción de la clase CrecimientoDaoImpl	76
Tabla 3.11: Descripción de la clase Movimiento	77
Tabla 3.12: Descripción de la clase MovimientoDao	77
Tabla 3.13: Descripción de la clase MovimientoDaoImpl	77

Tabla 3.14: Descripción de la clase Alta	78
Tabla 3.15: Descripción de la clase AltaDao	79
Tabla 3.16: Descripción de la clase AltaDaoImpl	81
Tabla 3.17: Descripción de la clase Baja	81
Tabla 3.18: Descripción de la clase BajaDao.....	82
Tabla 3.19: Descripción de la clase BajaDaoImpl.....	82
Tabla 3.20: Descripción de la clase Movimiento	86
Tabla 3.21: Descripción de la clase Alta	86
Tabla 3.22: Descripción de la clase Baja	87
Tabla 3.23: Descripción de la clase Crecimiento.....	88
Tabla 3.24: Descripción de la clase UniversoJuvenil.....	88
Tabla 3.25: Descripción de la clase Persona	89
Tabla 3.26: Descripción de la clase Militante	90
Tabla 4.1: Estándar de codificación.....	95
Tabla 4.2: Valores válidos e inválidos para el caso de prueba	100
Tabla 4.3: Caso de prueba con todos los datos correctos.....	101
Tabla 4.4: Caso de prueba con el nombre incorrecto	102
Tabla 4.5: Caso de prueba con el campo Nombre vacío	103
Tabla 4.6: Caso de prueba con el segundo nombre incorrecto.....	104
Tabla 4.7: Caso de prueba con el primer apellido incorrecto	104
Tabla 4.8: Caso de prueba con el campo Apellido vacío.....	105
Tabla 4.9: Caso de prueba con el segundo apellido incorrecto.....	106
Tabla 4.10: Caso de prueba con el campo Segundo Apellido vacío.....	107
Tabla 4.11: Caso de prueba con el campo Carné de Identidad vacío	108
Tabla 4.12: Caso de prueba con el carné de identidad incorrecto (cantidad de dígitos diferente de 11)	108
Tabla 4.13: Caso de prueba con el carné de identidad incorrecto (introduciendo letras)	109
Tabla 4.14: Caso de prueba con el carné de identidad incorrecto (introduciendo caracteres extraños)	110
Tabla 4.15: Caso de prueba con el campo Color de piel vacío	111
Tabla 4.16: Caso de prueba con el campo Sexo vacío.....	112
Tabla 4.17: Caso de prueba con el campo Militancia vacío	112
Tabla 4.18: Caso de prueba con el campo Nivel cultural vacío	113
Tabla 4.19: Caso de prueba con el campo Ubicación vacío.....	114
Tabla 4.20: Valores válidos e inválidos para el caso de prueba	114
Tabla 4.21: Caso de prueba con todos los datos correctos	115
Tabla 4.22: Caso de prueba con el campo Nombre vacío	115
Tabla 4.23: Caso de prueba introduciendo el nombre incorrecto.....	115
Tabla 4.24: Peso de los actores	117
Tabla 4.25: Peso de los casos de uso.....	117
Tabla 4.26: Factores para el cálculo de la complejidad técnica	119
Tabla 4.27: Factores para el cálculo del factor ambiente	120
Tabla 4.28: Esfuerzo de todo el proyecto	121

INTRODUCCIÓN

En la mayoría de las instituciones, empresas y organizaciones resulta primordial contar con la información adecuada para la toma de decisiones en el momento preciso. Actualmente los sistemas automatizados se han convertido en un recurso de vital importancia para el desarrollo de la nueva sociedad, ya que el uso de estas modernas tecnologías garantizan procesar y obtener una mayor variedad de información, seguridad, y poder contar con información más actualizada en espacios de tiempo relativamente muy cortos.

La Unión de Jóvenes Comunistas ha ido incorporando de manera progresiva la informática y los avances de las nuevas tecnologías de la información y las comunicaciones, a los procesos relacionados con el gran número de actividades que desarrollan, con el propósito de lograr una mayor eficiencia y eficacia en su trabajo.

Actualmente la UJC tiene dificultades para llevar el control de las informaciones y estadísticas de sus integrantes y de la propia organización; así como el óptimo manejo de datos referentes al funcionamiento y dirección de la misma. Los trabajos se realizan utilizando métodos obsoletos, como lápiz y papel o con herramientas que no son las más idóneas para la gestión de grandes volúmenes de información. Esta situación provoca que las cifras que se gestionan sean inexactas, teniendo un margen de error elevado; además que se dificulte buscar información referente a un Comité de Base o de un miembro de la organización.

La elevada movilidad durante cada año en las plazas de jefes y técnicos de las oficinas de control y documentación en los municipios, distritos y centros con derechos municipales en el país, hace que muchos compañeros carezcan de los conocimientos y la experiencia para desarrollar su trabajo con eficiencia a pesar de los esfuerzos que se realizan desde el Comité Nacional y los Comités Provinciales en la preparación de este personal.

La UJC no cuenta con las herramientas informáticas necesarias para validar la veracidad de los indicadores que permiten medir: distribución por edades, sexos, clasificadores ocupacionales, antigüedad en la organización, niveles culturales y sectores de la economía.

Otra problemática es el hecho de tener que conciliar varios controles para validar las informaciones relativas a la movilidad de los militantes, lo que hace que en muchas oportunidades debido a la premura con que se trabaja, no se actualicen al unísono algunos de estos controles, provocándose descuadres de manera constante, afectándose la veracidad de la información. Los elementos que presentan desactualización y falta de coincidencia son: cómputo de la movilidad de efectivos, plantilla de militantes por C/B, expedientes de los efectivos y listados de movimientos por categorías y por meses.

Teniendo en cuenta la situación problemática descrita anteriormente, se plantea como **problema científico**: ¿Cómo mejorar la gestión de la información relacionada con los militantes de la UJC Nacional y los miembros del universo juvenil?

Se define como **objeto de estudio**: proceso de gestión de información de la UJC Nacional; enmarcando así el **campo de acción** como la gestión de los procesos del Módulo Militante.

Para dar solución al problema científico expuesto se define como **objetivo general** de la investigación, implementar el Módulo Militante como parte del Sistema Integral para la UJC Nacional, que permita introducir mejoras en la gestión de la información de los militantes y el universo juvenil.

A partir del análisis del objetivo general se derivan los siguientes **objetivos específicos**:

- Analizar sistemas de gestión existentes con propósitos similares en el ámbito nacional e internacional.
- Identificar los procesos que requieran automatización del funcionamiento actual de la UJC Nacional y la información que se gestiona.
- Identificar los requerimientos del futuro sistema.
- Analizar y diseñar el Módulo Militante.
- Implementar el Módulo Militante como parte del Sistema Integral para la UJC Nacional.

Para alcanzar los objetivos específicos enunciados, se definen como **tareas** de la investigación:

- Revisión de bibliografías sobre el tema y elaboración del marco teórico conceptual.

- Estudio de la documentación existente sobre las tareas que realiza actualmente de forma manual la UJC Nacional.
- Identificación de los procesos de negocio.
- Investigación a profundidad de la estructura actual de la UJC a nivel nacional que permita ver las especificidades de cada estructura.
- Definición y especificación de los casos de uso.
- Estudio de las herramientas seleccionadas para obtener una solución óptima para el análisis y diseño del Módulo Militante.
- Estudio de las herramientas seleccionadas que permiten desarrollar de manera segura una aplicación informática para el Módulo Militante.

Para guiar la investigación se define como **hipótesis** que si se implementa el Módulo Militante del Sistema de Gestión de la UJC Nacional se introducirán mejoras en la gestión de la información relacionada con los militantes y los miembros del universo juvenil de la UJC en el país.

Estrategia de investigación: Investigación exploratoria.

Explorar significa incursionar en un territorio desconocido. Por lo tanto, se emprende una investigación exploratoria cuando no se conoce el tema por investigar, o cuando los conocimientos que se tienen al respecto son tan vagos e imprecisos que impiden sacar conclusiones sobre qué aspectos son relevantes y cuáles no. La estrategia utilizada en la presente investigación es la exploratoria, pues no existen investigaciones previas sobre el objeto de estudio, y por lo tanto se requiere explorar e indagar, con el fin de alcanzar el objetivo planteado. (CRUZ, 2006)

Como parte del presente trabajo se realizó un estudio de los diferentes métodos investigativos, llegando a conclusiones concretas de cuáles se utilizarán y cómo se ponen de manifiesto en la investigación.

Se pueden establecer dos grandes clases de métodos de investigación: los métodos teóricos y los empíricos. Los métodos teóricos permiten estudiar las características del objeto de investigación que no son observables directamente, facilitan la construcción de modelos e hipótesis de investigación y crean las condiciones para ir más allá de las características fenomenológicas y superficiales de la realidad,

contribuyendo al desarrollo de las teorías científicas y para su ejecución se apoyan en el proceso de análisis y síntesis.

Análisis: permite la división mental del fenómeno en sus múltiples relaciones y componentes para facilitar su estudio.

Síntesis: establece mentalmente la unión entre las partes previamente analizadas, posibilita descubrir sus características generales y las relaciones esenciales entre ellas.

Por su parte los métodos empíricos describen y explican las características fenomenológicas del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional. (ROLANDO ALFREDO HERNÁNDEZ LEÓN, 2002)

Dentro de los métodos teóricos se utilizarán el histórico, la modelación y el analítico-sintético y la entrevista como método empírico.

Histórico: Se encarga de analizar el fenómeno en toda su trayectoria histórica destacando las principales etapas de su desenvolvimiento. Se utilizará este método para encontrar las problemáticas existentes en el campo de acción, a través de un análisis de los procesos de negocio del objeto de estudio durante toda su trayectoria histórica.

Modelación: Mediante este método se crean abstracciones con el propósito de explicar la realidad. Este método se pondrá en práctica ya que dentro de los objetivos está realizar el análisis del sistema informático y para llevar a cabo este proceso se hace el análisis de la realidad mediante diversos modelos y diagramas que ayudan a comprender el objeto en su totalidad.

Analítico-Sintético: El método consiste en dividir el fenómeno a estudiar en las partes que lo integran para hacer más comprensible el estudio, esto se hace a través del análisis y la síntesis que es la operación inversa. Este método se utilizará pues se dividirá el fenómeno a estudiar en los componentes que lo integran: expediente del militante, crecimiento, los diferentes tipos de altas y bajas, entre otros, para comprender cómo funciona cada uno por separado. Como resultado de ello, se toma de cada uno las características principales para modelar un sistema que logre una integración y armonía dentro de los procesos que rigen su comportamiento.

Entrevista: Este método facilita la obtención de información para comprender mejor el problema a resolver y precisar los objetivos. Se utilizará mucho porque además de ser la principal fuente de conocimiento para entender el negocio con mayor profundidad, la cercanía geográfica con el cliente, permite realizar con mayor periodicidad los encuentros.

El documento estará estructurado en cuatro capítulos de la siguiente forma:

Capítulo 1: En este capítulo se realizará la fundamentación teórica donde se incluye el estado del arte de la investigación a nivel nacional e internacional. Se abordará y profundizará en el estudio de las herramientas y metodologías definidas para el desarrollo del sistema informático. Además, se expondrán los roles que se desarrollarán producto de las necesidades actuales del proyecto.

Capítulo 2: En este capítulo se describen los procesos del negocio del objeto de estudio. Se plantean los requisitos funcionales y no funcionales que debe tener el sistema para obtener de él resultados satisfactorios. Se describen los casos de uso tanto del negocio como del sistema y a través de diagramas se muestra cómo interactúan actores y casos de uso en el negocio y en el sistema.

Capítulo 3: En este capítulo se realizará el análisis del sistema para obtener una visión sobre qué es lo que hace el sistema, a través de diagramas que suavizarán la transición al diseño, el cual se desarrollará también en el presente capítulo, de modo que sea consistente con el entorno de implementación, permitiendo mostrar los detalles descriptivos de la solución propuesta. Se mostrarán diagramas con el objetivo de comprender la estructura interna de la aplicación a desarrollar, así como el estudio y selección de los patrones de diseño que se usarán para el desarrollo del sistema.

Capítulo 4: En este capítulo se describirá la implementación del sistema donde los elementos del diseño se convertirán en elementos de implementación y éstos en términos de componentes. Se explicará el estándar de codificación a utilizar en la implementación del mismo así como las pruebas realizadas al producto. Partiendo de los resultados del diseño se identificarán los posibles componentes en los cuales se puede implementar la aplicación y se mostrarán además los diferentes diagramas de componentes.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se realizará la fundamentación teórica donde se incluye el estado del arte de la investigación a nivel nacional e internacional. Se abordará y profundizará en el estudio de las herramientas y metodologías definidas para el desarrollo del sistema informático. Además, se expondrán los roles que se desarrollarán producto de las necesidades actuales del proyecto.

1.1 Un poco de historia de la UJC

La Unión de Jóvenes Comunistas se constituyó el 4 de abril de 1962. Esta organización selectiva surge como resultado del proceso de unidad del movimiento juvenil cubano, de la victoria definitiva del socialismo tras los combates de Playa Girón y el desarrollo hacia formas superiores de organización de la Asociación de Jóvenes Rebeldes, su predecesora.

Su objetivo es asegurar la unidad de los jóvenes cubanos, movilizarlos en torno a la Revolución socialista y contribuir a su educación. Estos principios rectores de la organización no sólo son para su membresía, sino también los extiende a toda la juventud, la cual representa y vela por sus intereses.

Durante su historia laboró en la gestación y desarrollo de las organizaciones secundaristas: la Unión de Estudiantes Secundarios (1962) y las Brigadas José Antonio Echeverría (1967).

En los años finales de la década de los 80 la UJC continuó la lucha contra los mecanismos burocráticos y formales de funcionamiento que dañaban la vida interna, así se renovaron los métodos y estilos de trabajo, se fortaleció el trabajo con los jóvenes, haciéndose énfasis en el trabajo político ideológico a partir de la historia patria.

La Unión de Jóvenes Comunistas es la organización política de la juventud cubana, que protagoniza y dirige los más de 150 Programas de la Revolución que contribuyen a la formación de las nuevas generaciones y a una mayor educación y cultura de la sociedad.

El ingreso es de carácter voluntario y selectivo. En ella militan más de 500 000 jóvenes y la proyección de su trabajo va más allá de sus miembros, está dirigida a toda la población joven del país.

La organización está estructurada a lo largo del país, y para dirigir los trabajos entre congreso y congreso cuenta con un Comité Nacional y un Buró Nacional. Hoy la juventud cubana protagoniza y lleva adelante todos los programas de la Revolución haciendo realidad el sueño de nuestro Comandante en Jefe cuando dice que un mundo mejor es posible.()

1.2 Estado actual del funcionamiento de la UJC

Los miembros de la organización deben cumplir con diferentes tareas y obligaciones, para llevar el control de estas actividades y controlar el funcionamiento de la organización. La dirección de la UJC cuenta con departamentos y cada uno de ellos cumple responsabilidades específicas:

El departamento de control y documentación se encarga en primer lugar del control de la movilidad de los efectivos de la organización, ejecutando los procesos de traslado e incorporación de los militantes en cada uno de los Comités de Base (C/B). De igual manera ejecuta las diferentes bajas de estos aprobadas en el departamento de procesos políticos, también incorpora todos los miembros del universo juvenil aprobados como nuevos militantes a partir del resultado del proceso de crecimiento.

El departamento de control y documentación se encarga además de controlar activaciones y desactivaciones de las diferentes organizaciones de base, que incluyen: Comités UJC, Comités Primarios y C/B. Se encarga además de custodiar los expedientes políticos de los militantes y por último, se encarga de controlar a los militantes pendientes a incorporar.

El departamento de procesos políticos se encarga del control y seguimiento del proceso de crecimiento a la organización.

El departamento de funcionamiento se encarga del control del funcionamiento mensual de las organizaciones de base, así como las activaciones y desactivaciones de las mismas.

El departamento de asuntos generales se encarga del funcionamiento de los Burós y Comités de la Organización a los diferentes niveles, así como del aseguramiento de cada una de las actividades que realicen estos organismos de dirección.

El departamento de cuadros se encarga de elaborar la documentación de los diferentes procesos de cuadros de la organización, la atención a los cuadros y su reserva.

1.3 La estructura de la UJC

La Unión de Jóvenes Comunistas es la organización juvenil del Partido Comunista de Cuba (PCC) y presenta una estructura nacional compuesta por 14 Comités Provinciales, 169 Comités Municipales, incluyendo el Comité Municipal de la Isla de la Juventud, así como 8 Comités Distritales, de estos últimos 4 se ubican en el municipio Camagüey y 4 en el municipio Santiago de Cuba. Adicionalmente la Organización por su complejidad cuenta con 3 Comités Especiales: El Comité Especial UJC, el Comité Especial del Exterior y el Comité Especial UCI. La UJC cuenta con Direcciones Juveniles en las Fuerzas Armadas Revolucionarias (FAR) y el Ministerio del Interior (MININT) que se subordinan a las Direcciones Políticas del PCC de ambos mandos, y que tienen distribuidos los militantes a lo largo y ancho del país.

1.4 Sistemas de Gestión

Un sistema de gestión es una estructura probada para la gestión y mejora continua de las políticas, los procedimientos y procesos de la organización. Un sistema de gestión ayuda a lograr los objetivos de la organización mediante una serie de estrategias, que incluyen la optimización de procesos, el enfoque centrado en la gestión y el pensamiento disciplinado.

1.4.1 Necesidad de los sistemas de gestión

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

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

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

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

- Mejorar la efectividad operativa.
- Reducir costos.
- Aumentar la satisfacción de clientes y partes interesadas.
- Lograr mejoras continuas.
- Potenciar la innovación.
- Aportar claridad.

El uso de un sistema de gestión probado permite renovar constantemente el objetivo, las estrategias, las operaciones y niveles de servicio. (INSTITUTION, 2009)

1.5 Sistemas de información

Los sistemas de información están concebidos para capturar, actualizar, integrar, consultar y analizar información pertinente a la organización a la cual pertenecen.

Un sistema de información es la composición de elementos que operan unidos en orden de capturar, procesar, almacenar y distribuir información. Esta información se utiliza generalmente para tomar decisiones, la coordinación, el control y el análisis en una organización. Frecuentemente, el propósito básico del sistema es la gestión de la información.

Los sistemas de información:

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

1.6 Sistemas automatizados existentes vinculados al campo de acción

Como la UJC es una organización que sólo radica en Cuba, no existen precedentes de sistemas automatizados a nivel internacional relacionados con ella. A nivel nacional los sistemas que existen que pudieran constituir antecedentes de esta investigación, son sitios Web en varios lugares, principalmente en las universidades y joven club que se dedicaban solamente a informar y tener noticias actualizadas del funcionamiento de la organización como por ejemplo el portal provincial de la UJC en Matanzas y el sitio de la UJC de la UCI. Sin embargo se estudiaron algunos sistemas con propósitos similares de los cuales se realiza un análisis a continuación.

Del ámbito internacional se destaca el Sistema de Gestión Escolar Módulo Registro Académico del Ministerio de Educación del Gobierno de El Salvador publicado en San Salvador en noviembre del 2003. El sistema de Registro Académico fue creado con la finalidad de permitir el registro, procesamiento y recuperación de datos de los estudiantes que se manejan en cada Centro Educativo del país, con el propósito de mecanizar el procesamiento de la matrícula, notas y emisión de reportes de una manera efectiva y eficiente.

En este módulo se podrá ingresar la solicitud de matrícula, información general del alumno para actualizar la ficha, asignación al grado correspondiente, ingreso de las notas en el transcurso del año cursado de cada una de las actividades, además, el sistema podrá llevar un historial académico por alumno, es decir, que a través del Código Único de Alumno se podrá saber todo acerca de un estudiante específico.

En Cuba, específicamente en Ciego de Ávila se realizó un sistema automatizado para el control de los expedientes de tutoría en la carrera de Psicología en la Sede Universitaria de Florencia. El objetivo de la investigación era elaborar una herramienta Web que contribuyera a la correcta actualización de los expedientes de tutoría. (MSC. NIRIA CASTILLO ARZOLA, 2007)

En la facultad de informática de la universidad de Cienfuegos se realizó un sistema de gestión de información para la empresa procesadora de café Eladio Machín de Cumanayagua, como alternativa de solución a las deficiencias en la entrega de información vinculada a la producción en dicha entidad. Este sistema llamado SGIEPC, da la posibilidad de almacenar datos relacionados con el proceso productivo de la empresa en cuestión. El sistema también establece diferentes tipos de usuarios de acuerdo a las

políticas de seguridad de la empresa. Los informes son confeccionados en diferentes intervalos de tiempo, los cuales fija el usuario que hace la petición, posibilitando también imprimir dichos reportes. Tiene asociada una base de datos que organiza y almacena eficientemente la información necesaria, y una ayuda para el trabajo con la aplicación.(LLANES, 2008)

En la Universidad de las Ciencias Informáticas se desarrolló un sistema de gestión para la Federación Estudiantil Universitaria (FEU) con características muy similares a las que se desean para la UJC en el país. El sistema fue desarrollado por el proyecto Kainos.

El pasado curso se presentó en la UCI un trabajo de diploma en el que se desarrolló el análisis y diseño de un sistema para la UJC del propio centro estudiantil, con el propósito que sirviera de base para la implementación del mismo. Este sistema se puso en marcha el presente curso, aplicación que se desarrolla de forma paralela con el presente trabajo y a pesar de sus similitudes tienen características diferentes, por ejemplo las estructuras que se gestionan no son las mismas y la aplicación a la cual se hace alusión es una aplicación Web.

Después de conocer las funcionalidades fundamentales de los sistemas anteriores se decide realizar un sistema que automatice la gestión de los eventos de la UJC a nivel nacional. Se realizará una aplicación de escritorio que permita mejorar la gestión de la información relacionada con los militantes y los miembros del universo juvenil de la UJC en el país de manera segura y logre el control de la movilidad de los efectivos de la organización, así como el desarrollo de los procesos políticos que llevan a cabo. Este sería el primer sistema de gestión de información de la UJC en el país.

Las aplicaciones desktop o de escritorio están diseñadas para cubrir las necesidades del negocio y administrar la información interna del mismo. Con las aplicaciones de escritorio se puede contar con altos niveles de rendimiento, velocidad de respuesta y seguridad. Aunque actualmente existe un gran auge por las aplicaciones Web, muchas empresas y negocios siguen manteniendo este tipo de aplicaciones porque al estar instaladas en cada computadora aprovechan recursos del sistema operativo, adicionalmente, la información contenida en ellas no está expuesta a personas no autorizadas a través de Internet.

Se decide desarrollar una aplicación de escritorio teniendo en cuenta que en las direcciones de la UJC en todo el país no existen requerimientos de hardware suficientes para instalar un servidor de base de datos

y aplicación, necesario para el desarrollo de una aplicación web. Otra razón por la que se desarrollará una aplicación de escritorio se debe a que en la Juventud Nacional no hay un sistema de redes, y la aplicación se va a instalar en cada ordenador que lo necesite a lo largo del país. Teniendo en cuenta que en las direcciones municipales de la UJC todo el trabajo se realiza en un único ordenador, los distintos funcionarios que operarían con la aplicación lo harían en la misma máquina, por lo que no existe la dificultad de actualización de datos, además cada uno de los funcionarios de la organización tiene derecho a modificar sólo la información a la que tiene acceso, o sea, el funcionario de procesos políticos no puede modificar la información que maneja el funcionario de documentación y control y viceversa.

1.7 Fundamentación de las tecnologías y herramientas a utilizar

Luego de revisar las necesidades y características de la UJC Nacional, organización para la cual se desarrolla la aplicación, se realizará una aplicación de escritorio usando para su desarrollo *Java* como lenguaje de programación y como IDE (*Integrated Development Environment*, Entorno de Desarrollo Integrado) NetBeans, basado en una arquitectura en tres capas. La metodología empleada en la elaboración del sistema es el Proceso Unificado de Desarrollo (RUP), y como lenguaje de modelado UML, la herramienta case a usar es Visual Paradigm, DB4O como Sistema Gestor de Base de Datos y Subversión para el control de versiones.

1.7.1 Metodología de desarrollo de software

Rational Unified Process (RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. (JUAN PABLO GOMEZ GALLEGO, 2007)

RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización, utiliza UML para definir los modelos de software y se divide en 4 fases: Inicio, Elaboración, Construcción y Transición.

- Inicio: Tiene como objetivo determinar la visión del proyecto.
- Elaboración: En esta etapa el objetivo es determinar la arquitectura óptima.

- Construcción: El objetivo es obtener la capacidad operacional inicial.
- Transición: Obtener el release del producto.

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales, los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como flujos de apoyo. Tiene tres características esenciales: dirigido por casos de uso, centrado en la arquitectura e iterativo e incremental.

1.7.2 Herramienta CASE

Las Herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores. (LANCER, 2004)

Visual Paradigm es una poderosa herramienta para visualizar y diseñar elementos de software, para ello utiliza UML y ofrece una gama de facilidades para el modelado de aplicaciones. Está orientada a la creación de diseños usando el paradigma de programación orientada a objetos. (INTERNATIONAL, 2005)

Visual Paradigm es una herramienta que provee soporte para la generación de código, tiene integración con diversas IDE's como NetBeans (de Sun Microsystems), JDeveloper (de Oracle), Eclipse (de IBM), JBuilder (de Borland), así como la posibilidad de realizarse la ingeniería inversa para aplicaciones realizadas en JAVA, .NET, XML (*Extensible Markup Language*, Lenguaje de marcas extendido) e Hibernate¹.

Tiene dentro de sus características que es portable y posee gran facilidad de uso. Utiliza UML como lenguaje de modelado. Su diseño se centra en casos de uso y se enfoca al negocio que genera un software de mayor calidad. También tiene disponibilidad en múltiples plataformas y usa un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.

¹- Herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos XML que permiten establecer estas relaciones.

Se decide utilizar Visual Paradigm porque es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

1.7.3 Lenguaje de modelado: UML

El Lenguaje Unificado de Modelado (UML, por su nombre en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. (CHOVA, 2007)

UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Proporciona una organización en el proceso de diseño de forma tal que los analistas, clientes, desarrolladores y otras personas involucradas en el desarrollo del sistema lo comprendan.

Dentro de sus características se pueden mencionar algunas tales como:

- Tecnología orientada a objetos.
- Viabilidad en la corrección de errores.
- Desarrollo iterativo e incremental.
- Participación del cliente en todas las etapas del proyecto.
- Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones).
- Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.

- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.
- Permite realizar una verificación y validación del modelo realizado.

Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa. Esto permite que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño, de más alto nivel, de la estructura de un proyecto.

Es importante resaltar que UML es un lenguaje para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, detallar los artefactos en el sistema, documentar y construir. Es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como RUP), pero no especifica en sí mismo qué metodología o proceso usar.

1.7.4 Lenguaje de programación

La implementación será en el lenguaje Java dadas sus características y potencialidades. Java está totalmente basado en clases y objetos. Todo en Java (aparte de algunos tipos primitivos) es un objeto. Contrariamente a lenguajes híbridos como C++ o Visual Basic, en Java no se permite programar fuera de un objeto o clase. No hay módulos ni funciones globales.

El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros. Tiene una arquitectura neutral, portátil y robusta. Posee las estructuras mínimas de un lenguaje de programación tradicional, sin añadir ninguna estructura más.

Una característica todavía más distintiva de Java es su capacidad multiplataforma. Lenguajes como C o COBOL se han implementado en múltiples plataformas, pero siempre han necesitado recompilaciones o adaptaciones al pasar de una a otra. En cambio, Java desde el principio ha sido pensado para adaptarse a varios entornos. Esto lo consigue no sólo a nivel de código fuente, sino también a nivel de código compilado. El programa que se escribe se puede compilar en Windows o en Linux, y funciona. Pero,

además, hasta el programa compilado puede ejecutarse sin más preparación, en distintas máquinas. Eso lo consigue porque Java se compila y ejecuta, no en un procesador o entorno en particular, sino en una "máquina virtual". ²El programa Java podrá ejecutarse en cualquier sistema operativo que tenga una máquina virtual Java compatible.

Tiene otras características como por ejemplo:

- Simple.
- Orientado a objetos.
- Distribuido.
- Robusto.
- Arquitectura neutral.
- Seguro.
- Portable.
- Interpretado.
- Dinámico.

1.7.5 Entorno de desarrollo: NetBeans

Un IDE es un programa compuesto por un conjunto de herramientas para un programador.

Los componentes que deben tener los IDEs son:

- Un editor de texto.
- Un compilador³.
- Un intérprete⁴.

²- Es un software que emula a un ordenador y puede ejecutar programas como si fuese un ordenador real.

³ - Un compilador es un programa que permite traducir el código fuente de un programa en lenguaje de alto nivel, a otro lenguaje de nivel inferior (típicamente lenguaje máquina).

- Herramientas de automatización.
- Un depurador.⁵
- Posibilidad de ofrecer un sistema de control de versiones.
- Factibilidad para ayudar en la construcción de interfaces gráficas de usuarios.

NetBeans es un Entorno Integrado de Desarrollo gratuito, de código abierto para desarrolladores de software. Puede obtener todas las herramientas que necesite para crear aplicaciones profesionales para el escritorio, la empresa, la web y equipos móviles con el lenguaje Java, C/C++, y Ruby. NetBeans IDE es fácil de instalar y de uso instantáneo y se ejecuta en varias plataformas incluyendo Windows, Linux y Mac OS X y Solaris. (NETBEANS, 2009)

La versión 6.0 incluye significativas mejoras y nuevas características, incluyendo una reestructuración completa de la infraestructura del editor, soporte para lenguajes adicionales, nuevas características productivas, y un proceso simplificado de instalación que permite instalar y configurar fácilmente el IDE según las necesidades del usuario. (GADGETCABARET, 2007)

Otras características que esta versión incluye son mejor edición del código, capacidades de navegación e inspección, historia local, soporte integrado para Subversión, y mayores capacidades de personalización integradas en la distribución estándar.

1.7.6 Sistema de Base de Datos: db4o

Se decide usar el sistema gestor de base de datos DB4O para el almacenamiento de la información, debido a que en los centros donde será desplegado el sistema no se cuenta con los recursos de hardware necesarios para dedicar a guardar los datos y DB4O permite almacenar datos de manera local.

⁴ - Es un programa informático capaz de analizar y ejecutar otros programas, escritos en un lenguaje de alto nivel. Los intérpretes se diferencian de los compiladores en que mientras estos traducen un programa desde su descripción en un lenguaje de programación al código máquina del sistema destino, los primeros (los intérpretes) sólo realizan la traducción a medida que sea necesario, típicamente, instrucción por instrucción, y normalmente no guardan el resultado de dicha traducción.

⁵ - Es un programa que permite depurar o limpiar los errores de otro programa informático.

DB4O es un novedoso motor de base de datos orientado a objetos. Sus siglas se corresponden con la expresión "DataBase (for) Objects", que a su vez es el nombre de la compañía que lo desarrolla. Es una base de datos orientada a objetos completamente nativa de alto rendimiento, compacta y válida como base de datos embebida en la aplicación. El término: "Embebido", es sinónimo de "empotrado" o "incrustado". En informática dicho término suele verse como el hecho de que de lo que se habla está incluido directamente en el producto.

DataBase (for) Objects:

- Soporta aplicaciones Cliente/Servidor (Aplicaciones distribuidas).
- Disponible para entornos Java o .Net.
- Mínimo consumo de recursos.
- Alto rendimiento.
- Fácil Implementación (Mínimo código).
- Portabilidad.
- Confiabilidad.
- Transparencia.
- Administración Nula.
- Código Abierto (Bajo 2 licencias).

Con DB4O se elimina el proceso de diseño, implementación y mantenimiento de la base de datos pues:

- El modelo de clases es el esquema de base de datos.
- El motor de la base de datos consiste tan solo en un pequeño archivo .jar ⁶ de menos de 400kb.
- La instalación supone agregar el motor de la base de datos (db4o-.jar) a nuestro CLASSPATH.⁷

⁶- Es simplemente un archivo comprimido que incluye una estructura de directorios con Clases. Los archivos jar permiten introducir todos los archivos necesarios para ejecutar nuestra aplicación (clases, imágenes, sonidos, archivos de texto) en uno solo con la posibilidad de comprimirlos.

- La documentación de la base de datos está creada mediante JavaDoc⁸ y viene incluida en el mismo paquete.

1.7.7 Sistema de control de versiones: Subversión

Un sistema de control de versiones es un sistema de gestión de archivos y directorios, cuya principal característica es mantener la historia de los cambios y modificaciones que se han realizado sobre ellos a lo largo del tiempo. De esta forma, el sistema es capaz de “recordar” las versiones antiguas de los datos, lo que permite examinar el historial de cambios o recuperar versiones anteriores de un fichero, incluso aunque haya sido borrado.(GRANADO, 2004)

Para garantizar un correcto control de versiones y gestión de cambio se decidió utilizar el cliente *TortoiseSVN*, que es un cliente gratuito de código abierto, que maneja ficheros y directorios a lo largo del tiempo y algunas de sus características son:

- Mantiene versiones no sólo de archivos, sino también de directorios.
- Atomicidad de las actualizaciones: una lista de cambios constituye una única transacción o actualización del repositorio. Esta característica minimiza el riesgo de que aparezcan inconsistencias entre distintas partes del repositorio.
- Soporte tanto de ficheros de texto como de binarios.
- Un buen uso del ancho de banda, ya que en las transacciones se transmiten sólo las diferencias y no los archivos completos.

⁷- Una opción admitida en la línea de órdenes o mediante variable de entorno que indica a la Máquina Virtual de Java dónde buscar paquetes y clases definidas por el usuario a la hora de ejecutar programas.

⁸- Es una utilidad de Sun Microsystems para la generación de documentación de APIs (es el conjunto de funciones y procedimientos (o métodos, si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.) en formato HTML a partir de código fuente Java. Javadoc es el estándar de la industria para documentar clases de Java.

1.8 Roles y Artefactos

Un rol define el comportamiento y responsabilidades de un individuo, es una definición abstracta de un conjunto de actividades realizadas y de artefactos obtenidos. Los roles son realizados típicamente por un individuo, o un conjunto de individuos, trabajando juntos en equipo. Un miembro del equipo de proyecto cumple normalmente muchos roles, así como un mismo rol puede ser representado por varias personas. Los roles no son individuos; sino que, describen cómo los individuos se comportan en el negocio y qué responsabilidades tienen.

RUP define grupos de roles, agrupados por participación en actividades relacionadas. Estos grupos son: ***Analistas, Desarrolladores, Gestores, Apoyo, Especialista en Prueba y Otros roles.***

En el desarrollo del presente trabajo de diploma de acuerdo a las necesidades actuales del proyecto se desarrollarán los roles de analista, diseñador e implementador pertenecientes al grupo de desarrolladores definido por RUP.

1.8.1 Rol Analista

Agrupar los roles que están involucrados fundamentalmente en la captura y gestión de los requisitos del sistema, que pueden estar representados por una o varias personas entre los que se encuentran: Analista de Procesos del Negocio, Diseñador del Negocio, Analista del Sistema y Especificador de Requerimientos.

Los artefactos que se obtendrán en el presente trabajo son:

- **Modelo de casos de uso del negocio:** Es un modelo que describe los procesos de un negocio y su interacción con elementos externos.
- **Modelo de casos de uso:** Es un modelo del sistema que contiene actores, casos de uso y sus relaciones.
- **Especificación de requerimientos del software:** Es la captura de los requerimientos del software para el sistema o una parte de éste.

- **Glosario:** Es un documento que define los términos comunes que se utilizan para describir el proyecto.

1.8.2 Rol Diseñador

En la metodología RUP el rol diseñador es el responsable de diseñar una parte del sistema cumpliendo con las restricciones de los requerimientos, arquitectura y proceso de desarrollo del proyecto. Identifica y define las responsabilidades, operaciones, atributos y relaciones de los elementos de diseño.

Debe asegurarse que el diseño es consistente con la arquitectura del software y que está detallado al punto que se puede proceder con la implementación.

Los artefactos que se obtendrán en el presente trabajo son:

- **Clases del diseño:** Una clase es una descripción de un conjunto de objetos que comparten las mismas responsabilidades, relaciones, operaciones, atributos, y la semántica.

1.8.3 Rol Implementador

El rol Implementador es responsable de desarrollar y de probar componentes de acuerdo con los estándares adoptados del proyecto para la integración en subsistemas más grandes. Cuando los componentes de prueba, tales como drivers o partes se deben crear para apoyar la prueba, el implementador es también responsable de desarrollar y de probar los componentes de prueba y los subsistemas correspondientes.

Los artefactos que se obtendrán en el presente trabajo son:

- **Elementos de implementación:** Los elementos de implementación son la parte física de la implementación, incluyen los archivos, directorios, ficheros de códigos (fuentes, binarios o ejecutables), ficheros de datos y de documentación como ficheros de ayuda online.
- **Artefactos de instalación:** Se refiere al software y las instrucciones documentadas requeridas para instalar el producto.

1.9 Conclusiones

Tras el análisis de las condiciones y los problemas que caracterizan el objeto de estudio, entre los que se encuentran el mal manejo de datos referente al funcionamiento y dirección de la UJC, duplicidad de la información y difícil búsqueda de datos referente a un comité de base o un miembro de la organización, se propone desarrollar una aplicación de escritorio que permita mejorar la gestión de la información relacionada con los militantes y los miembros del universo juvenil de la UJC en el país de manera segura y logre el control de la movilidad de los efectivos de la organización, así como el desarrollo de los procesos políticos que llevan a cabo. En este capítulo también se analizaron sistemas existentes vinculados al campo de acción pero ninguno cumplió con las especificidades de la UJC Nacional. La propuesta de solución disminuiría los riesgos del trabajo manual y agilizaría los diferentes procesos que se llevan a cabo en la organización. Una vez decidido el desarrollo de una aplicación informática como alternativa de solución de las dificultades existentes en la UJC Nacional, se demuestran las potencialidades de las herramientas con las cuales se desarrollará dicha aplicación.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En este capítulo se describen los procesos del negocio del objeto de estudio. Se plantean los requisitos funcionales y no funcionales que debe tener el sistema para obtener de él resultados satisfactorios. Se describen los casos de uso tanto del negocio como del sistema y a través de diagramas se muestra cómo interactúan actores y casos de uso en el negocio y en el sistema.

2.1 Modelo del negocio

El Proceso Unificado de Desarrollo propone un conjunto de artefactos entre los que se encuentran los Casos de Uso del Negocio. RUP propone varias fases y flujos de trabajos, dentro de ellos se encuentra el modelamiento del negocio, que brinda una visión general del alcance del proyecto y tiene mayor importancia durante la fase de inicio. En el presente trabajo se definen los actores, trabajadores y casos de usos de este flujo de trabajo.

2.1.1 Objetivos del modelado del negocio

La creación de modelos que organicen y presenten los detalles importantes de problemas reales que se vinculan con el sistema informático a construir es de gran utilidad para obtener buenos resultados. El modelado del negocio tiene como objetivos:

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar el sistema.
- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización.
- Derivar los requerimientos del sistema que va a soportar la organización.

2.1.2 Análisis de los procesos de negocio

La UJC Nacional registra toda la información relacionada con los militantes existentes en el país, para ello existe un Expediente que es único por militante en el que se recogen todos los datos de la trayectoria del mismo. Además de esto en cada uno de los comités de base existe una ficha en la que se recogen los datos específicos de cada uno de los militantes que han pertenecido a dicho comité. Todo este trabajo es manejado por cada uno de los departamentos que conforman esta organización. Como parte de la información que se almacena de los militantes está el control de su movilidad y el proceso de ingreso a la organización. Además de los datos de los militantes se almacena la información de cada miembro del universo juvenil y todo su proceso de crecimiento hasta el momento en que se incorpora a la organización (Alta por crecimiento), en este momento esta persona inicia su vida como militante y por tanto también se guarda un registro a partir de ese momento como se había explicado anteriormente. Todos estos procesos son muy engorrosos y por eso se decide automatizar gran parte de ellos, a continuación se exponen los procesos que serán objeto de automatización.

2.1.3 Objeto de automatización

Los procesos que serán objeto de automatización son los siguientes:

- Gestión de los diferentes usuarios que tendrán acceso al sistema.

El proceso comienza cuando el administrador del sistema decide agregar un nuevo usuario para el trabajo con la aplicación o cuando es necesario modificar o eliminar los datos de uno ya existente. Luego de haberse creado un usuario y una vez autenticado el mismo, el sistema solo mostrará las acciones a las que tiene acceso de acuerdo con el rol que le fue asignado.

- Gestión de los diferentes comités.

El proceso inicia cuando el Miembro del Buró UJC decide crear un nuevo comité o modificar uno ya existente. Los comités nunca se eliminan, por reglas del negocio cuando en ellos no exista la cantidad de militantes necesarios se desactivan, guardándose un registro de los diferentes periodos de desactivación. Un comité siempre es de un tipo de una estructura. Las diferentes estructuras son: Comité Nacional,

Comité Provincial, Comité Municipal, Comité Distrital, Comité UJC, Comité Primario, Comité de Base y Comité de Base Independientes.

- Gestión de centros.

El proceso comienza cuando el Miembro del Buró UJC decide agregar un nuevo centro o modificar los datos de uno ya existente.

- Gestión de los datos de cada miembro del universo juvenil.

El proceso comienza cuando una persona informa que no es militante, esta persona comienza a formar parte del universo juvenil del comité de base en el que radica y a partir de ese momento se comienza la gestión de sus datos.

- Realizar crecimiento del universo juvenil.

El proceso comienza cuando la persona acepta ser militante. Se selecciona un dúo de crecimiento que será el que realice las investigaciones de la persona a crecer. El resultado del proceso de la investigación se analizará para decidir si se va a crecer o no a la persona. Si en algún momento la persona se acoge al principio de voluntariedad se termina el proceso de crecimiento.

- Crear expediente del militante.

El proceso comienza cuando un miembro del universo juvenil es dado de alta como militante o cuando un militante es dado de alta por reincorporación a las filas de la UJC.

- Gestionar expediente del militante.

El proceso comienza cuando es necesario modificar o eliminar el expediente del militante.

- Dar Alta por Crecimiento.

El proceso comienza cuando a un miembro del universo juvenil se le aprueba el proceso de crecimiento. Una vez aprobado este proceso se procede a ingresarle los datos como militante en la planilla de militantes (Planilla DM) y el miembro del universo juvenil inicia su vida como militante.

- Dar Alta por Traslado Externo.

El proceso comienza cuando un militante arriba a un centro, se considera un traslado externo cuando la incorporación se hace desde el MININT, las FAR, el exterior, una provincia o un municipio. El militante debe presentarse al departamento de Documentación y Control con su expediente y su boleta de traslado.

- Dar Alta por Ajuste.

El proceso comienza cuando se detectan errores de cómputo al controlar los militantes, cuando se incorpora a un militante del PCC que ostentaba la doble militancia y que por interés del Partido debe continuar con esa condición para fortalecer su C/B o cuando un militante es trasladado para un C/B y no llega a incorporarse en el mismo, regresando al C/B que lo trasladó.

- Dar Baja por Ajuste.

El proceso comienza cuando se detectan errores de cómputo al controlar los militantes, cuando el militante que presentaba doble militancia cumple con la tarea asignada por parte del PCC y deja de pertenecer a las filas de la UJC o cuando el militante no regresa al antiguo CB y se da de alta en un nuevo comité de base.

- Dar Baja Natural.

El proceso comienza cuando el militante presenta motivo de baja, se determina que el militante que llegó a la edad límite tiene buena conducta dentro de la UJC pero presenta problemas de salud o problemas familiares que le impiden ingresar a las filas del PCC.

- Dar Baja por Defunción.

El proceso comienza con el fallecimiento del militante.

- Dar Baja por Traslado Externo.

El proceso comienza cuando el militante informa que se va de la institución. Se da de baja al militante entregándole el expediente y la boleta de traslado. Los tipos de traslados son los mismos que los del alta por traslado externo.

- Realizar Traslado Interno.

El proceso comienza cuando un militante solicita ser trasladado de comité de base dentro del mismo centro. En este proceso se realizan dos acciones, primeramente se da baja al militante del comité de base en el cual radica (Baja por Traslado Interno) y acto seguido se incorpora en el nuevo comité de base (Alta por Traslado Interno).

2.2 Reglas del negocio a considerar

- Los nombres de los Comités Primarios dentro de un Comité UJC no pueden ser iguales.
- Los nombres de los Comités de Base de un Comité Primario o de un municipio no deben ser iguales.
- Para que un Comité de Base pueda ser activado debe tener 3 militantes o más.
- Si en un centro hay más de un Comité de Base Independiente hay que crear un Comité UJC.
- Para eliminar un Comité UJC debe quedar un solo C/B.
- En la lista de las provincias hay que incorporar a la UCI y el Comité Especial de la UJC, la Isla de la Juventud y el Exterior. Debe tenerse en cuenta a las FAR, el MININT y el exterior como otras categorías para los movimientos.
- Los movimientos hacia otra provincia se realizarían hacia uno de los municipios de las distintas provincias del país. Se consideran provincias: El municipio Especial Isla de la Juventud, el Comité Especial UJC y la UCI.
- El expediente de un militante debe ser eliminado al cabo de los 5 años siempre que su último movimiento sea una baja
- El militante puede acogerse al principio de voluntariedad una vez iniciado el proceso de Crecimiento en cualquier paso excepto una vez que tenga fecha de ingreso a la organización.

2.3 Actores del negocio

Un actor del negocio es cualquier individuo, grupo, organización o máquina que interactúa con el negocio. Un actor siempre permanece fuera de las fronteras del negocio. El término actor significa el rol que algo o

alguien juega cuando interactúa con el negocio, por tanto no representa un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en el negocio.

Actor	Descripción
Persona	Es la persona que llega a la institución que no es militante y que en ese momento tampoco forma parte del universo juvenil; es la misma persona a la que se le realiza el proceso de crecimiento una vez que forme parte del universo juvenil.
Militante	Es el que inicia todos los procesos que tienen que ver con el control del Estado de la Militancia y el control de Vida Interna.
Miembro de la Dirección UJC	Es el miembro de la Dirección de la UJC encargado de informar el inicio de procesos orientados por la dirección de la organización. Los procesos que inicia son las altas y bajas por ajustes.

Tabla 2.1: Actores del Negocio

2.4 Trabajadores del negocio

Un trabajador del negocio representa a personas o sistemas dentro del negocio que son las que realizan las actividades que están comprendidas dentro de un caso de uso. Los trabajadores están dentro de las fronteras del negocio, son los que en un futuro se convertirán en actores del sistema que se quiere construir.

Actor	Descripción
Funcionario de Control y Documentación	Interviene en la gestión del expediente de un militante y en la gestión de los movimientos de los mismos.
Funcionario de Procesos Políticos	Interviene en la gestión de los datos del universo juvenil y en el proceso de crecimiento a la UJC.
Secretario C/B	Interviene en todos los procesos del militante en el C/B.
Miembro del C/P	Interviene en todos los procesos de la militancia para

	tener un control del estado de la misma. Es el intermediario entre el Secretario C/B y el funcionario de control y documentación.
Comisión UJC	Interviene en el proceso de dar alta por crecimiento.
Dúo de Crecimiento	Interviene en el proceso de realizarle crecimiento a un universo juvenil.
Comisión de Crecimiento	Interviene en el proceso de crecimiento.

Tabla 2.2: Trabajadores del Negocio

2.5 Diagramas de casos de uso del negocio

Un caso de uso del negocio (CUN) representa un proceso dentro del negocio, por lo que se corresponde con una secuencia de acciones con un orden lógico y que producen un resultado observable para ciertos actores del negocio. Los diagramas de casos de uso son de estructura estática y representan las relaciones entre los actores y casos de uso.

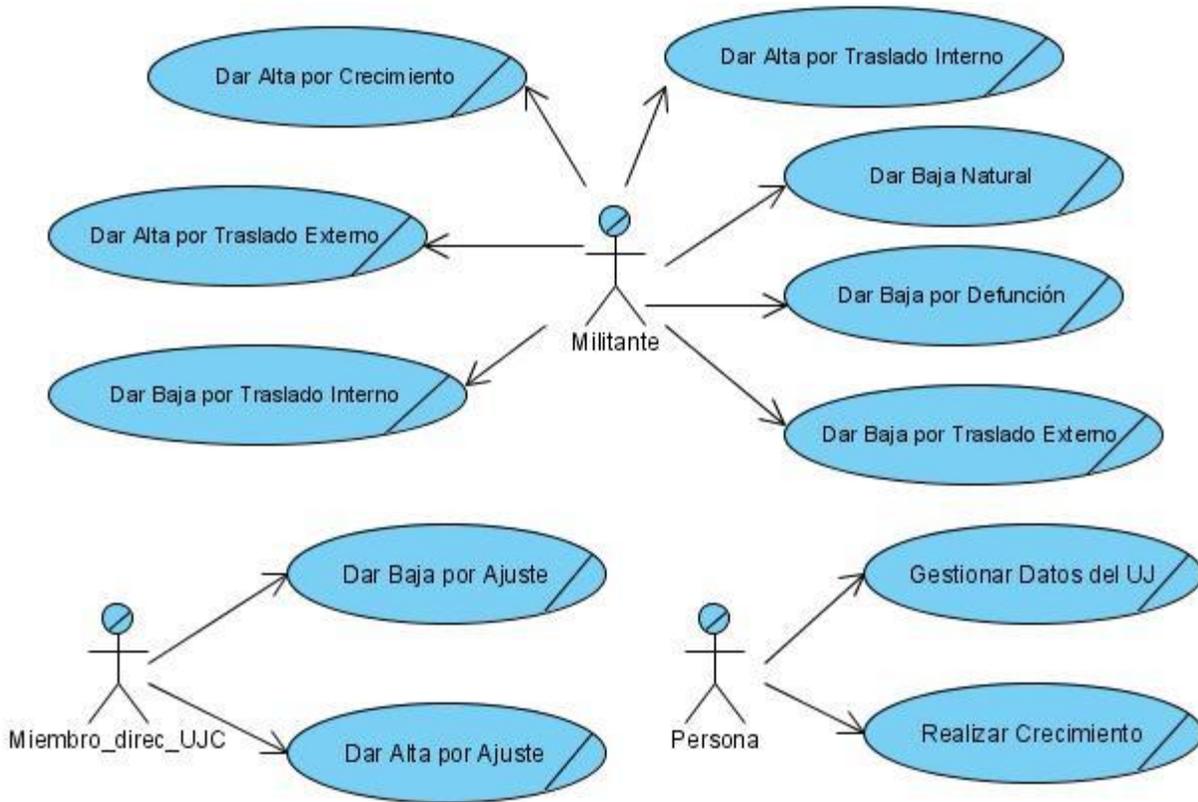


Figura 1: Diagrama de CUN

2.6 Descripción de los CUN y sus correspondientes diagramas de actividades

Por la cantidad de procesos que se automatizarán se mostrará en el documento un caso de uso, el cual servirá de guía para mostrar los principales artefactos que se irán obteniendo en el desarrollo de la investigación. Para ver las restantes descripciones de los CUN y sus correspondientes diagramas de actividades ver Anexo 2.

2.6.1 Caso de uso Gestionar UJ:

Caso de uso del negocio	Gestionar UJ
Actor	Persona
Resumen	El caso de uso inicia cuando una persona informa que no es militante, esta persona comienza a formar parte del universo juvenil del comité de base en el que radica y a partir de ese momento se comienza la gestión de sus datos, finalizando así el caso de uso.
Acción del Actor	Respuesta del proceso de negocio
1. La persona solicita ingresar al UJ	2. El funcionario de procesos políticos solicita los datos de la persona.
3. La persona informa sus datos al funcionario de procesos políticos.	4. El funcionario de procesos políticos adiciona los datos de la persona en la planilla UJ
	5. El funcionario de procesos políticos informa a la persona que sus datos fueron adicionados
6. La persona revisa los datos e informa al funcionario de procesos políticos que están correctos.	7. El funcionario de procesos políticos archiva la planilla UJ.
	8. Se termina el caso de uso.
Flujo Alterno 6.	
6. La persona revisa los datos e informa al funcionario de procesos políticos que están incorrectos.	6.1. El funcionario de procesos políticos actualiza los datos. Se pasa a la acción 7

Tabla 2.3: Descripción del CUN Gestionar UJ

2.6.2 Diagrama de actividades del caso de uso Gestionar UJ:

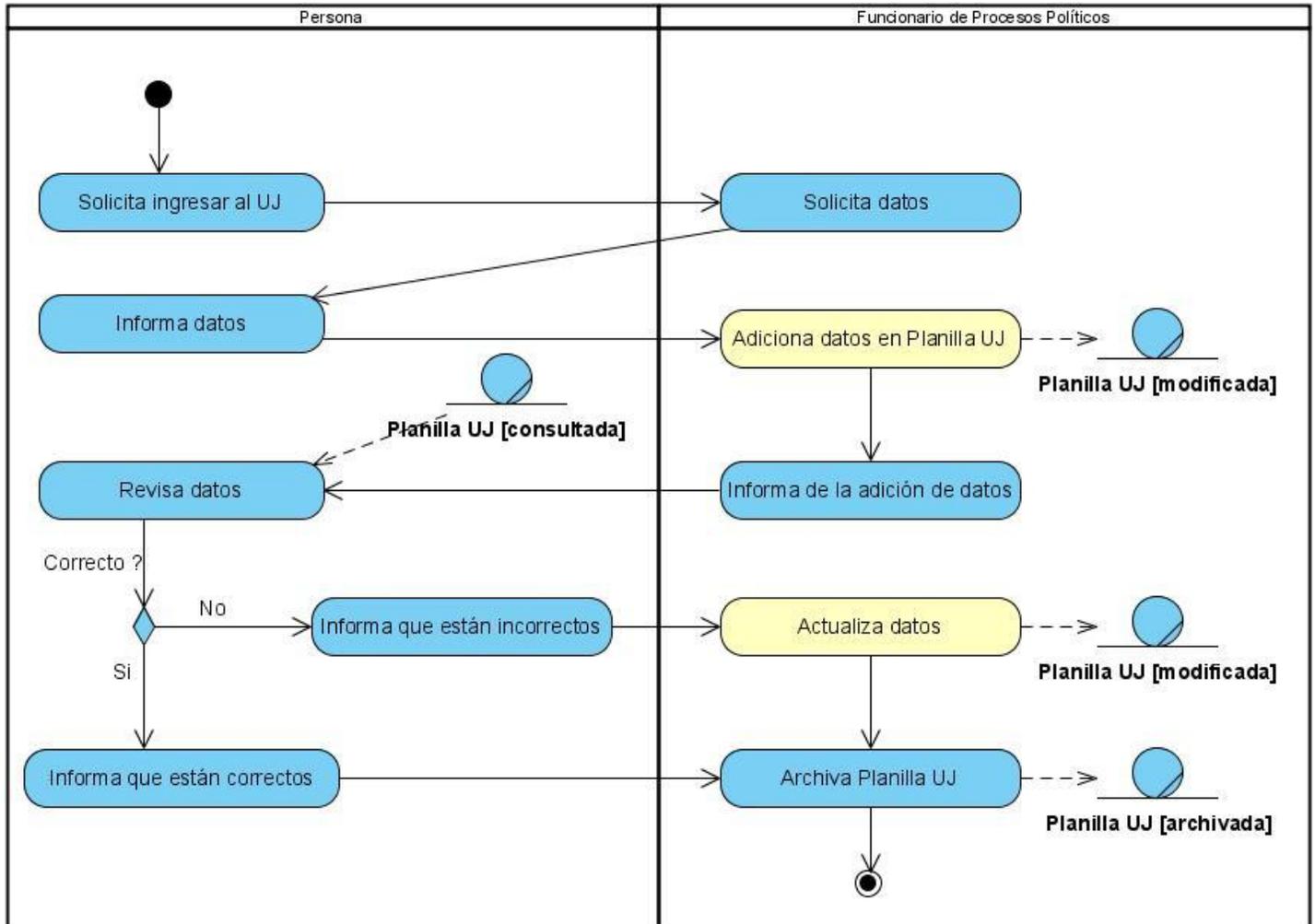


Figura 2: Diagrama de Actividades del Caso de uso Gestionar UJ

2.7 Levantamiento de requisitos

El flujo de trabajo Levantamiento de requisitos se realiza en la fase de inicio, en él se define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen. La IEEE ⁹(*Institute of Electrical and Electronics Engineers*, Instituto de Ingenieros Eléctricos y

⁹ - Organización técnica y profesional de ámbito mundial dedicada a dar soporte a la teoría y aplicaciones de la Ingeniería Eléctrica, Electrónica y de la Informática, dando cabida a ingenieros, informáticos, físicos y matemáticos. El IEEE se dedica a

Electrónicos) define un requerimiento como condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a requisitos. Los requisitos se clasifican en funcionales y no funcionales, estos últimos tienen varias clasificaciones, más adelante se detalla esta información.

2.7.1 Objetivos del levantamiento de requisitos

- Definir el ámbito del sistema.
- Definir una interfaz de usuarios para el sistema, enfocada a las necesidades y metas del usuario.
- Establecer y mantener un acuerdo entre clientes y otros involucrados sobre lo que el sistema debería hacer.
- Proveer a los desarrolladores un mejor entendimiento de los requerimientos del sistema.
- Proveer una base para estimar recursos y tiempo de desarrollo del sistema.
- Proveer una base para la planeación de los contenidos técnicos de las iteraciones.

2.8 Requerimientos funcionales (listado de las características básicas del sistema)

RF1 Crear un usuario.

1.1 Datos a insertar:

1.1.1 Nombre de usuario.

1.1.2 Contraseña.

1.1.3 Nombre completo.

1.1.4 Activado (si o no).

1.1.5 Tipo de Rol.

1.1.5.1 Funcionario de procesos políticos (tiene acceso a la gestión del universo juvenil y el crecimiento UJC).

promover el avance de la teoría y las aplicaciones de la electrotecnología. A través de sus miembros, el Instituto es una autoridad de primera línea en áreas que van desde la ingeniería de computadoras, la tecnología biomédica y las telecomunicaciones, hasta la potencia eléctrica, el aeroespacio y la electrónica de consumidores.

1.1.5.2 Funcionario de documentación y control (tiene acceso a realizar altas, bajas y gestión del expediente).

1.1.5.3 Miembro del Buró UJC (tiene acceso a la gestión de las estructuras y los centros).

1.1.5.4 Administrador (tiene acceso a la gestión de los usuarios).

RF2 Modificar un usuario.

2.1 Se realiza una búsqueda.

2.2 Se modifican los datos deseados, todos pueden ser modificados excepto el nombre de usuario.

RF3 Eliminar un usuario.

3.1 Se realiza una búsqueda.

3.2 Se muestra un mensaje de confirmación que si es aceptado se eliminará el usuario buscado.

RF4 Autenticar usuario.

4.1 Permite la entrada al sistema del usuario y el acceso solo a las acciones que puede hacer según su rol.

4.1.1 Datos a insertar:

4.1.1.1 Usuario.

4.1.1.2 Contraseña.

RF5 Crear Comité.

5.1 Se insertan los datos de una estructura que no estén registrada en el sistema.

5.1.1 Datos a insertar:

5.1.1.1 Tipo de comité.

5.1.1.2 Subordinado a.

5.1.1.3 Nombre.

5.1.1.4 Estructura.

5.1.1.5 Nombre del Centro (sólo si se crea un comité de base o un comité de base independiente).

5.1.1.6 Sector de la economía (sólo si se crea un comité de base o un comité de base independiente).

5.1.1.7 Rama (sólo si se crea un comité de base o un comité de base independiente).

5.1.1.8 Provincia.

5.1.1.9 Municipio.

5.1.1.10 Tipo de centro (sólo si se crea un comité de base o un comité de base independiente).

RF6 Modificar Comité.

6.1 Para modificar una estructura organizativa primeramente se selecciona el tipo de comité y luego el nombre del comité específico que se modificará. No podrán ser modificados los Comités Municipales, Comités Provinciales, el Comité Nacional ni el Comité Especial UJC.

6.2 De los comités se podrá modificar:

6.2.1 Nombre.

6.2.2 Subordinado a (comité al que se subordina).

6.3 Si es un comité de base o un comité de base independiente se podrán modificar además los siguientes datos:

6.3.1 Sector de la economía.

6.3.2 Rama.

6.3.3 Tipo de centro.

6.3.4 Nombre del Centro.

RF7 Listar estructura organizativa.

7.1 Muestra todas las estructuras organizativas existentes en un árbol jerárquico.

7.1.1 Comité Nacional.

7.1.2 Comité Provincial.

7.1.3 Comité Municipal.

7.1.4 Comité Distrital.

7.1.5 Comité UJC.

7.1.6 Comité Primario.

7.1.7 Comité de Base.

7.1.8 Comité de Base Independiente.

7.1.9 Comité Especial UJC.

RF8 Buscar Militante.

8.1 Criterio de búsqueda.

8.1.1 Nombre del militante.

8.2 Resultado de la búsqueda.

8.2.1 Lista de nombres correspondiente al criterio de búsqueda.

RF9 Crear un Universo Juvenil.

9.1 Se insertan los datos de un universo juvenil que no esté registrado en el sistema.

9.1.1 Datos a insertar:

9.1.1.1 Nombre(s).

9.1.1.2 Apellidos.

9.1.1.3 Carné de Identidad.

9.1.1.4 Sexo.

9.1.1.5 Raza.

9.1.1.6 Militancia (si ha sido militante en otra ocasión).

9.1.1.7 Nivel Cultural.

9.1.1.8 Observaciones.

9.1.1.9 Clasificador General.

9.1.1.10 Clasificador Ocupacional.

9.1.1.11 Ubicación.

9.1.1.12 Es Activo.

RF10 Modificar datos generales del Universo Juvenil.

10.1 Se realiza una búsqueda.

10.2 Se modifican los datos deseados. Todos los datos pueden ser modificados.

RF11 Eliminar Universo Juvenil.

11.1 Se realiza una búsqueda.

11.2 Se muestra un mensaje de confirmación que si es aceptado se eliminará el Universo Juvenil buscado.

RF12 Realizar Crecimiento.

12.1 Permite iniciar crecimiento a un Universo Juvenil.

12.1.1 Se realiza una búsqueda.

12.2 Agregar pasos y su fecha de cumplimiento dentro del proceso.

12.2.1 Si el miembro del Universo Juvenil se acoge al principio de voluntariedad debe terminar el proceso de crecimiento.

12.2.2 Pasos:

12.2.2.1 Vía de ingreso.

- 12.2.2.1.1 Asamblea de ejemplares.
- 12.2.2.1.2 Consulta con las masas.
- 12.2.2.2 Fecha de inicio del proceso.
- 12.2.2.3 Cursó escuelas políticas.
- 12.2.2.4 Principio de voluntariedad.
- 12.2.2.5 Ingreso (ingresa o no ingresa).
- 12.2.2.6 Ingreso a la organización.
- 12.2.2.7 Organización que realizó el proceso.
- 12.2.2.8 Provincia donde se realizó el proceso.
- 12.2.2.9 Municipio donde se realizó el proceso.
- 12.3 Modificar Pasos.
 - 12.3.1 Se realiza una búsqueda.
 - 12.3.2 Se modifican los datos. Todos los datos pueden ser modificados.
- RF13** Crear Expediente del Militante.
 - 13.1 Se crea un expediente cuando se da un alta que no sea por traslado interno.
 - 13.1.1 Datos del expediente.
 - 13.1.1.1 Datos Personales:
 - 13.1.1.1.1 Nombres (s).
 - 13.1.1.1.2 Apellidos.
 - 13.1.1.1.3 Dirección.
 - 13.1.1.1.4 Provincia.
 - 13.1.1.1.5 Municipio.
 - 13.1.1.1.6 Ciudadanía.
 - 13.1.1.1.7 Carné de Identidad.
 - 13.1.1.1.8 Número de Carné del militante.
 - 13.1.1.1.9 Sexo.
 - 13.1.1.1.10 Raza.
 - 13.1.1.1.11 Teléfono.
 - 13.1.1.1.12 Lugar de nacimiento.
 - 13.1.1.1.13 Nivel cultural vencido.

13.1.1.2 Datos Familiares:

13.1.1.2.1 Parentesco.

13.1.1.2.2 Nombre (s).

13.1.1.2.3 Apellidos.

13.1.1.3 Datos Laborales:

13.1.1.3.1 Clasificador General.

13.1.1.3.2 Clasificador Ocupacional.

13.1.1.3.3 Labor.

13.1.1.4 Proceso de Ingreso:

13.1.1.4.1 Vía de ingreso utilizada.

13.1.1.4.1.1 Asamblea de ejemplares.

13.1.1.4.1.2 Consulta con las masas.

13.1.1.4.2 Escuelas políticas cursadas (Si o No).

13.1.1.4.3 Escuelas administrativas (Si o No).

13.1.1.4.4 Fecha de ingreso a la UJC.

13.1.1.4.5 Fecha de ingreso al PCC.

13.1.1.4.6 Organización que desarrollo el proceso.

13.1.1.4.6.1 Nombre de la organización.

13.1.1.4.7 Cargos del militante.

13.1.1.5 Datos de Interés:

13.1.1.5.1 Misión Internacionalista.

13.1.1.5.1.1 Seleccionar misión (ver detalles de la misión).

13.1.1.5.2 Nueva misión Internacionalista.

13.1.1.5.2.1 Tipo de misión.

13.1.1.5.2.2 Duración.

13.1.1.5.2.3 País.

13.1.1.5.2.4 Motivo.

13.1.1.5.2.5 Fecha.

13.1.1.6 Asignarle un comité de base al militante.

13.1.1.5.3 Comité de Base.

RF14 Modificar datos generales del Expediente.

14.1 Se realiza una búsqueda.

14.2 Se modifican los datos. Todos los datos pueden ser modificados.

RF15 Eliminar Expediente.

15.1 Se realiza una búsqueda.

15.2 Se muestra un mensaje de confirmación que si es aceptado se eliminará el expediente.

RF16 Alta por Crecimiento.

16.1 Se realiza una búsqueda. El resultado son todos los miembros del universo juvenil que tengan el proceso de crecimiento terminado.

16.2 Se selecciona la persona y se le crea un nuevo expediente.

RF17 Alta por Ajuste.

17.1 Tipos de Altas por ajuste:

17.1.1 Error de cómputo.

17.1.1.1 Se crea un expediente.

17.1.2 Doble militancia.

17.1.2.1 Se crea un expediente.

17.1.3 Regreso al antiguo Comité de Base.

17.1.3.1 Se realiza una búsqueda.

17.1.3.2 Se actualiza el expediente.

RF18 Traslado Interno.

18.1 Se realiza una búsqueda.

18.2 Datos para realizar el movimiento:

18.2.1 Nombre.

18.2.2 Comité origen.

18.2.3 Comité destino.

18.2.4 Fecha.

18.3 Se actualiza el expediente (sección de asignación).

RF19 Alta por Traslado Externo.

19.1 Tipos de traslado externo:

19.1.1 Provincia.

19.1.1.1 Datos para realizar el alta:

- 19.1.1.1.1 Provincia origen.
- 19.1.1.1.2 Municipio origen.
- 19.1.1.1.3 Provincia destino.
- 19.1.1.1.4 Municipio destino.
- 19.1.1.1.5 Fecha de incorporación.

19.1.2 Municipio.

19.1.2.1 Datos para realizar el alta:

- 19.1.2.1.1 Municipio origen.
- 19.1.2.1.2 Municipio destino.
- 19.1.2.1.3 Fecha de incorporación.

19.1.3 FAR.

19.1.3.1 Datos para realizar el alta:

- 19.1.3.1.1 Fecha de incorporación.

19.1.4 MININT.

19.1.4.1 Datos para realizar el alta:

- 19.1.4.1.1 Fecha de incorporación.

19.1.5 Exterior.

19.1.5.1 Datos para realizar el alta:

- 19.1.5.1.1 Fecha de incorporación.

RF20 Baja por Traslado Externo.

20.1 Tipo de traslado externo.

20.1.1 Provincia.

- 20.1.1.1 Se realiza una búsqueda.
- 20.1.1.2 Datos para realizar la baja:
 - 20.1.1.2.1 Nombre.
 - 20.1.1.2.2 Provincia origen.
 - 20.1.1.2.3 Municipio origen.
 - 20.1.1.2.4 Fecha.

20.1.2 Municipio.

20.1.2.1 Se realiza una búsqueda.

20.1.2.2 Datos para realizar la baja:

20.1.2.1.1 Nombre.

20.1.2.1.2 Provincia origen.

20.1.2.1.3 Municipio origen.

20.1.2.1.4 Fecha.

20.1.3 FAR.

20.1.3.1 Se realiza una búsqueda.

20.1.3.2 Datos para realizar la baja:

20.1.3.2.1 Fecha.

20.1.3.2.2 Nombre.

20.1.4 MININT.

20.1.4.1 Se realiza una búsqueda.

20.1.4.2 Datos para realizar la baja:

20.1.4.2.1 Fecha.

20.1.4.2.2 Nombre.

20.1.5 Exterior.

20.1.5.1 Se realiza una búsqueda.

20.1.5.2 Datos para realizar la baja:

20.1.5.2.1 Fecha.

20.1.5.2.2 Nombre.

RF21 Baja por Ajuste.

21.1 Tipo de Baja por ajuste:

21.1.1 Doble militancia.

21.1.2.1 Se realiza una búsqueda.

21.1.2.2 Datos para realizar la baja:

21.1.2.2.1 Fecha.

21.1.2.2.2 Nombre.

RF22 Baja Natural.

22.1 Se realiza una búsqueda.

22.2 Datos para realizar la baja:

22.2.1 Motivo

22.2.2 Fecha.

22.2.3 Nombre.

RF23 Baja por Defunción.

23.1 Se realiza una búsqueda.

23.2 Datos para realizar la baja:

23.2.1 Fecha.

23.2.2 Nombre.

RF24 Crear Centro.

24.1 Datos a insertar.

24.1.1 Nombre.

24.1.2 Tipo.

RF25 Modificar Centro.

25.1 Se realiza una búsqueda.

25.2 Se modifican los datos. Todos los datos pueden ser modificados.

RF26 Buscar UJ.

26.1 Criterio de búsqueda.

26.1.1 Nombre del miembro del universo juvenil.

26.2 Resultado de la búsqueda.

26.2.1 Lista de nombres correspondientes al criterio de búsqueda.

RF27 Buscar Centro.

27.1 Criterio de búsqueda.

27.1.1 Nombre del centro.

27.2 Resultado de la búsqueda.

27.2.1 Lista de nombres correspondientes al criterio de búsqueda.

RF28 Buscar Comité.

28.1 Criterio de búsqueda.

29.1.1 Nombre del comité.

28.2 Resultado de la búsqueda.

28.2.1 Lista de nombres correspondientes al criterio de búsqueda.

2.9 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener y que lo hacen atractivo, usable, rápido y confiable.

✓ **Apariencia o interfaz externa**

El acceso a la aplicación y su presentación será sencilla y fácil de manejar por el personal de la UJC. Tendrá una interfaz amigable y similar al ambiente de desarrollo del sistema operativo Windows. Poseerá el color azul fundamentalmente cumpliendo con los requisitos siguientes:

- Las ventanas del sistema contendrán claro y bien estructurados los datos, permitirán la interpretación correcta e inequívoca de la información.
- La interfaz contará con teclas de función y menús desplegables que faciliten y aceleren su utilización.
- El diseño responderá a la ejecución de acciones de una manera rápida, minimizando los pasos a dar en cada proceso.
- La corrección de errores en la introducción de datos será clara y fácil de realizar. La entrada de datos incorrecta será detectada claramente e informada al usuario.
- Todos los textos y mensajes en pantalla aparecerán en idioma español.
- Su funcionamiento será intuitivo y requerirá de información mínima.

✓ **Usabilidad**

Para el uso óptimo de la aplicación es necesario tener conocimientos básicos de computación. Está previsto que sea utilizada por el personal del Comité Nacional de la UJC y todas las direcciones de la UJC tanto municipales como provinciales, así como los distritos de todo el país.

✓ **Confiabilidad**

La aplicación tiene que ser altamente confiable pues a través de ella se manipularán datos confidenciales que serán utilizados en estudios e investigaciones, en dependencia del buen uso de esta información serán los resultados obtenidos.

✓ **Portabilidad**

El sistema debe permitir ser ejecutado sobre los Sistemas Operativos Windows y Linux dada la tendencia migratoria al software libre.

✓ **Software**

Se deberá disponer para el uso de la aplicación de cualquier Sistema Operativo que tenga una máquina virtual de Java compatible.

✓ **Hardware**

Para el funcionamiento de la aplicación se requieren máquinas con los siguientes requisitos:

Como mínimo 512 MB de memoria RAM, 20 GB de capacidad del disco duro necesario para la instalación y funcionamiento y una tarjeta de red activada.

✓ **Restricciones de diseño e implementación**

Será una aplicación de escritorio, el desarrollo de la aplicación será basado en la metodología RUP que utiliza como lenguaje de modelado UML. Se utilizará como lenguaje de programación Java, y herramienta CASE Visual Paradigm para el modelado de los artefactos que se generan en cada uno de los flujos de trabajo definidos por RUP.

✓ **Rendimiento**

Será una aplicación de escritorio, lo que le confiere una mayor rapidez, pues no tiene ningún tipo de intercambio a través de la red.

✓ **Seguridad**

- Se mantendrá seguridad y control a nivel de usuarios y contraseñas, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan.

Las contraseñas sólo podrán ser cambiadas por el propio usuario o por el administrador del sistema.

Confidencialidad: La información manejada por el sistema estará protegida de acceso no autorizado y divulgación.

Integridad: La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes.

Disponibilidad: La disponibilidad del sistema debe estar bajo la modalidad 24 x 7 todos los días naturales del año, no permitiéndose servicios de mantenimiento preventivos que afecten la vitalidad del mismo. Estos procesos deberán realizarse en caliente.

✓ Legales

El Módulo Militante será propiedad de la Unión de Jóvenes Comunistas, entregándose la aplicación, el código fuente y la documentación del sistema.

✓ Confiabilidad

- Se debe prever el mayor número de errores en el sistema y cómo solucionarlos.
- Se realizarán salvas periódicas de la información en otros dispositivos y lugares.

2.10 Actores del sistema

Los actores del sistema no son parte de él, intercambian información con el sistema, son un recipiente pasivo de información y representan el rol que juega una o varias personas, un equipo o un sistema automatizado.

Actor	Descripción
Control y documentación	Es el responsable de gestionar el expediente del militante y los movimientos de los mismos.
Funcionario de procesos políticos	Es el responsable de gestionar los datos del universo juvenil y realizar el crecimiento a la UJC.

Administrador	Es el responsable de gestionar los usuarios y asignarle sus responsabilidades.
Miembro del buró UJC	Es el responsable de gestionar las estructuras organizativas y los centros.
Usuario	Es la generalización de todos los actores.

Tabla 2.4: Actores del sistema

2.11 Listado de casos de uso

CU_1	CUS_ Gestionar Usuario
Actor	Administrador
Descripción	<p>El caso de uso se inicia cuando el Administrador va a realizar alguna de las siguientes operaciones relacionadas con la gestión de los usuarios:</p> <ul style="list-style-type: none">• Crear• Modificar• Eliminar <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.</p>
Referencia	RF 1, RF 2, RF 3

Tabla 2.5: Resumen del caso de uso Gestionar Usuario

CU_2	CUS_ Autenticar Usuario
Actor	Usuario
Descripción	El caso de uso inicia cuando algún Usuario desea hacer una operación en la aplicación, permite al Usuario autenticarse para tener acceso al sistema, finalizando así el caso de uso.
Referencia	RF 4

Tabla 2.6: Resumen del caso de uso Autenticar Usuario

CU_3	CUS_ Gestionar Estructura
Actor	Miembro del Buró UJC
Descripción	<p>El caso de uso se inicia cuando el Miembro del buró UJC va a realizar alguna de las siguientes operaciones relacionadas con la gestión de las estructuras:</p> <ul style="list-style-type: none">• Crear• Modificar• Listar <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.</p>
Referencia	RF 5, RF 6, RF 7

Tabla 2.7: Resumen del caso de uso Gestionar Estructura

CU_4	CUS_ Buscar Militante
Actor	Funcionario de Procesos Políticos y funcionario de Documentación y Control
Descripción	<p>Permite la búsqueda de un militante. El caso de uso inicia cuando algún actor desea buscar un militante, se introduce el parámetro de búsqueda y el sistema busca los militantes que coincidan con el parámetro entrado, el actor en cuestión selecciona el militante deseado finalizando así el caso de uso.</p>
Referencia	RF 8

Tabla 2.8: Resumen del caso de uso Buscar Militante

CU_5	CUS_ Gestionar Datos UJ
Actor	Funcionario de Procesos Políticos
Descripción	<p>El caso de uso se inicia cuando el Funcionario de Procesos Políticos va a realizar alguna de las siguientes operaciones relacionadas con la gestión de los datos del universo juvenil.</p> <ul style="list-style-type: none">• Insertar• Modificar• Eliminar <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.</p>

Referencia	RF 9, RF 10, RF 11
-------------------	--------------------

Tabla 2.9: Resumen del caso de uso Gestionar Datos UJ

CU_6	CUS_ Gestionar Crecimiento
Actor	Funcionario de Procesos Políticos
Descripción	El caso de uso inicia cuando se comienza el crecimiento de un miembro del universo juvenil, en el se recogen los pasos que se siguen en el crecimiento y su fecha de cumplimiento. El caso de uso finaliza cuando son completados todos los pasos (queriendo decir que ingresa a la UJC) o cuando se acoge al principio de voluntariedad (en este caso no ingresa a la UJC).
Referencia	RF 12

Tabla 2.10: Resumen del caso de uso Gestionar Crecimiento

CU_7	CUS_ Crear Expediente
Actor	Funcionario de Control y Documentación
Descripción	<p>El caso de uso se inicia cuando el funcionario de Control y Documentación va a crear un expediente:</p> <ul style="list-style-type: none"> • Crear <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta la acción necesaria. El caso de uso finaliza cuando se ejecuta la acción solicitada.</p>
Referencia	RF 13

Tabla 2.11: Resumen del caso de uso Crear Expediente

CU_8	CUS_ Gestionar Expediente
Actor	Funcionario de Documentación y Control
Descripción	<p>El caso de uso se inicia cuando el funcionario de Control y Documentación va a realizar alguna de las siguientes operaciones relacionadas con la gestión del expediente:</p> <ul style="list-style-type: none"> • Modificar • Eliminar <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.</p>

Referencia	RF 14, RF 15
-------------------	--------------

Tabla 2.12: Resumen del caso de uso Gestionar Expediente

CU_9	CUS_ Realizar Alta por Crecimiento
Actor	Funcionario de Documentación y Control
Descripción	El caso de uso inicia cuando el funcionario de Control y Documentación selecciona realizar alta por crecimiento, el sistema permite insertar un nuevo militante en el sistema y crear el expediente correspondiente. Finalizando así el caso de uso.
Referencia	RF 16

Tabla 2.13: Resumen del caso de uso Realizar Alta por Crecimiento

CU_10	CUS_ Realizar Alta por Ajuste
Actor	Funcionario de Documentación y Control
Descripción	El caso de uso inicia cuando el funcionario de Control y Documentación selecciona realizar alta por ajuste, el sistema permite realizar el alta, actualizar el expediente del militante y de ser necesario crear el expediente correspondiente. Finalizando así el caso de uso.
Referencia	RF 17

Tabla 2.14: Resumen del caso de uso Realizar Alta por Ajuste

CU_11	CUS_ Realizar Traslado Interno
Actor	Funcionario de Documentación y Control
Descripción	El caso de uso inicia cuando el funcionario de control y documentación selecciona realizar un traslado interno, el sistema da baja al militante del comité de base en el que se encuentre militando e inmediatamente lo incorpora en un nuevo comité de base. Finalizando así el caso de uso.
Referencia	RF 18

Tabla 2.15: Resumen del caso de uso Realizar Traslado Interno

CU_12	CUS_ Realizar Alta por Traslado Externo
Actor	Funcionario de Documentación y Control
Descripción	El caso de uso inicia cuando el funcionario de Control y Documentación selecciona realizar alta por traslado externo, el sistema permite realizar el alta y crear el expediente correspondiente.
Referencia	RF 19

Tabla 2.16: Resumen del caso de uso Realizar Alta por Traslado Externo

CU_13	CUS_ Realizar Baja por Traslado Externo
Actor	Funcionario de Documentación y Control
Descripción	El caso de uso inicia cuando el funcionario de Control y Documentación selecciona realizar baja por traslado externo, el sistema da baja al militante. Finalizando así el caso de uso.
Referencia	RF 20

Tabla 2.17: Resumen del caso de uso Realizar Baja por Traslado Externo

CU_14	CUS_ Realizar Baja por Ajuste
Actor	Funcionario de Documentación y Control
Descripción	El caso de uso inicia cuando el funcionario de Control y Documentación selecciona realizar baja por ajuste, el sistema da baja al militante. Finalizando así el caso de uso.
Referencia	RF 21

Tabla 2.18: Resumen del caso de uso Realizar Baja por Ajuste

CU_15	CUS_ Realizar Baja Natural
Actor	Funcionario de Documentación y Control
Descripción	El caso de uso inicia cuando el funcionario de Control y Documentación selecciona realizar baja natural, el sistema da baja al militante. Finalizando así el caso de uso.
Referencia	RF 22

Tabla 2.19: Resumen del caso de uso Realizar Baja Natural

CU_16	CUS_ Realizar Baja por Defunción
Actor	Funcionario de Documentación y Control
Descripción	El caso de uso inicia cuando el funcionario de Control y Documentación selecciona realizar baja por defunción, el sistema da baja al militante. Finalizando así el caso de uso.
Referencia	RF 23

Tabla 2.20: Resumen del caso de uso Realizar Baja por Defunción

CU_17	CUS_ Gestionar Centro
Actor	Miembro de buró UJC
Descripción	<p>El caso de uso se inicia cuando el Miembro de buró UJC va a realizar alguna de las siguientes operaciones relacionadas con la gestión del centro.</p> <ul style="list-style-type: none">• Crear• Modificar <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.</p>
Referencia	RF 24, RF 25

Tabla 2.21: Resumen del caso de uso Gestionar Centro

CU_18	CUS_ Buscar UJ
Actor	Funcionario de Procesos Políticos
Descripción	Permite la búsqueda de un miembro del universo juvenil. El caso de uso inicia cuando algún actor desea buscar un miembro del universo juvenil, se introduce el parámetro de búsqueda y el sistema busca los miembros del universo juvenil que coincidan con el parámetro entrado, el actor en cuestión selecciona el miembro del universo juvenil deseado finalizando así el caso de uso.
Referencia	RF 26

Tabla 2.22: Resumen del caso de uso Buscar UJ

CU_19	CUS_ Buscar Centro
Actor	Miembro del Buró UJC
Descripción	Permite la búsqueda de un centro. El caso de uso inicia cuando el Miembro del Buró UJC desea buscar un centro, se introduce el parámetro de búsqueda y el sistema busca los centros que coincidan con el parámetro entrado, el actor en cuestión selecciona el centro deseado finalizando así el caso de uso.
Referencia	RF 27

Tabla 2.23: Resumen del caso de uso Buscar Centro

CU_20	CUS_ Buscar Comité
Actor	Miembro del Buró UJC
Descripción	Permite la búsqueda de un comité. El caso de uso inicia cuando el Miembro del Buró UJC desea buscar un comité, se introduce el parámetro de búsqueda y el sistema busca los comité que coincidan con el parámetro entrado, el actor en cuestión selecciona el comité deseado finalizando así el caso de uso.
Referencia	RF 28

Tabla 2.24: Resumen del caso de uso Buscar Comité

2.12 Diagrama de casos de uso del sistema

Un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores.

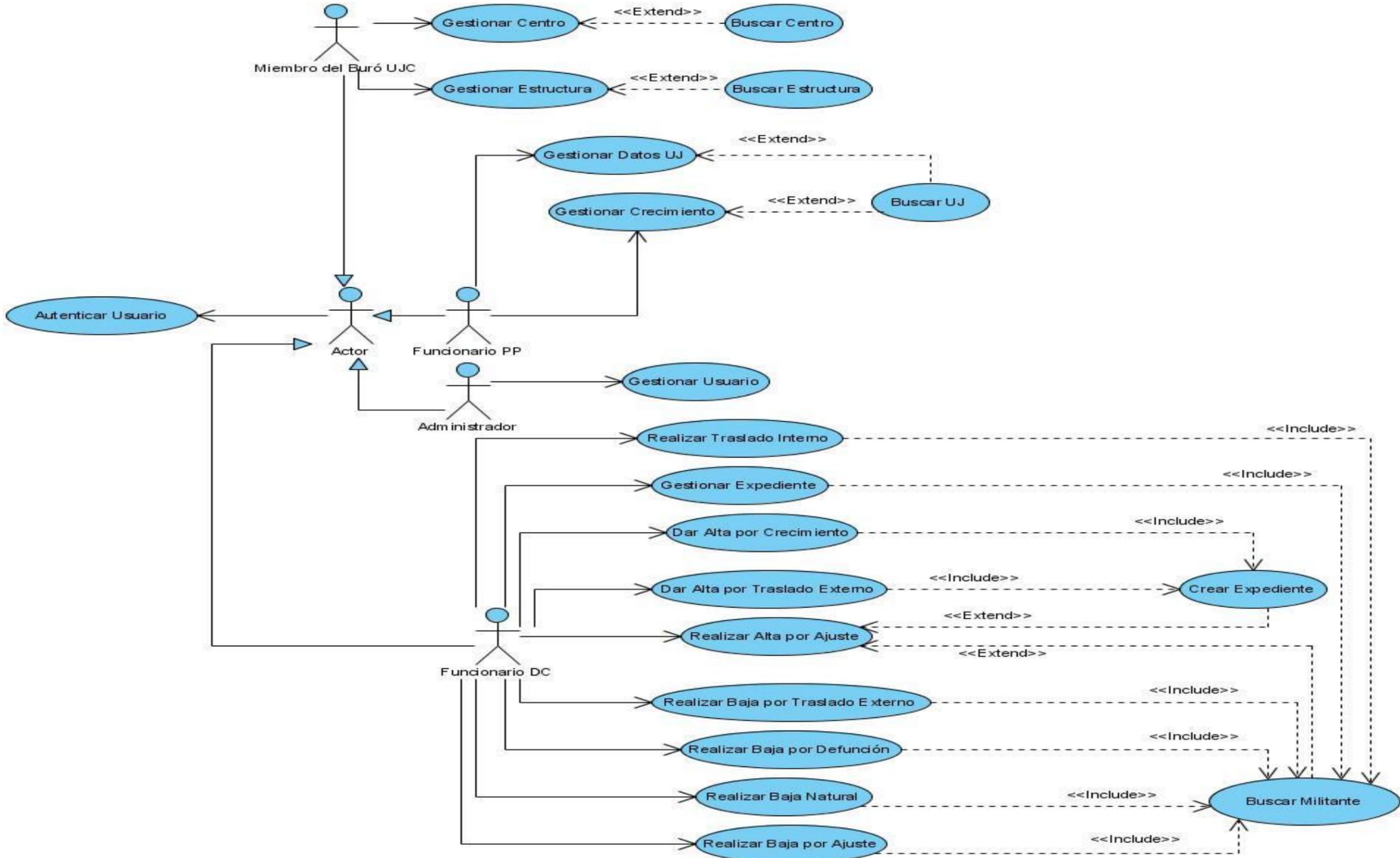


Figura 3: Diagrama de caso de uso del sistema

2.13 Descripción de los casos de uso del sistema

Para ver la descripción ampliada de los restantes casos de uso ver Anexo 4.

Caso de uso Gestionar Datos UJ:

Caso de Uso:	Gestionar Datos UJ	
Actores:	Funcionario de Procesos Políticos	
Resumen:	<p>El caso de uso se inicia cuando el Funcionario de Procesos Políticos va a realizar alguna de las siguientes operaciones relacionadas con la gestión de los datos del universo juvenil.</p> <ul style="list-style-type: none"> • Insertar • Modificar • Eliminar <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.</p>	
Precondiciones:	El Funcionario de Procesos Políticos debe haberse autenticado correctamente en el sistema.	
Referencias	RF 9, RF 10, RF 11	
Prioridad	Crítica	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Funcionario de Procesos Políticos selecciona la opción Universo Juvenil, a través del menú Vida Interna.	2. El sistema muestra una interfaz con las opciones Insertar, Modificar o eliminar.	
<p>3.a) Si el actor desea insertar un nuevo miembro al universo juvenil ver sesión "Insertar UJ".</p> <p>b) Si el actor desea modificar o eliminar los datos de un miembro del universo juvenil existente ver sección "Modificar o Eliminar UJ".</p>		

Sección "Insertar UJ"	
Acción del Actor	Respuesta del Sistema
3. El Funcionario de Procesos Políticos selecciona la opción Insertar.	4. El sistema muestra la interfaz que le permitirá al Funcionario de Procesos Políticos insertar un miembro al universo juvenil.
5. El Funcionario de Procesos Políticos llena los campos y solicita insertar los datos mediante el botón Insertar.	6. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos.
	7. El sistema guarda los datos.
	8. El sistema muestra un mensaje al Funcionario de Procesos Políticos informando que la inserción se realizó correctamente. Finalizando así el caso de uso.
Flujo Alterno 6	
Acción del Actor	Respuesta del Sistema
	6. El sistema detecta algún error y lo informa mediante un mensaje. Se pasa a la acción 5.
Prototipos de interfaz	

Sección "Modificar o Eliminar UJ"	
Acción del Actor	Respuesta del Sistema
3. El Funcionario de Procesos Políticos selecciona la opción modificar o eliminar.	4. El sistema muestra una interfaz de búsqueda que le permitirá al Funcionario de Procesos Políticos buscar al miembro del universo juvenil a través del parámetro "Nombre" (ver caso de uso Buscar UJ) además de la acción que desee realizar.
	5. Después de seleccionado el universo juvenil, el actor selecciona la acción que desea realizar.
6. a) Si el actor selecciona la opción Modificar ver sesión "Modificar UJ". b) Si el actor selecciona la opción Eliminar ver sesión "Eliminar UJ".	

Prototipos de interfaz



Sección "Modificar UJ"	
Acción del Actor	Respuesta del Sistema
6. El Funcionario de Procesos Políticos selecciona la opción modificar.	7. El sistema muestra todos los datos de la persona seleccionada.

8. El actor modifica los datos que desee y selecciona continuar con la operación mediante el botón Aceptar.	9. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos.
	10. El sistema guarda los datos, informando que la acción tuvo éxito mediante un mensaje. Finalizando así el caso de uso.

Flujo Alterno 9

Acción del Actor	Respuesta del Sistema
	9. El sistema encuentra algún error y lo informa mediante un mensaje. Se pasa a la acción 8.

Prototipos de interfaz

Modificar datos del Universo Juvenil

Datos Personales

Nombre: José Carné de identidad: 12345678901

2do Nombre: Color de piel: Blanca

Apellido: Martínez Sexo: Masculino

2do Apellido: Pérez Militancia: No

Posibilidad de ser procesado Nivel Cultural: Nivel Medio S...

Ubicación

Ubicación: CB Sandino

Traslado a:

- Comité Nacional
- Comité Provincial La Habana
- Comité Provincial Pinar del Río

Datos Laborales

Clasificador General: MILITANTES ESTUDIANTE...

Clasificador Ocupacional: FEU

Labor: Estudiante

Observaciones: Buen estudiante

Modificar Cancelar

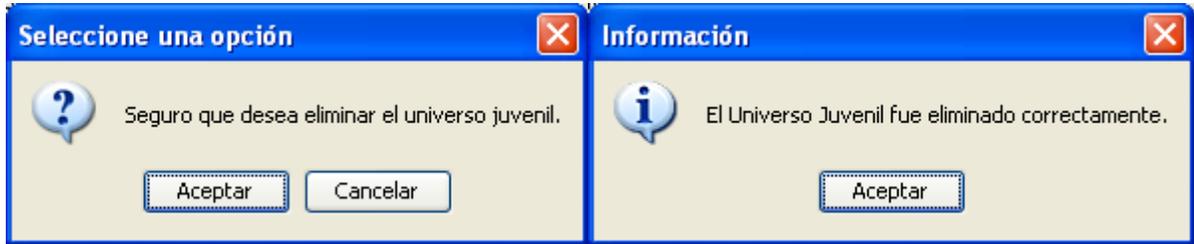
Sección "Eliminar UJ"	
Acción del Actor	Respuesta del Sistema
6. El Funcionario de Procesos Políticos selecciona la opción eliminar.	7. El sistema muestra un mensaje de confirmación preguntando al actor si está seguro de realizar la operación.
8. El Funcionario de Procesos Políticos acepta el mensaje.	9. El sistema elimina al miembro del universo juvenil y muestra un mensaje informando que la operación tuvo éxito. Finalizando así el caso de uso.
Flujo Alterno 8	
Acción del Actor	Respuesta del Sistema
8. El Funcionario de Procesos Políticos no acepta el mensaje. Se pasa a la acción 5.	
Prototipos de interfaz	
	
Poscondiciones	Se agregó un miembro al universo juvenil, modificó e eliminó uno ya existente.

Tabla 2.25: Descripción ampliada del caso de uso Gestionar Datos UJ

2.14 Conclusiones

En este capítulo se inició el desarrollo de la propuesta de solución y luego del análisis de los procesos del negocio y las acciones de los clientes y trabajadores expresados en las descripciones de los casos de uso y sus respectivos diagramas de actividades se obtuvieron los requisitos funcionalidades que deben cumplir el sistema, así como las características que harán el producto atractivo y fácil de usar expresadas en una lista de requisitos no funcionales. Tras el análisis de los procesos del negocio y las funcionalidades que debe cumplir el sistema finalmente se pudieron obtener los casos de uso del sistema, los cuales fueron descritos detalladamente y a través de los diagramas de casos de uso del sistema se mostró la interacción entre actores y casos de uso.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

En este capítulo se realizará el análisis del sistema para obtener una visión sobre qué es lo que hace el sistema, a través de diagramas que suavizarán la transición al diseño, el cual se desarrollará también en el presente capítulo, de modo que sea consistente con el entorno de implementación, permitiendo mostrar los detalles descriptivos de la solución propuesta. Se mostrarán diagramas con el objetivo de comprender la estructura interna de la aplicación a desarrollar, así como el estudio y selección de los patrones de diseño que se usarán para el desarrollo del sistema.

3.1 Estilo arquitectónico: Arquitectura en tres capas

La arquitectura de una aplicación es la vista conceptual de la estructura de esta. Toda aplicación contiene código de presentación, código de procesamiento de datos y código de almacenamiento de datos. La arquitectura de las aplicaciones difiere según como está distribuido este código. (CORNEJO, 2009)

La arquitectura en 3 capas es un estilo de programación, su objetivo primordial es la separación de la capa de presentación, capa de negocio y la capa de datos:

Capa de presentación

Esta capa resuelve la presentación de datos al usuario, se encarga de "dibujar" las pantallas de la aplicación al usuario, y tomar los eventos que el cliente genere (por ejemplo, el hacer click en un botón).

Capa de lógica de negocio

En esta capa irá todo el código que define las reglas de negocio (cálculos, validaciones). Surge de los procesos que hemos encontrado en el análisis.

Capa de acceso a datos

Esta capa resuelve el acceso a datos, abstrayendo a su capa superior de la complejidad del acceso e interacción con los diferentes orígenes de datos, en ella irá el código que permite acceder a las fuentes de datos. Esencialmente trata sobre 4 operaciones básicas, llamadas CRUD (por Create- Retrieve- Update y Delete), que se realizan sobre cualquier fuente de datos.

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado, además la centralización de los aspectos de seguridad y transaccionalidad, que serían responsabilidad del modelo, otras ventaja sería la no replicación de lógica de negocio en los clientes: esto permite que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de los usuarios, reduciendo los costes de mantenimiento.

3.2 Patrones de diseño

“Un patrón es una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas”. Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptado para resolver un problema de diseño general en un contexto particular, identifica: clases, instancias, roles, colaboraciones y la distribución de responsabilidades.(LARMAN, 2004)

Definición de patrón de diseño

“Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.” (SORIANO, 2008)

Patrones GRASP

“Los patrones GRASP, acrónimo que significa General Responsibility Assignment Software Patterns, describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones”.(LARMAN, 2004)

3.2.1 Patrones GRASP básicos

Existen actualmente 9 patrones GRASP, pero cinco de ellos son básicos para diseñar cualquier software, ellos son:

- **Experto:**

Se encarga de asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

- **Creador:**

Asignarle a la clase B la responsabilidad de crear una instancia de clase A, en uno de los siguientes casos:

- B contiene a A.
- B es una agregación (o composición) de A.
- B almacena a A.
- B tiene los datos de inicialización de A (datos que requiere su constructor).
- B usa a A.

Alta Cohesión:

Asignar una responsabilidad de modo que la cohesión siga siendo alta. Una clase tiene responsabilidades moderadas en un área funcional y colabora con las otras para llevar a cabo las tareas. Una clase de alta cohesión posee un número relativamente pequeño, con una importante funcionalidad relacionada y poco trabajo por hacer. Colabora con otros objetos para compartir el esfuerzo si la tarea es grande. El patrón alta cohesión presenta semejanzas con el mundo real, si alguna persona se le asignan demasiadas responsabilidades no será eficiente en ninguna de ellas.

- **Bajo Acoplamiento:**

Asigna una responsabilidad para mantener bajo acoplamiento. Las clases deben comunicarse con un número pequeño de clases tanto como sea posible.

- **Controlador:**

Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. En el desarrollo del presente trabajo de diploma se aplican todos los patrones básicos de GRASP en los diagramas de clases del diseño.

- **Patrón Polimorfismo**

El patrón Polimorfismo un importante patrón estratégico en el diseño orientado a objetos. Es un principio fundamental en que se fundan las estrategias globales, o planes de ataque, al diseño como organizar un sistema para que se encargue del trabajo. Un diseño basado en la asignación de responsabilidades mediante el polimorfismo puede ser extendido fácilmente para que realice nuevas variantes. (LARMAN, 2004)

3.2.2 Patrones GoF

Patrones de diseño que se presentan en el libro *Design Patterns*, dado que el libro fue escrito por cuatro autores (Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides), a estos 23 patrones expuestos en esa publicación se les conoce hoy con el nombre de “Gang of Four” (Pandilla de los Cuatro) o con la abreviatura en inglés “GoF”. Se clasifican en tres categorías: creación, estructurales y comportamiento.(SORIANO, 2008)

- Patrones de creación: Muestran la guía de cómo crear objetos cuando sus creaciones requieren tomar decisiones. Estas decisiones normalmente serán resueltas dinámicamente decidiendo que clases instanciar o sobre que objetos un objeto delegará responsabilidades.
- Patrones estructurales: Describen la forma en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros.
- Patrones de comportamiento: Se utilizan para organizar, manejar y combinar comportamientos.

3.2.3 Patrón estructural

Los patrones estructurales están relacionados con cómo las clases y los objetos se combinan para dar lugar a estructuras más complejas.

- **Patrón Facade**

Es un patrón que sirve para proveer de una interfaz unificada sencilla que haga de intermediaria entre un cliente y una interfaz o grupo de interfaces más complejas. La idea principal es la de ocultar todo lo posible la complejidad de un sistema, el conjunto de clases o componentes que lo forman, de forma que solo se ofrezca un (o unos pocos) punto de entrada al sistema tapado por la fachada.

Ventajas

- Reduce la complejidad y minimiza las dependencias.
- Los clientes que usan fachada no necesitan acceder a los objetos del sistema.
- Facilita la utilización y comprensión (acompañando la interfaz con una documentación mínima) de los sistemas ocultos.
- Aisla a los clientes de cambios sustanciales en los sistemas ocultos, ya sea un cambio de requisitos, de diseño de base de datos, de tecnología utilizada entre otros.

3.2.4 Patrón de acceso a datos

- **Patrón DAO (*Data Access Object*, Objeto de Acceso a Datos)**

Este patrón surge históricamente de la necesidad de gestionar una diversidad de fuentes de datos, aunque su uso se extiende al problema de encapsular no sólo la fuente de datos, sino además ocultar la forma de acceder a los datos. Se trata de que el software cliente se centre en los datos que necesita y se olvide de cómo se realiza el acceso a los datos o de cual es la fuente de almacenamiento.

Ventajas

- Se baja en nivel de acoplamiento entre clases, reduciendo la complejidad de realizar cambios.
- Se aísla las conexiones a la fuente de datos en una capa fácilmente identificable y mantenible.
- Cualquier objeto de negocio (aquel que contiene detalles específicos de operación o aplicación) no requiere conocimiento directo del destino final de la información que manipula.

A continuación se exponen las características de otro patrón GoF aplicado. (SORIANO, 2008)

3.3 Análisis

Modelo de análisis

Objetivos

Comprender los requisitos del software y no cómo se implementará la solución; logrando una profundización más precisa de los requerimientos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema entero, incluyendo su arquitectura.

3.3.1 Diagrama de clases del análisis

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones (incluyendo herencia, agregación, asociación, etcétera). Los diagramas de clase son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (diseño).

EL diagrama de clases del análisis es un artefacto del flujo de trabajo de Análisis y Diseño en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada. Las siguientes figuras muestran el diagrama de clases del análisis del caso de uso Gestionar Datos UJ. Para ver los restantes diagramas de clases del análisis consultar Anexo 7.

Diagrama de clases del análisis CU_Gestionar Datos UJ

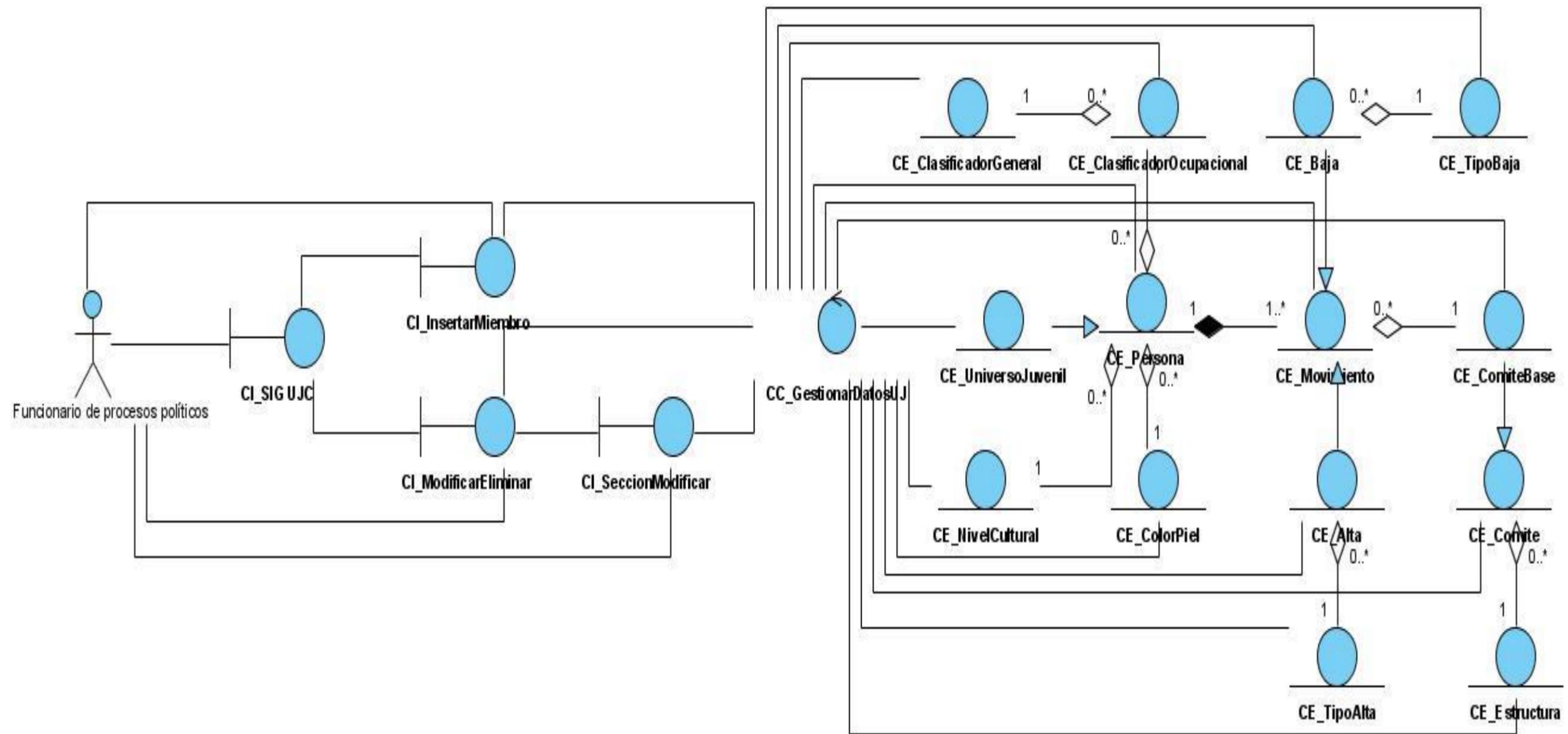


Figura 4: DCA CU_Gestionar Datos UJ

3.4 Diseño.

En el diseño se modela el sistema para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. El diseño es la primera etapa técnica del proceso de ingeniería del software y tiene como principal artefacto el modelo de diseño que constituye la entrada fundamental del flujo de trabajo de implementación.

3.4.1 Diagramas de clases del diseño

Un diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces de la aplicación. Sirve también para visualizar las relaciones entre las clases que involucran el sistema. Las siguientes figuras representan el diagrama de clases del diseño para el caso de uso Gestionar Datos UJ. Para ver los restantes diagramas de clases del diseño consultar Anexo 8.

Diagrama de clases del diseño CU_ Gestionar Datos UJ

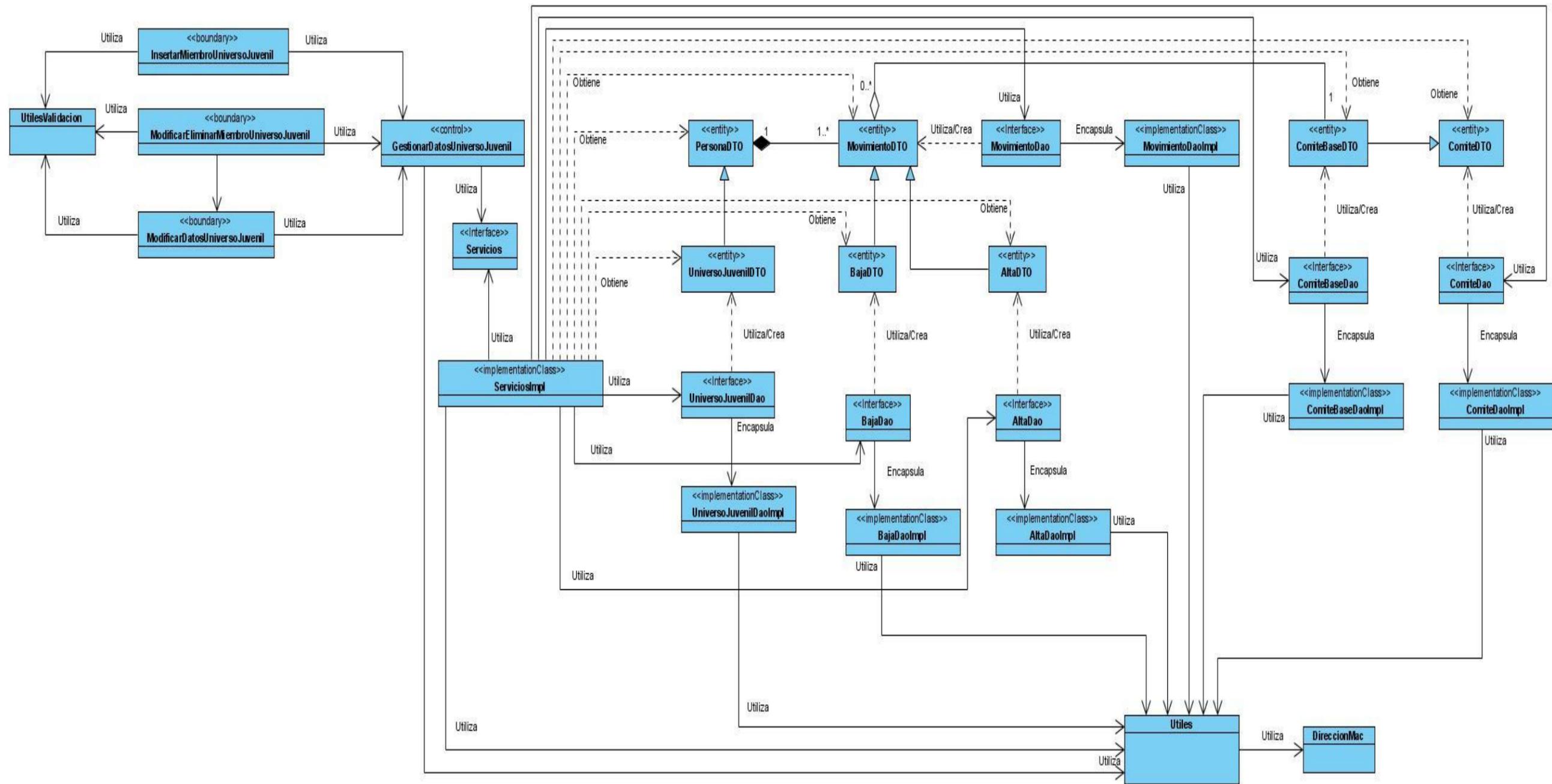


Figura 5: DCD CU_ Gestionar Datos UJ

3.4.2 Descripción de las clases del diseño

Para ver las restantes descripciones de las clases del diseño ver Anexo 5.

Nombre: Persona	
Tipo de clase: Entidad	
Atributo	Tipo
id	String
ci	String
primerNombre	String
primerApellido	String
segundoNombre	String
segundoApellido	String
sexo	Character
labor	String
clasificadorOcupacional	ClasificadorOcupacional
colorPiel	ColorPiel
nivelCultural	NivelCultural
insertado	Boolean
actualizado	Boolean
eliminado	Boolean
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Tabla 3.1: Descripción de la clase Persona

Nombre: Militante	
Tipo de clase: Entidad	
Atributo	Tipo
carnetMilitante	String
esActivo	Boolean
esServicioMilitar	Boolean
cursoEscuelaPolitica	Boolean
cursoEscuelaAdm	Boolean
esEstudiante	Boolean
annoCursa	AnnoCursa
ciudadania	Ciudadania
direccion	Direccion
lugarNacimiento	String
misiones	List<Mision>
telefonos	List<Telefono>
direccionesParticulares	List<Direccion>
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.
Nombre:	adicionarMision(Mision mision)
Descripción:	Adiciona una misión.
Nombre:	adicionarCargo(Cargo cargo)
Descripción:	Adiciona un cargo.

Nombre:	adicionarTelefono(Telefono telefono)
Descripción:	Adiciona un teléfono.
Nombre:	existeCargo(Cargo cargo)
Descripción:	Pregunta pro la existencia en la base de datos de un cargo dado.
Nombre:	existeMision(Mision mision)
Descripción:	Pregunta pro la existencia en la base de datos de una misión dada.
Nombre:	existeTelefono(Telefono telefono)
Descripción:	Pregunta pro la existencia en la base de datos de un teléfono dado.

Tabla 3.2: Descripción de la clase Militante

Nombre: MilitanteDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarMilitante(Militante militante)
Nombre:	eliminarMilitante(Militante militante)
Nombre:	actualizarMilitante(Militante militante)
Nombre:	obtenerMilitante(String primerNombre, String segundoNombre)
Nombre:	obtenerMilitanteCI(String ci)
Nombre:	obtenerTodosMilitantes()
Nombre:	obtenerMilitantesComite(ComiteBase comiteBase)

Tabla 3.3: Descripción de la clase MilitanteDao

Nombre: MilitanteDaoImpl
Tipo de clase:

Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarMilitante(Militante militante)
Descripción:	Guarda un militante dado en la base de datos.
Nombre:	eliminarMilitante(Militante militante)
Descripción:	Elimina un militante dado de la base de datos.
Nombre:	actualizarMilitante(Militante militante)
Descripción:	Actualiza los datos de un militante dado.
Nombre:	obtenerMilitante(String primerNombre, String segundoNombre)
Descripción:	Obtiene un militante dado su primer nombre y su segundo apellido.
Nombre:	obtenerMilitanteCI(String ci)
Descripción:	Obtiene un militante dado su carné de identidad.
Nombre:	obtenerTodosMilitantes()
Descripción:	Obtiene un listado de todos los militantes existentes en la base de datos.
Nombre:	obtenerMilitantesComite(ComiteBase comiteBase)
Descripción:	Obtiene los militantes de un comité de base dado.

Tabla 3.4: Descripción de la clase MilitanteDaolmpl

Nombre: UniversoJuvenil	
Tipo de clase: Entidad	
Atributo	Tipo
posibilidadProcesado	Boolean
fueMilitante	Boolean
observacion	String
esActivo	Boolean

Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Tabla 3.5: Descripción de la clase UniversoJuvenil

Nombre: UniversoJuvenilDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarUniverso(UniversoJuvenil universo)
Nombre:	eliminarUniverso(UniversoJuvenil universo)
Nombre:	actualizarUniverso(UniversoJuvenil universo)
Nombre:	obtenerUniverso(String nombre)
Nombre:	obtenerUniversoVerificar(String carne)
Nombre:	obtenerTodosUniversos()

Tabla 3.6: Descripción de la clase UniversoJuvenilDao

Nombre: UniversoJuvenilDaoImpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarUniverso(UniversoJuvenil universo)
Descripción:	Guarda un universo juvenil dado en la base de datos.
Nombre:	eliminarUniverso(UniversoJuvenil universo)
Descripción:	Elimina un universo juvenil dado de la base de datos.

Nombre:	actualizarUniverso(UniversoJuvenil universo)
Descripción:	Actualiza los datos de un universo juvenil dado.
Nombre:	obtenerUniverso(String nombre)
Descripción:	Obtiene un universo juvenil dado su nombre.
Nombre:	obtenerUniversoVerificar(String carne)
Descripción:	Obtiene un universo juvenil dado su carné.
Nombre:	obtenerTodosUniversos()
Descripción:	Obtiene una lista de todos los miembros del universo juvenil existentes en la base de datos.

Tabla 3.7: Descripción de la clase UniversoJuvenilDaolmpl

Nombre: Crecimiento	
Tipo de clase: Entidad	
Atributo	Tipo
idCrecimiento	String
universoJuvenil	UniversoJuvenil
vialIngreso	VialIngreso
comite	Comite
cursoEscuelasPoliticass	Date
fechalnicioProceso	Date
principioVoluntariedad	Date
ingreso	Boolean
ingresoOrganizacion	Date
aprobacionIngreso	Date

insertado	Boolean
actualizado	Boolean
eliminado	Boolean
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Tabla 3.8: Descripción de la clase Crecimiento

Nombre: CrecimientoDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarCrecimiento(Crecimiento crecimiento)
Nombre:	obtenerTodosCrecimiento()
Nombre:	actualizarCrecimiento(Crecimiento crecimiento)
Nombre:	obtenerCrecimiento(String idCrecimiento)
Nombre:	eliminarCrecimiento(Crecimiento crecimiento)
Nombre:	obtenerTodosProcesosCrecimiento(String idUniverso)

Tabla 3.9: Descripción de la clase CrecimientoDao

Nombre: CrecimientoDaoImpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarCrecimiento(Crecimiento crecimiento)

Descripción:	Guarda un crecimiento dado en la base de datos.
Nombre:	obtenerTodosCrecimiento()
Descripción:	Obtiene una lista de todos los crecimientos existentes en la base de datos.
Nombre:	actualizarCrecimiento(Crecimiento crecimiento)
Descripción:	Actualiza los datos de un crecimiento dado.
Nombre:	obtenerCrecimiento(String idCrecimiento)
Descripción:	Obtiene un crecimiento dado su identificador.
Nombre:	eliminarCrecimiento(Crecimiento crecimiento)
Descripción:	Elimina un crecimiento dado de la base de datos.
Nombre:	obtenerTodosProcesosCrecimiento(String idUniverso)
Descripción:	Obtiene un listado de los procesos de crecimiento de un universo juvenil dado el identificador del mismo.

Tabla 3.10: Descripción de la clase CrecimientoDaoImpl

Nombre: Movimiento	
Tipo de clase: Entidad	
Atributo	Tipo
idMovimiento	String
comiteBase	ComiteBase
universoJuvenil	UniversoJuvenil
persona	Persona
fechaMovimiento	Date
curso	String
insercion	Boolean
actualizacion	Boolean

eliminacion	Boolean
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Tabla 3.11: Descripción de la clase Movimiento

Nombre: MovimientoDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarMovimiento(Movimiento movimiento)
Nombre:	obtenerTodosMovimientosAntesDeUnaFecha(Date fin)

Tabla 3.12: Descripción de la clase MovimientoDao

Nombre: MovimientoDaoImpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarMovimiento(Movimiento movimiento)
Descripción:	Guarda un movimiento dado en la base de datos.
Nombre:	obtenerTodosMovimientosAntesDeUnaFecha(Date fin)
Descripción:	Obtiene una lista de los movimientos que fueron realizados antes de una fecha especificada.

Tabla 3.13: Descripción de la clase MovimientoDaoImpl

Nombre: Alta

Tipo de clase: Entidad	
Atributo	Tipo
tipoAlta	TipoAlta
motivoTipoAlta	MotivoTipoAlta
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Tabla 3.14: Descripción de la clase Alta

Nombre: AltaDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarAlta(Alta alta)
Nombre:	obtenerTodasAltas()
Nombre:	actualizarAlta(Alta alta)
Nombre:	obtenerAlta(Integer idAlta)
Nombre:	obtenerAltaUniverso(String idUniversoJuvenil)
Nombre:	eliminarAlta(Alta alta)
Nombre:	obtenerTodasALtasAntesDeUnaFecha(Date fin)
Nombre:	obtenerTodasPersonasAltasAntesDeUnaFecha(Date fin)
Nombre:	obtenerTodasPersonasAltasPorUniversoJuvenilAntesDeUnaFecha(Date fin)
Nombre:	obtenerTodasPersonasAltasPorCrecimientoAntesDeUnaFecha(Date fin)
Nombre:	obtenerTodasPersonasAltasPorIncorporacionesInternasAntesDeUnaFecha(Date fin)

Nombre:	obtenerTodasPersonasAltasPorAjusteDeUnaFecha(Date fin)
Nombre:	obtenerTodasPersonasAltasIncorporacMunicipiosDeUnaFecha(Date fin)
Nombre:	obtenerTodasPersonasAltasIncorporacProvinciasDeUnaFecha(Date fin)
Nombre:	obtenerTodasPersonasAltasIncorporacDeLasFARAntesDeUnaFecha(Date fin)
Nombre:	obtenerTodasPersonasAltasIncorporacDeMININTAntesDeUnaFecha(Date fin)
Nombre:	obtenerTodasPersonasAltasIncorporacExteriorAntesDeUnaFecha(Date fin)

Tabla 3.15: Descripción de la clase AltaDao

Nombre: AltaDaoImpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarAlta(Alta alta)
Descripción:	Guarda un alta dada en la base de datos.
Nombre:	obtenerTodasAltas()
Descripción:	Obtiene un listado de todas las altas de la base de datos.
Nombre:	actualizarAlta(Alta alta)
Descripción:	Actualiza los datos de un alta dada.
Nombre:	obtenerAlta(Integer idAlta)
Descripción:	Obtiene un alta dado su identificador.
Nombre:	obtenerAltaUniverso(String idUniversoJuvenil)
Descripción:	Obtiene un alta perteneciente a un universo juvenil dado el identificador del mismo.
Nombre:	eliminarAlta(Alta alta)
Descripción:	Elimina un alta dada de la base de datos.

Capítulo 3: Análisis y Diseño del Sistema.

Nombre:	obtenerTodasAltasAntesDeUnaFecha(Date fin)
Descripción:	Obtiene un listado de todas las altas que han sido efectuadas antes de una fecha dada.
Nombre:	obtenerTodasPersonasAltasAntesDeUnaFecha(Date fin)
Descripción:	Obtiene un listado de las personas que han sido dadas de alta antes de una fecha indicada.
Nombre:	obtenerTodasPersonasAltasPorUniversoJuvenilAntesDeUnaFecha(Date fin)
Descripción:	Obtiene un listado de las personas que han sido dadas de alta de tipo universo juvenil antes de una fecha indicada.
Nombre:	obtenerTodasPersonasAltasPorCrecimientoAntesDeUnaFecha(Date fin)
Descripción:	Obtiene un listado de las personas que han sido dadas de alta por crecimiento antes de una fecha indicada.
Nombre:	obtenerTodasPersonasAltasPorIncorporacionesInternasAntesDeUnaFecha(Date fin)
Descripción:	Obtiene un listado de las personas que han sido dadas de alta por traslado interno antes de una fecha indicada.
Nombre:	obtenerTodasPersonasAltasPorAjusteDeUnaFecha(Date fin)
Descripción:	Obtiene un listado de las personas que han sido dadas de alta por ajuste antes de una fecha indicada.
Nombre:	obtenerTodasPersonasAltasIncorporacMunicipiosDeUnaFecha(Date fin)
Descripción:	Obtiene un listado de las personas que han sido dadas de alta por traslado externo desde un municipio antes de una fecha indicada.
Nombre:	obtenerTodasPersonasAltasIncorporacProvinciasDeUnaFecha(Date fin)
Descripción:	Obtiene un listado de las personas que han sido dadas de alta por traslado externo desde una provincia antes de una fecha indicada.
Nombre:	obtenerTodasPersonasAltasIncorporacDeLasFARAntesDeUnaFecha(Date fin)
Descripción:	Obtiene un listado de las personas que han sido dadas de alta por traslado externo desde las FAR antes de una fecha indicada.
Nombre:	obtenerTodasPersonasAltasIncorporacDeMININTAntesDeUnaFecha(Date fin)
Descripción:	Obtiene un listado de las personas que han sido dadas de alta por traslado externo desde el

	MININT antes de una fecha indicada.
Nombre:	obtenerTodasPersonasAltasIncorporacExteriorAntesDeUnaFecha(Date fin)
Descripción:	Obtiene un listado de las personas que han sido dadas de alta por traslado externo desde el exterior antes de una fecha indicada.

Tabla 3.16: Descripción de la clase AltaDaoImpl

Nombre: Baja	
Tipo de clase: Entidad	
Atributo	Tipo
tipoBaja	TipoBaja
motivoTipoBaja	MotivoTipoBaja
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Tabla 3.17: Descripción de la clase Baja

Nombre: BajaDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarBaja(Baja baja)
Nombre:	obtenerTodasBajas()
Nombre:	actualizarBaja(Baja baja)
Nombre:	obtenerBaja(Integer idBaja)
Nombre:	obtenerTodasBajasAntesDeUnaFecha(Date fin)

Nombre:	obtenerTodasPersonasBajasAntesDeUnaFecha(Date fin)
----------------	--

Tabla 3.18: Descripción de la clase BajaDao

Nombre: BajaDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarBaja(Baja baja)
Descripción:	Guarda una baja dada en la base de datos.
Nombre:	obtenerTodasBajas()
Descripción:	Obtiene un listado de todas las bajas de la base de datos.
Nombre:	actualizarBaja(Baja baja)
Descripción:	Actualiza los datos de una baja dada.
Nombre:	obtenerBaja(Integer idBaja)
Descripción:	Obtiene una baja dado su identificador.
Nombre:	obtenerTodasBajasAntesDeUnaFecha(Date fin)
Descripción:	Obtiene una lista de las bajas que han sido realizadas antes de una fecha indicada.
Nombre:	obtenerTodasPersonasBajasAntesDeUnaFecha(Date fin)
Descripción:	Obtiene una lista de las personas que han sido dadas de baja antes de una fecha indicada.

Tabla 3.19: Descripción de la clase BajaDaolmpl

3.4.3 Diagramas de interacción

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos, junto

con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento.

Diagramas de secuencia

Un diagrama de secuencia muestra las interacciones entre objetos ordenadas en secuencia temporal, muestra los objetos que se encuentran en el escenario y la secuencia de mensajes intercambiados entre los objetos para llevar a cabo la funcionalidad descrita por el escenario. (ANA, 2008)

Consultar anexo 9.

3.5 Diseño de la base de datos

Modelo de clases persistentes

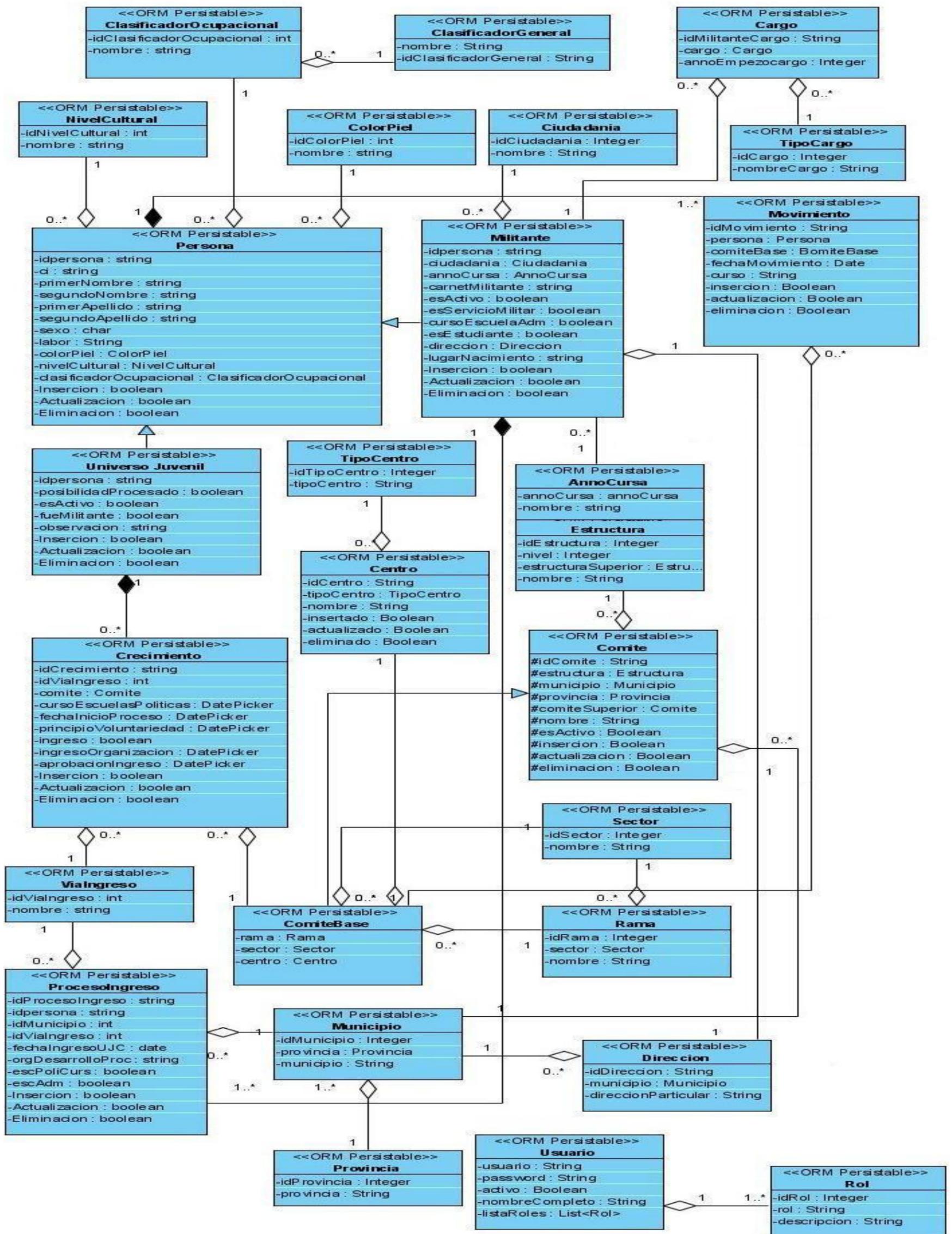


Figura 6: Modelo de clases persistentes (Parte 1)

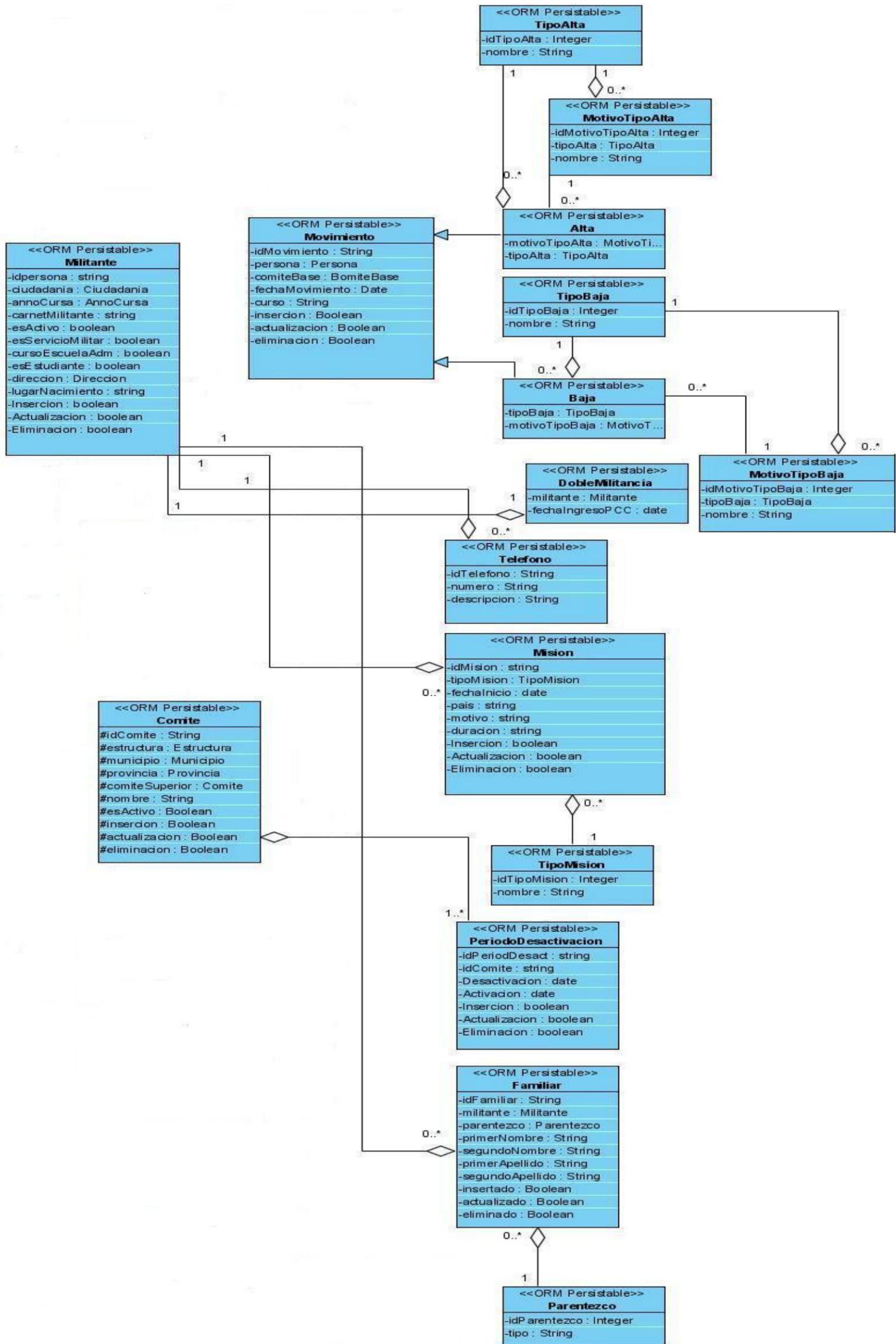


Figura 7: Modelo de clases persistentes (Parte 2)

3.5.1 Descripción de las clases persistentes

Para ver las restantes descripciones de las clases persistentes ver Anexo 6.

Nombre: Movimiento		
Descripción: Recoge los datos relacionados con los movimientos.		
Atributo	Tipo	Descripción
idMovimiento	String	Identificador principal del movimiento.
persona	Persona	Representa a la persona que causa el movimiento.
comiteBase	ComiteBase	Hace referencia al comité de base al que pertenece el militante.
fechaMovimiento	Date	Fecha de alta y/o fecha de baja.
curso	String	Fecha del curso escolar, Ej. 2008-2009.
insercion	Boolean	Toma valor verdadero si es insertado un movimiento.
actualizacion	Boolean	Toma valor verdadero si un movimiento es actualizado.
eliminacion	Boolean	Toma valor verdadero si es eliminado un movimiento.

Tabla 3.20: Descripción de la clase Movimiento

Nombre: Alta		
Descripción: Recoge los datos relacionados con dar alta a las filas de la UJC.		
Atributo	Tipo	Descripción
motivoTipoAlta	MotivoTipoAlta	Hace referencia al motivo del tipo de alta.
tipoAlta	TipoAlta	Hace referencia al tipo de alta.

Tabla 3.21: Descripción de la clase Alta

Nombre: Baja		
Descripción: Recoge los datos relacionados con dar baja de las filas de la UJC.		
Atributo	Tipo	Descripción
tipoBaja	TipoBaja	Hace referencia al tipo de baja.
motivoTipoBaja	MotivoTipoBaja	Hace referencia al motivo del tipo de baja.

Tabla 3.22: Descripción de la clase Baja

Nombre: Crecimiento		
Descripción: Recoge los datos relacionados con el proceso de crecimiento de los militantes.		
Atributo	Tipo	Descripción
idCrecimiento	String	Identificador principal del crecimiento.
vialIngreso	VialIngreso	Hace referencia a la vía por la que se le hace el crecimiento a la persona.
universoJuvenil	UniversoJuvenil	Hace referencia al universo juvenil al que se le realiza el proceso de crecimiento.
comite	Comite	Hace referencia al comité donde se está realizando el proceso de crecimiento.
cursoEscuelasPoliticas	Date	Hace referencia a la fecha en la que cursa escuela política.
fechaInicioProceso	Date	Hace referencia a la fecha en la que se inicia el proceso.
principioVoluntariedad	Date	Hace referencia a la fecha en la que se acoge al principio de voluntariedad.
ingreso	Boolean	Devuelve verdadero si finalmente ingresa.
ingresoOrganizacion	Date	Hace referencia a la fecha de ingreso a las filas.
aprobacionIngreso	Date	Hace referencia a la fecha donde se aprueba el ingreso.
Insercion	Boolean	Toma valor verdadero si es insertado.
Actualizacion	Boolean	Toma valor verdadero si es

		actualizado.
Eliminacion	Boolean	Toma valor verdadero si es eliminado.

Tabla 3.23: Descripción de la clase Crecimiento

Nombre: UniversoJuvenil		
Descripción: Recoge los datos relacionados con el universo juvenil.		
Atributo	Tipo	Descripción
idpersona	String	Identificador principal de la persona que es universo juvenil.
posibilidadProcesado	Boolean	Devuelve verdadero si tiene posibilidad de ser procesado.
esActivo	Boolean	Devuelve verdadero si está como militante activo en algún comité.
fueMilitante	Boolean	Devuelve verdadero si la persona fue militante anteriormente.
observacion	String	Hace referencia a alguna observación que se le quiera hacer a la persona.
Insercion	Boolean	Toma valor verdadero si es insertado.
Actualizacion	Boolean	Toma valor verdadero si es actualizado.
Eliminacion	Boolean	Toma valor verdadero si es eliminado.

Tabla 3.24: Descripción de la clase UniversoJuvenil

Nombre: Persona		
Descripción: Recoge los datos relacionados con la persona.		
Atributo	Tipo	Descripción
idpersona	String	Identificador principal de la persona
ci	String	Hace referencia al carné de identidad de la persona.
primerNombre	String	Hace referencia al primer nombre de la persona.

segundoNombre	String	Hace referencia al segundo nombre de la persona.
primerApellido	String	Hace referencia al primer apellido de la persona.
segundoApellido	String	Hace referencia al segundo apellido de la persona.
sexo	Character	Hace referencia al sexo de la persona.
clasificadorOcupacional	ClasificadorOcupacional	Hace referencia a la labor de la persona.
colorPiel	ColorPiel	Hace referencia al color de la piel de la persona.
nivelCultural	NivelCultural	Hace referencia al nivel cultural de la persona.
labor	String	Hace referencia a la clasificación ocupacional de la persona.
Insercion	Boolean	Toma valor verdadero si es insertado.
Actualizacion	Boolean	Toma valor verdadero si es actualizado.
Eliminacion	Boolean	Toma valor verdadero si es eliminado.

Tabla 3.25: Descripción de la clase Persona

Nombre: Militante		
Descripción: Recoge los datos relacionados con la evaluación de los militantes.		
Atributo	Tipo	Descripción
idpersona	String	Identificador principal de la persona.
ciudadania	Ciudadania	Hace referencia a la ciudadanía de un militante.
annoCursa	AnnoCursa	Hace referencia al año que cursa un militante.
carnetMilitante	String	Hace referencia al carnet de la militancia en la UJC
esActivo	Boolean	Hace referencia a si el militante está activo en algún comité.
esServicioMilitar	Boolean	Hace referencia a si el militante pasó el servicio militar.

cursoEscuelaAdm	Boolean	Hace referencia a si el militante cursó escuelas administrativas.
esEstudiante	Boolean	Hace referencia a si el militante es estudiante o no.
direccion	Direccion	Hace referencia a la dirección del militante.
lugarNacimiento	String	Hace referencia al lugar de nacimiento del militante.
Insercion	Boolean	Toma valor verdadero si es insertado.
Actualizacion	Boolean	Toma valor verdadero si es actualizado.
Eliminacion	Boolean	Toma valor verdadero si es eliminado.

Tabla 3.26: Descripción de la clase Militante

3.6 Diagrama de Despliegue

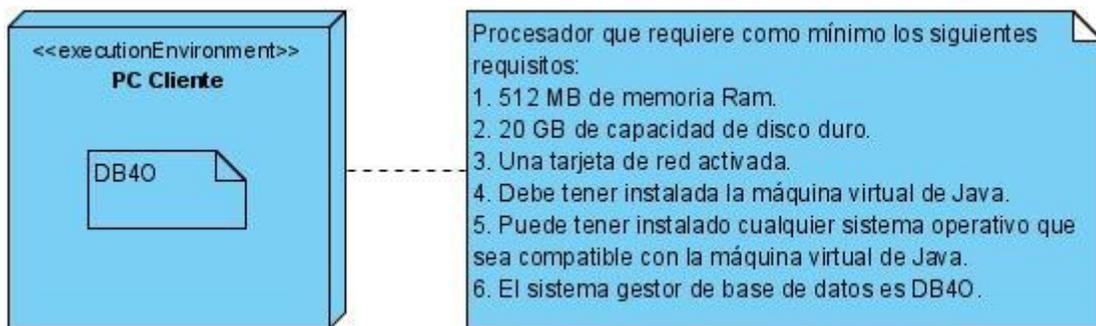


Figura 8: Diagrama de Despliegue

3.7 Conclusiones

El principal resultado del diseño es el Modelo de Diseño que conserva la estructura del sistema propuesto por el Modelo de Análisis. Se realizó la distribución física del sistema a través de un modelo de despliegue, que describe todas las configuraciones sobre las cuáles debe implementarse el sistema. Se pudo definir sobre la base de la modelación correspondiente a este capítulo, la estructura que debe seguir la implementación para una correcta práctica de la Programación Orientada a Objetos (POO) en cada uno de sus escenarios. Los diagramas de diseño y despliegue se consideran la entrada principal para el flujo de trabajo de implementación.

CAPÍTULO 4: IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se describirá la implementación del sistema donde los elementos del diseño se convertirán en elementos de implementación y éstos en términos de componentes. Se explicará el estándar de codificación a utilizar en la implementación del mismo así como las pruebas realizadas al producto. Partiendo de los resultados del diseño se identificarán los posibles componentes en los cuales se puede implementar la aplicación y se mostrarán además los diferentes diagramas de componentes.

4.1 Estándar de Codificación

Un lenguaje de programación es, como todo lenguaje, una forma de comunicación. Es decir, permite transmitir información entre dos entidades de forma que la idea original del transmisor pueda ser recuperada por el receptor.

Los lenguajes de programación tienen la particularidad de cumplir un doble rol. Por un lado sirven para comunicar humanos con computadoras. Son la forma de transformar una serie de abstracciones como algoritmos, módulos, tipos de datos y sistemas en algo que una computadora pueda ejecutar. El segundo rol, y que no se ve tanto a primera vista, es que un lenguaje de programación sirve para comunicar humanos con humanos. Por ejemplo, para que alguien le cuente un algoritmo a otro. O en muchos casos para que un programador pueda mañana recuperar las ideas que volcó en código hoy. La capacidad de expresar un significado usando un lenguaje depende tanto del lenguaje, como del emisor. Pero el emisor usualmente influye mucho más: es claro que la forma de construir un mensaje puede impactar muy fuertemente en su claridad. A la forma de construir el mensaje es frecuentemente llamado estilo. En el caso de los programadores el mensaje sería el código fuente, y la forma sería el estilo del código.

Si se utiliza un buen estilo de codificación el código constará de algunas cualidades, por ejemplo:

- **Extensibilidad:** la facilidad con que se adapta el software a cambios de especificación. Un buen estilo de código fomenta programas que no sólo resuelven el problema, sino que también reflejan claramente la relación problema/solución. Esto tiene como efecto que muchos cambios simples en el problema reflejen de forma obvia los cambios a hacer en el programa.

- **Verificabilidad:** la facilidad con que pueden comprobarse propiedades de un sistema. Si el estilo de código hace obvia la estructura del programa, eso ayuda a verificar que el comportamiento sea el esperado.
- **Reparabilidad:** la posibilidad de corregir errores sin demasiado esfuerzo.
- **Capacidad de evolución:** la capacidad de adaptarse a nuevas necesidades.
- **Comprensibilidad:** la facilidad con que el programa puede ser comprendido.

4.2 Estándar de codificación utilizado

Tipos de identificadores	Reglas para nombres	Ejemplos
Paquetes	El prefijo del nombre de un paquete se escribe siempre con letras <i>ASCII</i> en minúsculas, y debe ser uno de los nombres de dominio de alto nivel, actualmente com, edu, gov, mil, net, org, o uno de los códigos ingleses de dos letras que identifican cada país como se especifica en el ISO Standard 3166, 1981. Los subsecuentes componentes del nombre del paquete variarán de acuerdo a las convenciones de nombres internas de cada organización. Dichas convenciones pueden especificar que algunos nombres de los directorios correspondan a divisiones, departamentos, proyectos o máquinas.	cu.uci.kainos.ujc.gestión.modelo cu.uci.kainos.ujc.seguridad.modelo
Clases	Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas. Intentar mantener los nombres de las clases simples y descriptivas. Usar palabras completas,	class Persona; class Comite; class Crecimiento;

	<p>evitar acrónimos y abreviaturas (a no ser que la abreviatura sea mucho más conocida que el nombre completo, como <i>URL</i> o <i>HTML</i>).</p> <p>Nota: El estándar de codificación para los nombres de las clases será especificado con mayor detalle a continuación de esta tabla.</p>	
Interfaces	Los nombres de las interfaces siguen la misma regla que las clases.	<pre>interface UsuarioDao; interface ComiteDao;</pre>
Métodos	Los métodos deben ser verbos. Cuando son compuestos tendrán la primera letra en minúscula, y la primera letra de las siguientes palabras que lo forma en mayúscula.	<pre>guardarUsuario (); obtenerTodosUniversosJuveniles();</pre>
Variables	<p>Excepto las constantes, todas las instancias y variables de clase o método empezarán con minúscula. Las palabras internas que lo forman (si son compuestas) empiezan con su primera letra en mayúsculas. Los nombres de variables no deben empezar con los caracteres subguión "_" o signo del dólar "\$", aunque ambos están permitidos por el lenguaje. Los nombres de las variables deben ser cortos pero con significado. La elección del nombre de una variable debe ser un mnemónico, designado para indicar a un observador casual su función. Los nombres de variables de un solo caracter se deben evitar, excepto para variables índices temporales. Nombres comunes para variables temporales son i, j, k, m, y n para enteros; c, d, y e para caracteres.</p>	<pre>String nombre; Boolean esActivo;</pre>

Constantes	Los nombres de las variables declaradas como constantes deben ir totalmente en mayúsculas separando las palabras con un subguión ("_").	static final String DB_NAME="db/UJC_DB.yap"
Componentes	<p>Los nombres de los componentes visuales que se van a utilizar deben escribirse empezando con minúscula y comenzando por la j en caso de sean componentes de Swing y seguidamente las letras que identifican a cada componente, en caso contrario solo las letras que lo identifican, luego el nombre que se le de al componente comenzando con mayúscula y minúscula después, si es compuesto el segundo nombre comienza con mayúscula y no hay espacios entre el primero y el segundo.</p> <p>DatePicker Label Button Checkbox Radio Button Button Group Combo Box List TextArea Text Field Password Field Scroll Pane Panel</p>	<p>dtpPrincipioVoluntariedad jlbRol jbtnBuscar jckbEsActivo jrbIngresa btngpSeleccionarIngreso jcbRoles jlstUsuarios jtaObservaciones jtfNombre jpfPassword jspUsuarios jpnIngresoOrganizacion</p>

Tabla 4.1: Estándar de codificación

4.2.1 Convenciones de nombres utilizados

Se definen convenciones de nombres para las distintas clases java dependiendo de las funciones que tengan cada una de estas en la aplicación:

- Clases de la capa de Acceso a Datos:
 - Las interfaces que representan las operaciones sobre los objetos de acceso a datos, correspondientes al patrón de diseño Data Access Object (DEEPAK ALUR 2003) terminan con la palabra “Dao”. Ejemplo: UsuarioDao.
 - Las implementaciones reales de las interfaces Dao comienza con el nombre de la interfaz correspondiente y terminan con la palabra “Impl”. Ejemplo: UsuarioDaoImpl.
- Clases de la capa de Negocio:
 - Las interfaces que representan las funcionalidades de los DAO se declaran comenzando con la palabra “Servicios” seguido del nombre del módulo donde están ubicadas. Ejemplo: ServiciosGestión.
 - Las implementaciones reales de las interfaces que representan las funcionalidades de los DAO comienzan con el nombre de la interfaz y terminan con la palabra “Impl”. Ejemplo: ServiciosGestiónImpl.
- Clases de la capa de Presentación:
 - En esta capa se definen los JFrame y los JDialog. Su nombre se define empezando con mayúscula, si es un nombre compuesto tendrán la primera letra de cada palabra con mayúscula, los nombres deben ser sugerentes, de tal manera que se pueda tener idea de lo que contiene, pueden ser sustantivos o verbos. Ejemplo: InsertarMiembroUniversoJuvenil.

4.3 Diagrama de componentes

El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes. Un diagrama de componentes muestra las organizaciones y dependencias

lógicas entre componentes software, sean estos componentes de código fuente, binarios, archivos, paquetes, bibliotecas cargadas dinámicamente o ejecutables.

Las siguientes figuras muestran el diagrama de componentes del caso de uso Gestionar Datos UJ, para consultar los restantes ver Anexo 10.

CU_Gestionar Datos UJ

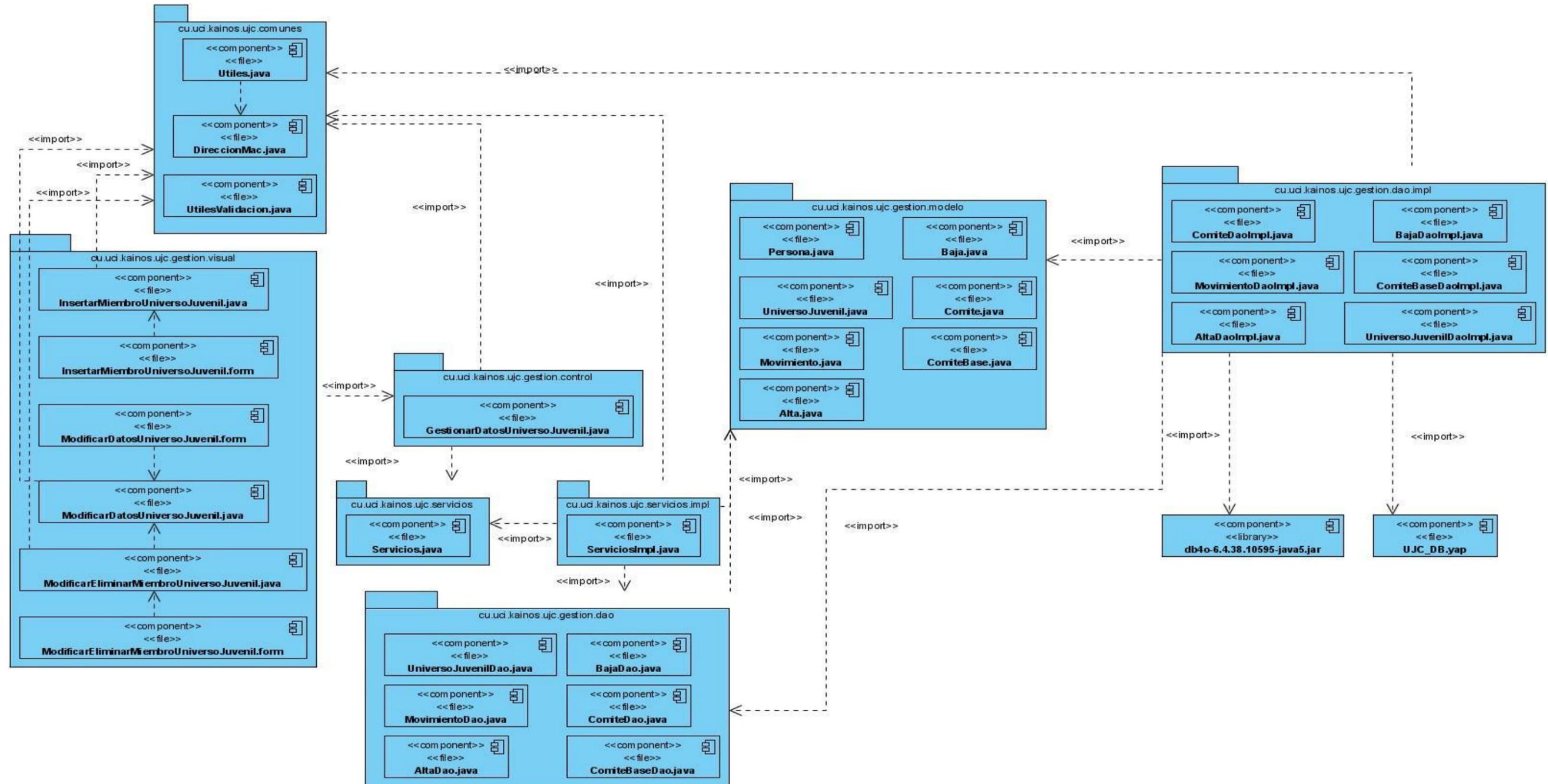


Figura: 9 Diagrama de componente del CU_Gestionar Datos UJ

4.4 Modelo de Prueba

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo condiciones o requerimientos especificados, los resultados son observados y registrados, y se hace una evaluación de algún aspecto del sistema o componente.

4.4.1 Pruebas de caja negra

Estas pruebas se realizan sobre la interfaz del software y con ellas se pretende demostrar no sólo que las funciones del software son operativas sino también que la entrada se acepta de forma adecuada y que se produce una salida correcta; además que la integridad de la información externa se mantiene.

Las pruebas de caja negra se centran principalmente en los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos.

Estas pruebas son de vital importancia para el software pues mediante ellas se pueden encontrar funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a las Bases de Datos externas, errores de rendimiento así como errores de inicialización y terminación.

Dentro de las técnicas de Prueba de Caja Negra, se utilizó la Partición de Equivalencia. Esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones de software. Mediante la aplicación de esta se examinaron los valores válidos e inválidos de las entradas existentes en el software, descubriendo de forma inmediata clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo así el número de clases de prueba que hay que desarrollar.

Casos de prueba

Para ver los restantes casos de prueba consultar Anexo 11.

Caso de Prueba CU: Gestionar Datos UJ

Sección Insertar

Condición de Entrada	Casos válidos	Casos no válidos
Nombre	Letras y números.	No introducir el nombre o introducir caracteres extraños.
Segundo Nombre	Letras y números.	Introducir caracteres extraños.
Apellido	Letras y números.	No introducir el primer apellido o introducir caracteres extraños.
Segundo Apellido	Letras y números.	No introducir el segundo apellido o introducir caracteres extraños.
Carné de Identidad	Números.	No introducir el carné de identidad, que la cantidad de números sea menor o mayor que 11 o introducir letras o caracteres extraños.
Color de piel	Seleccionable.	No seleccionar un color de piel.
Sexo	Seleccionable.	No seleccionar un sexo.
Militancia	Seleccionable.	No seleccionar una opción.
Nivel cultural	Seleccionable.	No seleccionar un nivel cultural.
Ubicación	Seleccionable.	No seleccionar un comité.

Tabla 4.2: Valores válidos e inválidos para el caso de prueba

Caso de uso: Gestionar Datos UJ
Caso de Prueba: Realizar inserción de los datos de un universo juvenil entrando correctamente todos los datos.
Entrada: Nombre: Emis Segundo Nombre: Kamila Apellido: Cantillo Segundo Apellido: Avilés Carné de Identidad: 86081528819 Color de piel: Mestiza Sexo: Femenino Militancia: No Nivel cultural: Secundaria Básica Ubicación: CB Mantua
Resultado: Se insertan los datos del universo juvenil en la base de datos.
Condiciones: Los datos para insertar un universo juvenil deben estar en el rango de valores válidos.

Tabla 4.3: Caso de prueba con todos los datos correctos

Caso de uso: Gestionar Datos UJ
Caso de Prueba: Realizar inserción de los datos de un universo juvenil entrando incorrectamente el primer nombre.
Entrada: Nombre: Fr@nklis

Segundo Nombre: Apellido: Cantillo Segundo Apellido: Ramirez Carné de Identidad: 86081528819 Color de piel: Mestiza Sexo: Masculino Militancia: No Nivel cultural: Secundaria Básica Ubicación: CB Mantua
Resultado: El sistema muestra un mensaje de error.
Condiciones: El nombre no puede contener caracteres extraños.

Tabla 4.4: Caso de prueba con el nombre incorrecto

Caso de uso: Gestionar Datos UJ
Caso de Prueba: Realizar inserción de los datos de un universo juvenil dejando el campo Nombre vacío.
Entrada: Nombre: Segundo Nombre: Apellido: Cantillo Segundo Apellido: Ramirez Carné de Identidad: 86081528819

Color de piel: Mestiza
Sexo: Masculino
Militancia: No
Nivel cultural: Secundaria Básica
Ubicación: CB Mantua
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Tabla 4.5: Caso de prueba con el campo Nombre vacío

Caso de uso: Gestionar Datos UJ
Caso de Prueba: Realizar inserción de los datos de un universo juvenil entrando incorrectamente el segundo nombre.
Entrada: Nombre: Ana Segundo Nombre: Mar&a Apellido: Fernández Segundo Apellido: Ramirez Carné de Identidad: 86081528819 Color de piel: Blanca Sexo: Femenino Militancia: Sí Nivel cultural: Secundaria Básica

Ubicación: CB Mantua
Resultado: El sistema muestra un mensaje de error.
Condiciones: El nombre no puede contener caracteres extraños.

Tabla 4.6: Caso de prueba con el segundo nombre incorrecto

Caso de uso: Gestionar Datos UJ
Caso de Prueba: Realizar inserción de los datos de un universo juvenil entrando incorrectamente el primer apellido.
Entrada: Nombre: Ariel Segundo Nombre: Apellido: Fernández? Segundo Apellido: Ramirez Carné de Identidad: 86081528819 Color de piel: Blanca Sexo: Masculino Militancia: No Nivel cultural: Secundaria Básica Ubicación: CB Mantua
Resultado: El sistema muestra un mensaje de error.
Condiciones: El apellido no puede contener caracteres extraños.

Tabla 4.7: Caso de prueba con el primer apellido incorrecto

Caso de uso: Gestionar Datos UJ
Caso de Prueba: Realizar inserción de los datos de un universo juvenil dejando el campo Apellido vacío.
Entrada: Nombre: Ariel Segundo Nombre: Apellido: Segundo Apellido: Ramirez Carné de Identidad: 86081528819 Color de piel: Blanca Sexo: Masculino Militancia: No Nivel cultural: Secundaria Básica Ubicación: CB Mantua
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Tabla 4.8: Caso de prueba con el campo Apellido vacío

Caso de uso: Gestionar Datos UJ
Caso de Prueba: Realizar inserción de los datos de un universo juvenil entrando incorrectamente el segundo apellido.
Entrada: Nombre: Ariel

Segundo Nombre: Apellido: Fernández Segundo Apellido: Ram/rez Carné de Identidad: 86081528819 Color de piel: Blanca Sexo: Masculino Militancia: No Nivel cultural: Secundaria Básica Ubicación: CB Mantua
Resultado: El sistema muestra un mensaje de error.
Condiciones: El apellido no puede contener caracteres extraños.

Tabla 4.9: Caso de prueba con el segundo apellido incorrecto

Caso de uso: Gestionar Datos UJ
Caso de Prueba: Realizar inserción de los datos de un universo juvenil dejando el campo Segundo Apellido vacío.
Entrada: Nombre: Ariel Segundo Nombre: Apellido: Fernández Segundo Apellido: Carné de Identidad: 86081528819

Color de piel: Blanca
Sexo: Masculino
Militancia: No
Nivel cultural: Secundaria Básica
Ubicación: CB Mantua
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Tabla 4.10: Caso de prueba con el campo Segundo Apellido vacío

Caso de uso: Gestionar Datos UJ
Caso de Prueba: Realizar inserción de los datos de un universo juvenil dejando el campo Carné de Identidad vacío.
Entrada:
Nombre: Emis
Segundo Nombre:
Apellido: Ramirez
Segundo Apellido: Matos
Carné de Identidad:
Color de piel: Blanca
Sexo: Femenino
Militancia: No
Nivel cultural: Secundaria Básica

Ubicación: CB Mantua
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Tabla 4.11: Caso de prueba con el campo Carné de Identidad vacío

Caso de uso: Gestionar Datos UJ
Caso de Prueba: Realizar inserción de los datos de un universo juvenil insertando un carné de identidad con una cantidad de dígitos distinto de 11.
Entrada: Nombre: Leidys Segundo Nombre: Mariam Apellido: Ramirez Segundo Apellido: Odelin Carné de Identidad: 030330 Color de piel: Mestiza Sexo: Femenino Militancia: No Nivel cultural: Secundaria Básica Ubicación: CB Mantua
Resultado: El sistema muestra un mensaje de error.
Condiciones: El carné de identidad debe tener 11 dígitos.

Tabla 4.12: Caso de prueba con el carné de identidad incorrecto (cantidad de dígitos diferente de 11)

Caso de uso: Gestionar Datos UJ
Caso de Prueba: Realizar inserción de los datos de un universo juvenil insertando un carné de identidad con letras.
Entrada: Nombre: Leidys Segundo Nombre: Mariam Apellido: Ramirez Segundo Apellido: Odelin Carné de Identidad: 030330sdr8k Color de piel: Mestiza Sexo: Femenino Militancia: No Nivel cultural: Secundaria Básica Ubicación: CB Mantua
Resultado: El sistema muestra un mensaje de error.
Condiciones: El carné de identidad debe contener sólo números.

Tabla 4.13: Caso de prueba con el carné de identidad incorrecto (introduciendo letras)

Caso de uso: Gestionar Datos UJ
Caso de Prueba: Realizar inserción de los datos de un universo juvenil insertando un carné de identidad con caracteres extraños.
Entrada: Nombre: Leidys

Segundo Nombre: Mariam
Apellido: Ramirez
Segundo Apellido: Odelin
Carné de Identidad: 03^301238*
Color de piel: Mestiza
Sexo: Femenino
Militancia: No
Nivel cultural: Secundaria Básica
Ubicación: CB Mantua
Resultado: El sistema muestra un mensaje de error.
Condiciones: El carné de identidad debe contener sólo números.

Tabla 4.14: Caso de prueba con el carné de identidad incorrecto (introduciendo caracteres extraños)

Caso de uso: Gestionar Datos UJ
Caso de Prueba: Realizar inserción de los datos de un universo juvenil dejando el campo Color de piel vacío.
Entrada: Nombre: Leidys Segundo Nombre: Mariam Apellido: Ramirez Segundo Apellido: Odelin Carné de Identidad: 03033012345

<p>Color de piel:</p> <p>Sexo: Femenino</p> <p>Militancia: No</p> <p>Nivel cultural: Secundaria Básica</p> <p>Ubicación: CB Mantua</p>
<p>Resultado: El sistema muestra un mensaje de error.</p>
<p>Condiciones: Debe llenar todos los campos obligatorios.</p>

Tabla 4.15: Caso de prueba con el campo Color de piel vacío

<p>Caso de uso: Gestionar Datos UJ</p>
<p>Caso de Prueba: Realizar inserción de los datos de un universo juvenil dejando el campo Sexo vacío.</p>
<p>Entrada:</p> <p>Nombre: Leidys</p> <p>Segundo Nombre: Mariam</p> <p>Apellido: Ramirez</p> <p>Segundo Apellido: Odelin</p> <p>Carné de Identidad: 03033012345</p> <p>Color de piel: Mestiza</p> <p>Sexo:</p> <p>Militancia: No</p> <p>Nivel cultural: Secundaria Básica</p>

Ubicación: CB Mantua
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Tabla 4.16: Caso de prueba con el campo Sexo vacío

Caso de uso: Gestionar Datos UJ
Caso de Prueba: Realizar inserción de los datos de un universo juvenil dejando el campo Militancia vacío.
Entrada: Nombre: Leidys Segundo Nombre: Mariam Apellido: Ramirez Segundo Apellido: Odelin Carné de Identidad: 03033012345 Color de piel: Mestiza Sexo: Femenino Militancia: Nivel cultural: Secundaria Básica Ubicación: CB Mantua
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Tabla 4.17: Caso de prueba con el campo Militancia vacío

Caso de uso: Gestionar Datos UJ
Caso de Prueba: Realizar inserción de los datos de un universo juvenil dejando el campo Nivel cultural vacío.
Entrada: Nombre: Leidys Segundo Nombre: Mariam Apellido: Ramirez Segundo Apellido: Odelin Carné de Identidad: 03033012345 Color de piel: Mestiza Sexo: Femenino Militancia: No Nivel cultural: Ubicación: CB Mantua
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Tabla 4.18: Caso de prueba con el campo Nivel cultural vacío

Caso de uso: Gestionar Datos UJ
Caso de Prueba: Realizar inserción de los datos de un universo juvenil dejando de seleccionar su ubicación.
Entrada: Nombre: Leidys

Segundo Nombre: Mariam
Apellido: Ramirez
Segundo Apellido: Odelin
Carné de Identidad: 03033012345
Color de piel: Mestiza
Sexo: Femenino
Militancia: No
Nivel cultural: Secundaria Básica
Ubicación:
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Tabla 4.19: Caso de prueba con el campo Ubicación vacío

Sección Modificar o Eliminar

Para realizar algunas de estas operaciones primeramente se realizará la búsqueda del universo juvenil deseado y luego se modificarán sus datos o se eliminarán del sistema, los siguientes casos de prueba están referidos a la búsqueda del universo juvenil.

Condición de Entrada	Casos válidos	Casos no válidos
Nombre	Letras y números.	No introducir el nombre o introducir caracteres extraños.

Tabla 4.20: Valores válidos e inválidos para el caso de prueba

Caso de uso: Buscar UJ
Caso de Prueba: Realizar búsqueda del universo juvenil entrando correctamente todos los datos.
Entrada: Nombre: Leidys
Resultado: El sistema muestra un listado de todos los universos juveniles que coincidan con el parámetro entrado.
Condiciones: Los datos para buscar un universo juvenil deben estar en el rango de valores válidos.

Tabla 4.21: Caso de prueba con todos los datos correctos

Caso de uso: Buscar UJ
Caso de Prueba: Realizar búsqueda del universo juvenil dejando el campo Nombre vacío.
Entrada: Nombre:
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe introducir un nombre para realizar la búsqueda.

Tabla 4.22: Caso de prueba con el campo Nombre vacío

Caso de uso: Buscar UJ
Caso de Prueba: Realizar búsqueda del universo juvenil introduciendo incorrectamente el nombre.
Entrada: Nombre: *Leidita*
Resultado: El sistema muestra un mensaje de error.
Condiciones: El nombre no puede contener caracteres extraños.

Tabla 4.23: Caso de prueba introduciendo el nombre incorrecto

4.5 Análisis de costo y beneficios

A partir de la información obtenida como resultado de la fase de inicio se tienen los conocimientos necesarios para tener una idea de lo que debe hacer el producto. Por lo tanto se puede definir qué es lo que realmente hace falta en cuanto a recursos humanos y materiales, lo que permite hacer una estimación del tiempo de duración del proyecto a través del Análisis por Puntos de Casos de Uso.

Análisis por Puntos de Casos de Uso

Se trata de 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, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

Existe una posibilidad de predecir el tamaño de un sistema a partir de las características de sus requisitos, expresados en los casos de uso.

1. Cálculo de Puntos de Casos de Uso sin ajustar.

Se calcula a partir de la siguiente ecuación:

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

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

1.1 Factor de Peso de los Actores sin ajustar (UAW)

Este valor se calcula mediante un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos. Los criterios a tener en cuenta se detallan a continuación.

Tipo de actor	Descripción	Factor de Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación.	1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2

Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3
----------	--	---

Tabla 4.24: Peso de los actores

En la presente investigación existen 5 actores en el sistema y todos constituyen personas que interactúan con el sistema mediante una interfaz gráfica por lo que su factor de peso sería 3.

Por tanto:

UAW = Cantidad de Actores * Factor de Peso

UAW = 4 * 3

UAW = 12

1.2 Factor de Peso de los Casos de Uso sin ajustar (UUCW)

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos.

Tipo de Caso de Uso	Descripción	Factor de peso
Simple	El caso de Uso contiene de 1 a 3 transacciones.	5
Medio	El caso de Uso contiene de 4 a 7 transacciones.	10
Complejo	El caso de Uso contiene más de 8 transacciones.	15

Tabla 4.25: Peso de los casos de uso

Listado de casos de uso del sistema:

1. Autenticar Usuario (2) Simple
2. Gestionar Usuario (9) Complejo
3. Gestionar Estructura (8) Complejo
4. Gestionar Datos UJ (8) Complejo
5. Gestionar Centro (7) Medio
6. Gestionar Crecimiento (12) Complejo
7. Crear Expediente (3) Simple

8. Gestionar Expediente (7) Medio
9. Realizar Alta por Crecimiento (2) Simple
10. Realizar Alta por Traslado Externo (13) Complejo
11. Realizar Alta por Ajuste (5) Medio
12. Realizar Traslado Interno (3) Simple
13. Realizar Baja Natural (3) Simple
14. Realizar Baja por Defunción (3) Simple
15. Realizar Baja por Traslado Externo (12) Complejo
16. Realizar Baja por Ajuste (3) Simple
17. Buscar Militante (2) Simple
18. Buscar UJ (2) Simple
19. Buscar Centro (2) Simple
20. Buscar Comité (2) Simple

Simples: $11 \times 5 = 55$

Medios: $3 \times 10 = 30$

Complejos: $6 \times 15 = 90$

$55 + 30 + 90 = 175$

Por tanto los Puntos de Casos de Uso sin ajustar sería:

$UUCP = UAW + UUCW$

$UUCP = 12 + 175$

$UUCP = 187$

2. Cálculo de Puntos de Casos de Uso ajustados

Que está dado por la siguiente ecuación:

$$UCP = UUCP \times TCF \times EF$$

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

Para el cálculo del Factor de complejidad técnica (TCF) se tiene en cuenta 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.

Factor	Descripción	Peso	Valor asignado	Pesoi * Valor asignadoi
T1	Sistema distribuido	2	0	0
T2	Objetivos de performance o tiempo de respuesta.	1	5	5
T3	Eficiencia del usuario final.	1	4	4
T4	Procesamiento interno complejo.	1	2	2
T5	El código debe ser reutilizable.	1	3	3
T6	Facilidad de instalación.	0.5	5	2.5
T7	Facilidad de uso.	0.5	5	2.5
T8	Portabilidad.	2	5	10
T9	Facilidad de cambio.	1	3	3
T10	Concurrencia.	1	0	0
T11	Incluye objetivos especiales de seguridad.	1	2	2
T12	Provee acceso directo a terceras partes.	1	5	5
T13	Se requieren facilidades especiales de entrenamiento a usuarios.	1	3	3

Tabla 4.26: Factores para el cálculo de la complejidad técnica

$$TCF = 0.6 + 0.01 * \Sigma (\text{Pesoi} \times \text{Valor asignadoi})$$

$$TCF = 0.6 + 0.01 * 42$$

$$TCF = 0.6 + 0.42$$

$$TCF = 1.02$$

Para el cálculo del **factor ambiente** se tienen en cuenta las habilidades y el entrenamiento del grupo involucrado en el desarrollo del sistema. El cálculo del mismo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

Factor	Descripción	Peso	Valor Asignado	Peso _i * Valor asignado _i
E1	Familiaridad con el modelo del proyecto utilizado.	1.5	3	4.5
E2	Experiencia en la aplicación.	0.5	2	1
E3	Experiencia en orientación a objetos.	1	4	4
E4	Capacidad del analista líder.	0.5	4	2
E5	Motivación.	1	3	3
E6	Estabilidad de los requerimientos.	2	2	4
E7	Personal part-time.	-1	0	0
E8	Dificultad del lenguaje de programación.	-1	2	-2

Tabla 4.27: Factores para el cálculo del factor ambiente

$$EF = 1.4 - 0.03 * \Sigma (\text{Peso}_i * \text{Valor asignado}_i)$$

$$EF = 1.4 - 0.03 * 16.5$$

$$EF = 1.4 - 0.495$$

$$EF = 0.905$$

Calculando los puntos de Caso de Uso Ajustados quedaría:

$$UCP = UUCP * TCF * EF$$

$$UCP = 187 * 1.02 * 0.905$$

$$UCP = 172.6197$$

$$UCP = 172.6197$$

3. De los Puntos de Casos de Uso a la estimación del esfuerzo

Que está dado por la siguiente ecuación:

$$E = UCP * CF$$

donde:

E: Esfuerzo estimado en horas hombres.

UCP: Punto de casos de usos ajustados.

CF: Factor de conversión.

Para calcular el **Factor de conversión** se tiene en cuenta:

- ✓ Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6 (4 factores por debajo de 3).
- ✓ 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 (0 factores por encima de 3).

Como el total contabilizado es 4 se utiliza el **factor de conversión 28 horas-hombre/Punto de Casos de Uso.**

$$E = UCP * CF$$

$$E = 172.6197 * 28$$

$$E = 4833.3516 \text{ Horas-Hombre}$$

4. Calcular el esfuerzo de todo el proyecto

Actividad	Porcentaje	Esfuerzo
Análisis	20.00 %	3222.2344
Diseño	20.00%	3222.2344
Programación	30.00 %	4833.3516
Pruebas	15.00 %	2416.6758
Sobrecarga(otras actividades)	15.00%	2416.6758
Total	100.00 %	16111.172

Tabla 4.28: Esfuerzo de todo el proyecto

El proyecto requiere (hasta las pruebas) de 2416.6758 horas-hombre para su desarrollo. Trabajando 8 horas diarias se obtiene aproximadamente 302.084475 días para el desarrollo del proyecto, por tanto, para un equipo de 2 personas trabajando 24 días al mes el proyecto tiene una duración de 6 meses aproximadamente.

Para la realización del sistema requerido se trabaja 8 horas diarias los 24 días hábiles del mes.

Si el esfuerzo total (ET) es de 16111.172 horas-hombre y por cada 192 horas se tiene 1 mes eso daría un ET = 83.91235416 mes-hombre.

Ahora:

- El salario de un desarrollador (Estipendio) es: \$ 100.00
- Cantidad de Hombre (CH): 2

- Costo Hombre-Mes (CHM):

$$\text{CHM} = \text{Estipendio} * \text{CH}$$

$$\text{CHM} = \$ 200.00$$

- Costo Total (Costo):

$$\text{Costo} = \text{ET} * \text{CHM}$$

$$\text{ET} = 83.91235416 \text{ mes-hombre} / 2$$

$$\text{ET} = 41.95617708 \text{ meses}$$

$$\text{Costo} = 41.95617708 * 200.00$$

$$\text{Costo} = \$ 8391$$

4.5.1 Beneficios

El presente trabajo, como parte de las aplicaciones que se desarrollan con el objetivo de ser usadas en el ámbito nacional, no reporta un beneficio monetario directo. Sin embargo; desde el punto de vista de los beneficios intangibles que reporta, se puede decir que con el desarrollo del módulo Militante como parte del Sistema Integral de Gestión de la UJC Nacional, se humaniza el proceso de control de información dentro del área de Vida Interna y Estado de la Militancia. Además permite un manejo más inmediato de las informaciones que se generan en esas áreas. Se evita la pérdida y duplicidad de información.

4.6 Conclusiones

Como resultado del trabajo realizado y una vez obtenidos los artefactos del flujo de trabajo de implementación se cumplen los objetivos trazados para la realización de forma exitosa de la aplicación, se refleja el estándar de codificación utilizado lo que permite un mejor entendimiento del código. Al realizarle las pruebas al sistema se puede evidenciar la calidad del producto final y tras el análisis de costos y beneficios se exponen los beneficios intangibles que aporta la realización del presente trabajo de diploma.

CONCLUSIONES GENERALES

El Módulo Militante constituye un importante aporte al desarrollo de la informatización de este país, específicamente a la Unión de Jóvenes Comunistas.

En este trabajo se demostró la necesidad de diseñar e implementar un sistema informático que permita el desarrollo de los procesos políticos que se llevan a cabo en la UJC Nacional y la interacción fácil y rápida del personal de la UJC con los datos de los militantes y los miembros del universo juvenil. La propuesta de solución introduce mejoras en dichos procesos en pos de humanizar el trabajo de los funcionarios de la organización y disminuir el tiempo de atención al personal y con ello aumentar la calidad, lo que permitirá aumentar la eficiencia de la organización.

La utilización de la aplicación en todas las direcciones de la UJC a lo largo y ancho del país permite eliminar el trabajo tedioso de los funcionarios y posibilita la seguridad de los datos con los que trabajan y facilita las búsquedas de los datos en un corto tiempo y con un mínimo de errores, además evita la necesidad de impartir cursos de capacitación al personal que se inicie en la tarea, debido a que la aplicación muestra clara y sencillamente los pasos a seguir y los datos a introducir en cada uno de los procesos.

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

- Tras el análisis de sistemas de gestión existentes con propósitos similares en el ámbito nacional e internacional se concluye no utilizar ninguno de ellos pues no cumplen con las características y necesidades de la UJC en el país, por lo que se lleva a cabo la implementación del Módulo Militante.
- La identificación de los procesos que requerían automatización arrojó resultados importantes para la determinación de los requisitos funcionales del sistema.
- La especificación de los requisitos funcionales posibilitó la definición de las funcionalidades de la aplicación informática para la gestión de información de la UJC Nacional.
- Con el análisis y diseño del Módulo Militante se obtuvo como resultado el modelo de diseño que sirvió como entrada principal para la fase de Construcción en la que se implementó el Módulo Militante como parte del Sistema Integral para la UJC Nacional aportando los beneficios expuestos anteriormente.

RECOMENDACIONES

Los objetivos generales de este trabajo fueron alcanzados, pero durante su desarrollo, han surgido ideas que serían recomendables tener en cuenta para su futuro perfeccionamiento, teniendo en cuenta las conclusiones del trabajo se recomienda:

Desarrollar nuevas funcionalidades al módulo que permitan ampliar la gestión de los procesos políticos que se desarrollan, tales como el proceso de sanciones y evaluaciones de los militantes, así como el registro de su cotización.

REFERENCIAS BIBLIOGRÁFICAS

. Disponible en: http://www.tiempo21.cu/hipertextos/union_jovenes_comunistas.htm.

ANA, V. F. UML. 2008, nº Disponible en: <http://www-gris.det.uvigo.es/~avilas/UML/node42.html>.

BUSCAR. Disponible en: http://www.bvs.sld.cu/revistas/aci/vol14_4_06/aci11406.htm>

CHOVA, G. *Manual de Informatica*. 2007, nº Disponible en:

<http://manuales.astalaweb.com/Manuales/UML.asp>

CORNEJO, J. E. G. *Arquitectura en Capas*. 2009, nº Disponible en: <http://www.docirs.cl/josego.html>.

CRUZ, F. N. LA INVESTIGACIÓN EXPLORATORIA. 2006, nº Disponible en:

<http://www.gestiopolis.com/canales7/mkt/investigacion-exploratoria-y-algunos-aportes-a-la-investigacion-de-mercados.htm>.

GADGETCABARET. *Netbeans 6.0 – Instalacion y características* Disponible en:

<http://gadgetcabaret.wordpress.com/2007/12/05/netbeans-60-instalacion-y-caracteristicas/>.

GRANADO, L. M. C. Sistema de control de versiones: SUBVERSION. 2004, nº Disponible en:

<http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=548>.

INSTITUTION, B. S. Disponible en: <http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion/>.

INTERNATIONAL, V. P. Free Download manager. 2005, nº Disponible en:

[http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)14720_p/).

JUAN PABLO GOMEZ GALLEGO, I. J. *Fundamentos de la Metodología RUP(Rational Unified Process)*. 2007

LANCER. Trabajo de Auditoria "Herramientas Case". 2004, nº Disponible en:
http://html.rincondelvago.com/herramientas-case_2.html.

LARMAN, C. *UML y PATRONES Introducción al análisis y diseño orientado a objetos*

LLANES, D. S. *SGIEPC. SISTEMA DE GESTIÓN DE INFORMACIÓN EN LA EMPRESA PROCESADORA DE CAFÉ ELADIO MACHÍN*. 2008.

MSC. NIRIA CASTILLO ARZOLA, L. A. C. V., LIC. JULIO VALENTIN SANTANA CRUZ. *Sistema automatizado para el control de los expedientes de tutoría en la Sede Universitaria de Florencia*. 2007.

NETBEANS. Disponible en: <http://www.netbeans.org/>.

ROLANDO ALFREDO HERNÁNDEZ LEÓN, S. C. G. *EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTIFICA*. Ciudad de la Habana: 2002.

SE, N. Disponible en: <http://www.xorcon.com/servicio.php>

SORIANO, F. *Diseño Orientado a Objetos Patrones GoF*. 2008.

BIBLIOGRAFÍAS

#8, U. D. L. C. I. F. Curso de Patrones y Anti patrones de Diseño. 2008, nº

ADDISON WESLEY, G. B., JAMES RUMBAUGH, IVAR JACOBSON El lenguaje unificado de Modelado.
nº

ANA, V. F. UML. 2008, nº Disponible en: <http://www-gris.det.uvigo.es/~avilas/UML/node42.html>.

BAEZ, M. C. Introducción a la Ingeniería de Software. . En 2008.

---. Fase de Inicio. Flujo de trabajo de requerimientos. En 2009.

---. Fase de Inicio. Modelo del Negocio. 2009,

BARTLE, P. Información para la gestión y gestión de la información. . 2008, nº Disponible en:
<http://www.scn.org/mpfc/modules/mon-miss.htm>.

BERZAL, F. Introducción a Java. Capítulo: Relaciones Entre Clases. Diagramas de Clases UML. 2009.

BUSCAR. Disponible en: <http://www.bvs.sld.cu/revistas/aci/vol14_4_06/aci11406.htm>

CAMPANARIO, M. L. R. Sistema de información para la gestión integral de los archivos de la junta de Andalucía. 2006, nº Disponible en:

http://www.csi.map.es/csi/tecmap/tecmap_2006/01T_PDF/sistema%20de%20informacion.pdf.

CCIA, D. Introducción a los patrones

de diseño. Algunos patrones básicos. 2009, nº Disponible en: <http://www.jtech.ua.es/j2ee/2008-2009/doc/patrones-sesion01-apuntes.pdf>.

CHOVA, G. Manual de Informatica. 2007, nº Disponible en:

<http://manuales.astalaweb.com/Manuales/UML.asp>

COMPUTACIÓN, C. N. D. Sistema de Gestión de Recursos Humanos de la UNA. . 2005, nº Disponible en:

<http://www.cnc.una.py/desoft/sgrh.html>.

CORNEJO, J. E. G. Arquitectura en Capas. 2009, nº Disponible en: <http://www.docirs.cl/josego.html>.

CRUZ, F. N. LA INVESTIGACIÓN EXPLORATORIA. 2006, nº Disponible en:

<http://www.gestiopolis.com/canales7/mkt/investigacion-exploratoria-y-algunos-aportes-a-la-investigacion-de-mercados.htm>.

DAPENA., M. M. D. D. Determinación de Actores y Casos de Uso del Negocio. . 2008, nº Disponible en:

<http://www.inf.udec.cl/~revista/ediciones/edicion8/Rbc.pdf>.

DÍAZ, M. Á. Introducción al Diseño con Patrones. . 2007, nº Disponible en: <http://www.tic.udc.es/~mad>.

DR. ARIEL DELGADO RAMOS , I. M. C. H., LIC. ALFREDO RODRÍGUEZ DRA. VIRGINIA JUNCAL

Registro Informatizado de Salud (RIS) 2009, nº Disponible en:

<http://www.informatica2007.sld.cu/Members/arielr/el-registro-informatizado-de-salud-ris-solucion-informatica-integral-para-el-sistema-nacional-de-salud/>.

FANJUL, A. Base de datos Orientada a Objetos 2006.

FERNÁNDEZ, L. A. L. V. Sistema de registro de profesionales de salud pública en Cuba. . 2009, nº

Disponible en: http://bvs.sld.cu/revistas/san/vol5_3_01/san10301.htm.

FRANCISCO JOSÉ HERNÁNDEZ, E. A. Z., WILLIAN ALEXIE REBOLONE. Desarrollo de un Motor de Persistencia para la Plataforma .Net. 2009, nº Disponible en:

<http://www.wisis.ufg.edu.sv/www.wisis/documentos/TE/004.68-F363m/004.68-F363m-Capitulo%20IV.pdf>.

GADGETCABARET. Netbeans 6.0 – Instalacion y características Disponible en:

<http://gadgetcabaret.wordpress.com/2007/12/05/netbeans-60-instalacion-y-caracteristicas/>.

GONZÁLEZ-MANET, E. Las TIC 2007, nº Disponible en: <http://www.voltairenet.org/article154982.html>.

GRACIA, J. 2005, nº Disponible en: <http://www.ingenierossoftware.com/analisisydiseno/patrones-diseno.php>.

GRAELLS, D. P. M. Las grandes aportaciones de las TIC 2008, nº Disponible en: <http://dewey.uab.es/PMARQUES/tic.htm>.

GRANADO, L. M. C. Sistema de control de versiones: SUBVERSION. 2004, nº Disponible en: <http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=548>.

IA, D. C. D. L. C. E. Introducción y patrones básicos. 2007, nº Disponible en: <http://www.jtech.ua.es/j2ee/2008-2009/doc/patrones-sesion01-traspas.pdf>.

INSTITUTION, B. S. Disponible en: <http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion/>.

INTERNATIONAL, V. P. Free Download manager. 2005, nº Disponible en: [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).

JUAN BERNARDO QUINTERO, D. M. H., ANDREA YANZA Directrices para la construcción de artefactos de persistencia en el procesode desarrollo de software. 2008, nº Disponible en: <http://revista.eia.edu.co/articulos9/articulo%206.pdf>.

JUAN PABLO GOMEZ GALLEGO, I. J. Fundamentos de la Metodología RUP(Rational Unified Process). 2007

LAGO, R. Patrón Data Access Object. 2007, nº Disponible en: <http://www.proactiva-calidad.com/java/patrones/DAO.html>.

LANCER. Trabajo de Auditoria "Herramientas Case". 2004, nº Disponible en:
http://html.rincondelvago.com/herramientas-case_2.html.

-LANE, D. F. Dao Fast -Lane. 2009, nº Disponible en: <http://www.tic.udc.es/~fbellas/teaching/is-2002-2003/Tema5Apartado5.6.5.pdf>.

LARMAN, C. UML y PATRONES Introducción al análisis y diseño orientado a objetos

LLANES, D. S. SGIEPC. SISTEMA DE GESTIÓN DE INFORMACIÓN EN LA EMPRESA PROCESADORA DE CAFÉ ELADIO MACHÍN. 2008.

MACÍAS, J. L. Á. Nuevas Tecnologías de la Programación. 2006, nº Disponible en:
http://www.uhu.es/josel_alvarez/xNvasTecnProg/material/DAO.pdf.

MONTES, D. L. R. El diseño metodológico de la investigación científica. En 2008.

---. Cómo escribir una tesis. En 2008.

MSC. NIRIA CASTILLO ARZOLA, L. A. C. V., LIC. JULIO VALENTIN SANTANA CRUZ. Sistema automatizado para el control de los expedientes de tutoría en la Sede Universitaria de Florencia. 2007.

NETBEANS. Disponible en: <http://www.netbeans.org/>.

---. ¿Qué es NetBeans? . 2009, nº Disponible en: http://www.netbeans.org/index_es.html.

NOGUERA, Y. R. Disponible en: http://www.tiempo21.cu/hipertextos/union_jovenes_comunistas.htm.

O' BRIEN, J. A. Y. I. L. Sistemas de información gerencial 2005.

PACKO. Instalación del entorno de desarrollo integrado (IDE) Eclipse. 2008, nº Disponible en:
<http://serviciosweb.unam.mx/blog/>.

PAVÓN, E. L. Visual Paradigm, una herramienta de lo más útil. 2007, nº Disponible en:
<http://slion2000.blogspot.com/2007/04/visual-paradigm-una-herramienta-de-lo.html>.

PONJUÁN DANTE, G. Sistemas de Información : Principios y Aplicaciones. 2005.

PRIETO, F. Sistemas. Patrones de diseño. En 2009.

PÝREZ, M. Aplicaciones de escritorio eficientes. . 2005, nº Disponible en:
http://www.javahispano.org/contenidos/es/aplicaciones_de_escritorio_eficientes/.

ROJAS, D. S. G. Soporte DAO (Data Access Object) y JDBC 2007, nº Disponible en:
<http://www.sicuma.uma.es/sicuma/independientes/argentina08/Badaracco/index.htm>.

---. Tecnologías Emergentes Multiplataformas. 2007, nº Disponible en:
<http://www.sicuma.uma.es/sicuma/independientes/argentina08/Badaracco/sao.htm>.

ROLANDO ALFREDO HERNÁNDEZ LEÓN, S. C. G. EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA. Ciudad de la Habana: 2002.

SE, N. Disponible en: <http://www.xorcon.com/servicio.php>

SEGURA, H. A. Los sistemas de información gerencial y el control de gestión. 2007, nº Disponible en:
<http://www.monografias.com/trabajos44/sistemas-informacion-gerencial/sistemas-informacion-gerencial2.shtml>.

SORIANO, F. Diseño Orientado a Objetos Patrones GoF. 2008.

STALLMAN, R. Por que el código abierto pierde de vista el software libre. . 2009, nº Disponible en:
<http://www.gnu.org/philosophy/open-source-misses-the-point.es.html>.

SYSTEM, G. O. Software Libre y de Código Abierto 2009, nº Disponible en:
<http://www.gnu.org/philosophy/free-sw.es.html>.

TAMAYO, I. M. Y. C. C. Y. L. W. M. Sistema automatizado de gestión de recursos humanos del Instituto Superior de Ciencias Médicas de Santiago de Cuba 2009, nº Disponible en:
http://bvs.sld.cu/revistas/san/vol5_3_01/san10301.htm.

TECNOLOGICA, G. Netbeans. 2007, nº Disponible en: <http://www.gacetatecnologica.com/node/374>.

TELLERIA, I. O. M. Sistema automatizado informativo para la comunidad de residencia estudiantil de la Universidad de Pinar del Rio "Hermanos Saiz Montes de Oca" (Saicre). 2007.

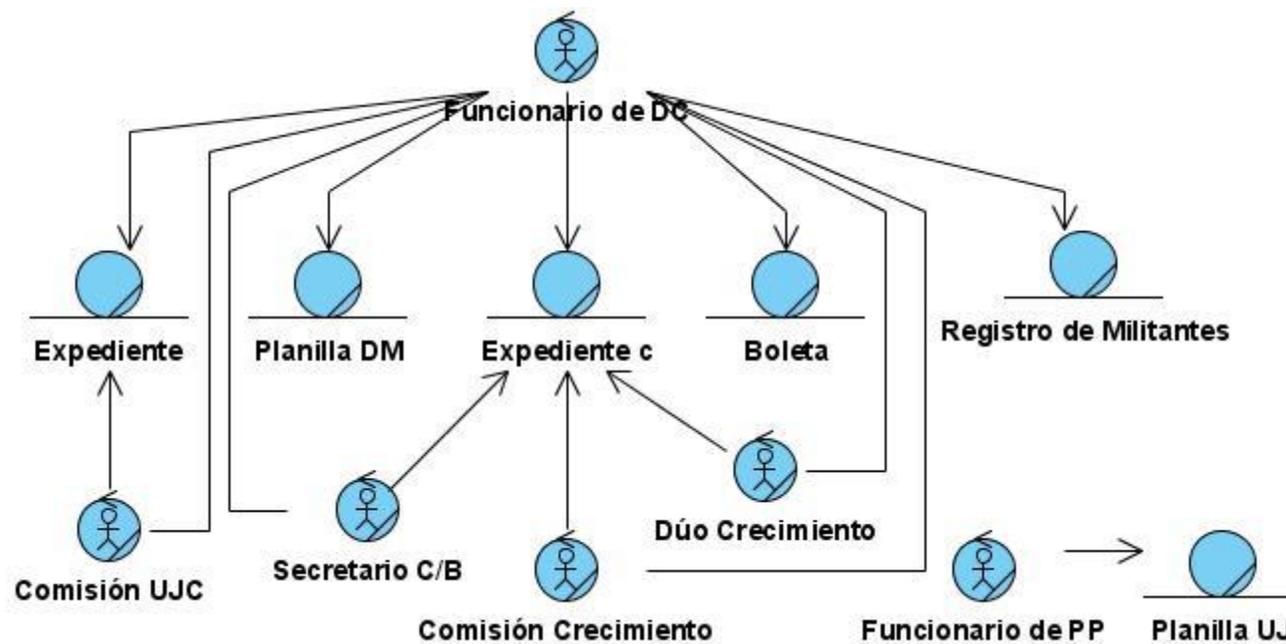
TRABAJO, O. I. D. Las TIC en procesos de aprendizaje. 2008, nº Disponible en:
<http://www.ilo.org/public/spanish/region/ampro/cinterfor/newsroom/turin/a250710.htm>.

UML, V. P. F. Visual Paradigm for UML 2007, nº Disponible en:
[http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).

XORCON. Aplicaciones de escritorio. 2007, nº Disponible en: <http://www.xorcon.com/servicio.php>

YORK, A. M. B. Introducción a los Patrones de diseño. 2007, nº Disponible en:
<http://www.portalfox.com/modules.php?op=modload&name=Sections&file=index&req=viewarticle&artid=143>.

Anexo 1. Modelo de Objeto



Anexo 2. Descripción de los CUN y sus correspondientes diagramas de actividades

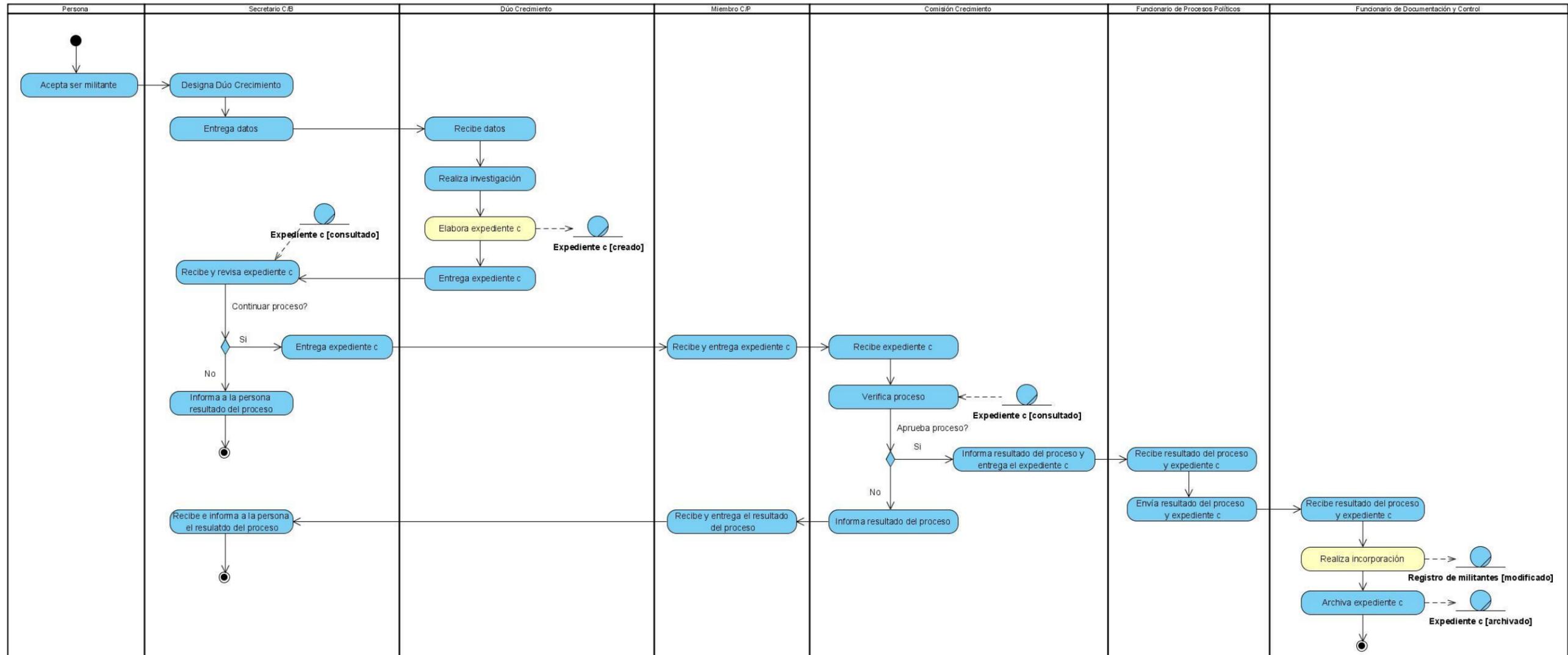
Caso de uso Realizar Crecimiento:

Caso de uso del negocio	Realizar Crecimiento
Actor	Persona
Resumen	El caso de uso inicia cuando la persona acepta ser militante. El secretario C/B selecciona un dúo de crecimiento que será el que realice las investigaciones de la persona a crecer. El resultado es entregado al secretario C/B, el cual entrega la información al miembro C/P y este la entrega a la comisión UJC que es la que verifica todo el proceso y decide si se puede crecer o no. El resultado es entregado al funcionario de documentación y control. Si en algún momento la persona se acoge al principio de voluntariedad se termina el proceso de crecimiento. Finalizando así el caso de uso.
Acción del Actor	Respuesta del proceso de negocio
1. La Persona acepta ser militante	2. El Secretario C/B designa a un Dúo de crecimiento.
	3. El Secretario C/B entrega los datos a investigar de la persona a crecer al Dúo de crecimiento.
	4. El Dúo de crecimiento recibe información, realiza investigación y elabora expediente de crecimiento.
	5.El Dúo de crecimiento entrega el expediente de crecimiento al Secretario C/B.
	6. El Secretario C/B recibe y revisa el expediente de crecimiento.

	7. Si el Secretario C/B decide continuar el proceso y entrega el expediente de crecimiento al miembro C/P.
	8. El miembro C/P recibe el expediente de crecimiento y lo entrega a la Comisión de Crecimiento.
	9. La Comisión de Crecimiento recibe el expediente de crecimiento y verifica todo el proceso.
	10. La Comisión de Crecimiento aprueba el proceso de crecimiento.
	11. La Comisión de Crecimiento informa al funcionario de procesos políticos que fue aprobado el proceso de crecimiento y le entrega el expediente de crecimiento.
	12. El funcionario de procesos políticos recibe el resultado del proceso y el expediente de crecimiento.
	13. El funcionario de procesos políticos envía el resultado del proceso y el expediente de crecimiento al funcionario de documentación y control.
	14. El funcionario de documentación y control recibe el resultado del proceso y el expediente de crecimiento y realiza la incorporación (Ver caso de uso Alta por Crecimiento).
	15. El funcionario de documentación y control archiva el expediente.
	16. Se termina el caso de uso.
Flujo Alternativo 7	
	7. El Secretario C/B decide no continuar con el proceso.

	8. Se informa a la persona que no fue aprobado el proceso.
	9. Se termina el caso de uso.
Flujo Alterno 10	
	10. La Comisión de Crecimiento no aprueba el proceso de crecimiento y envía resultado al miembro C/P.
	11. El miembro C/P recibe la información, la entrega al Secretario C/B y este informa a la persona que no fue aprobado el proceso de crecimiento.
	12. Se termina el caso de uso.

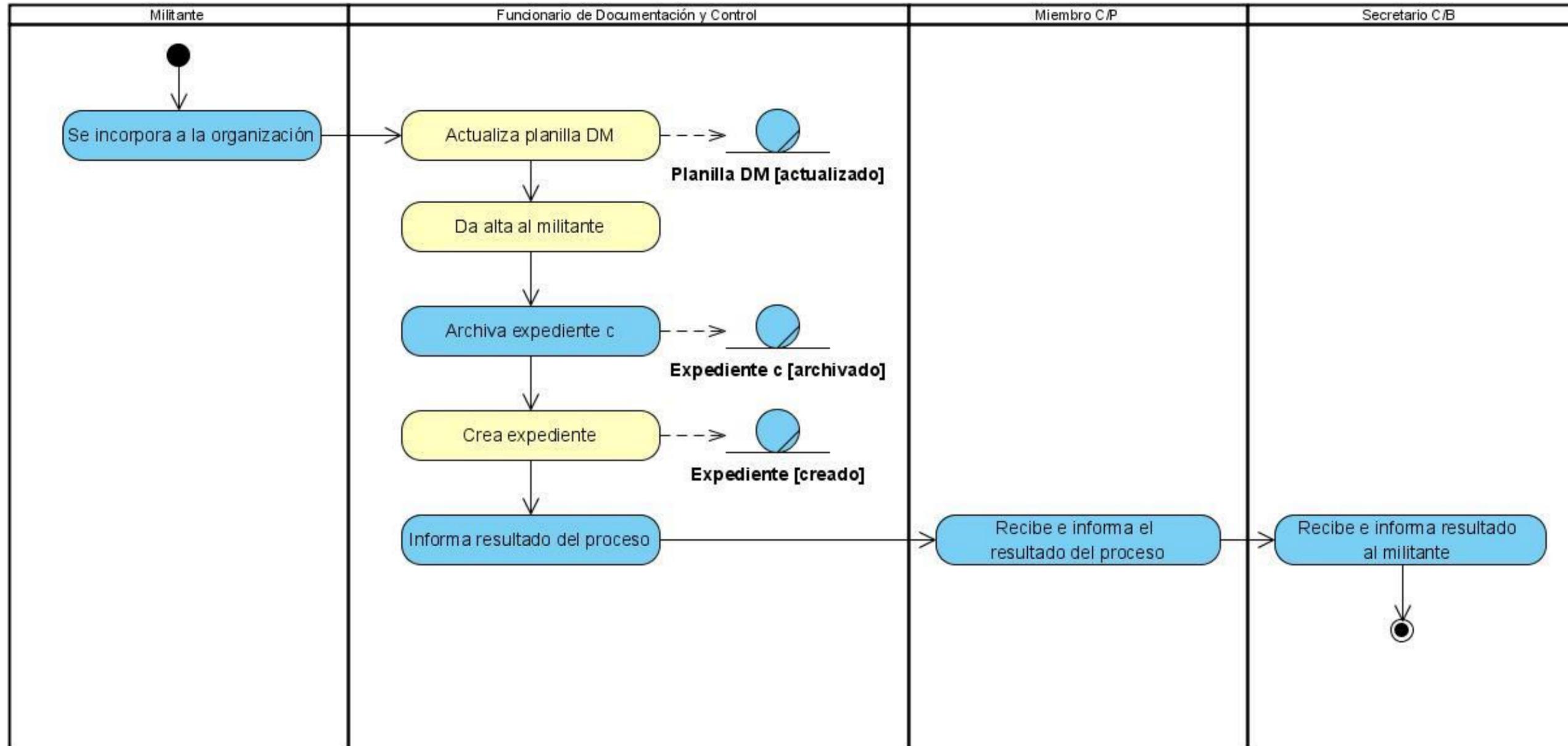
Diagrama de Actividades Caso de uso Realizar Crecimiento:



Descripción textual del CUN_ Dar Alta por Crecimiento

Caso de uso del negocio	Dar Alta por Crecimiento
Actores	Militante
Resumen	El caso de uso inicia cuando a un miembro del universo juvenil se le aprueba el proceso de crecimiento. Una vez aprobado este proceso se procede a ingresarle los datos como militante en la Planilla DM y el miembro del universo juvenil inicia su vida como militante.
Acción del actor	Respuesta del proceso de negocio
1. El miembro del universo juvenil se incorpora a la organización.	2. El funcionario de documentación y control actualiza la planilla de militante.
	3. El funcionario de documentación y control le da alta como militante, archiva el expediente de crecimiento y crea el expediente del militante.
	4. El funcionario de documentación y control informa al miembro C/P los resultados del proceso.
	5. El miembro C/P recibe información y la entrega al secretario C/B quien informa al miembro del universo juvenil que ya es militante.
	6. Se termina el caso de uso.

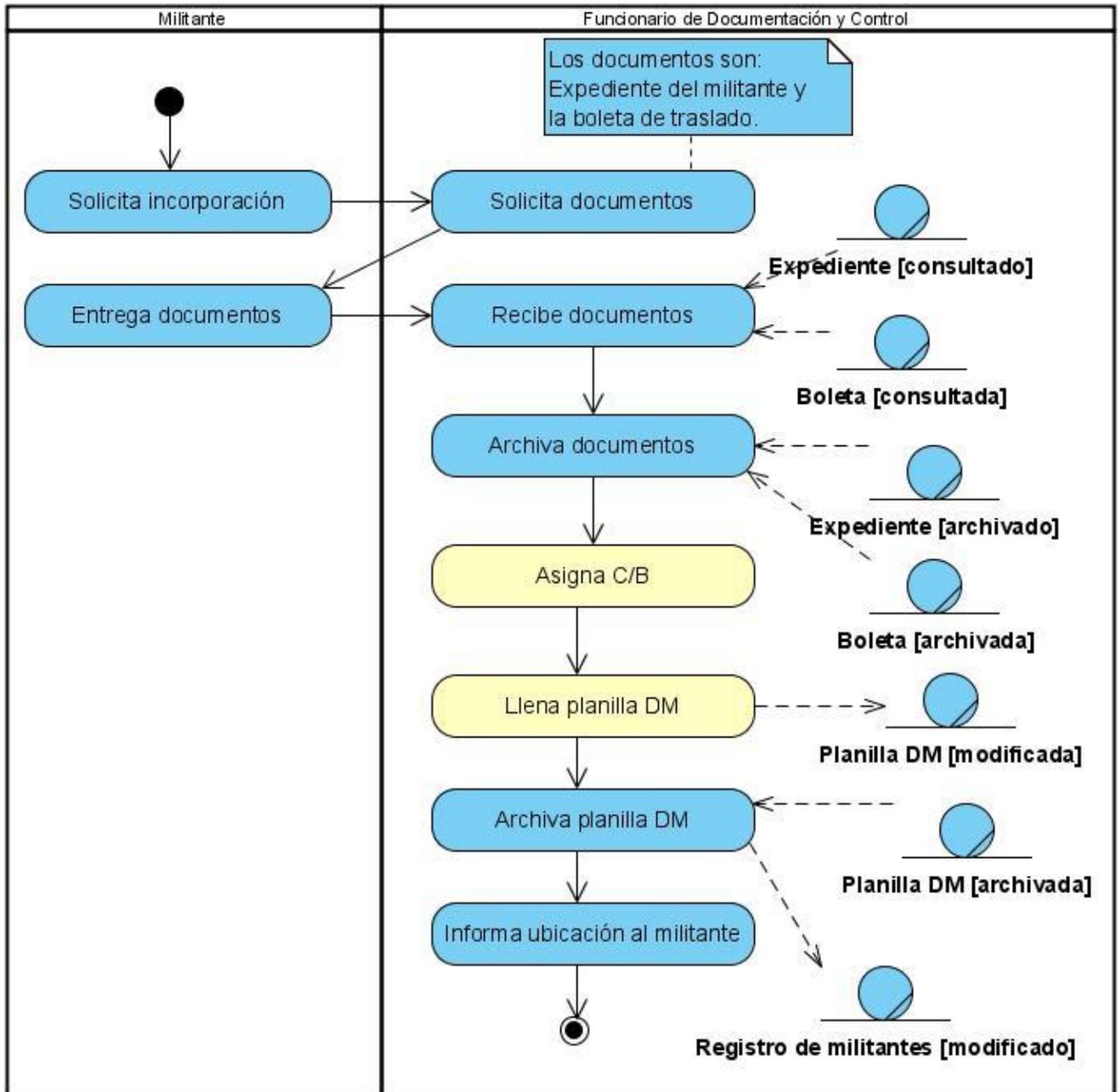
Diagrama de Actividades Caso de uso Dar Alta por Crecimiento:



Descripción textual del CUN_ Dar Alta por Traslado Externo

Caso de uso del negocio	Dar Alta por Traslado Externo
Actores	Militante
Resumen	El caso de uso inicia cuando el militante arriba a la universidad. Este debe presentarse al departamento de Documentación y Control con su expediente y su boleta.
Acción del actor	Respuesta del proceso de negocio
1. El militante solicita que lo incorporen a las filas de la UJC.	2. El funcionario de documentación y control solicita documentos (expediente y boleta).
3. El militante entrega al funcionario de documentación y control los documentos solicitados.	4. El funcionario de documentación y control recibe los documentos, los archiva y le asigna un C/B al militante.
	5. El funcionario de documentación y control llena la planilla de militante con los datos recogidos.
	6. El funcionario de documentación y control archiva la planilla de militante en el registro de militantes.
	7. El funcionario de documentación y control le informa al militante a que C/B pertenece.
	8. Se termina el caso de uso.

Diagrama de Actividades Caso de uso Dar Alta por Traslado Externo:

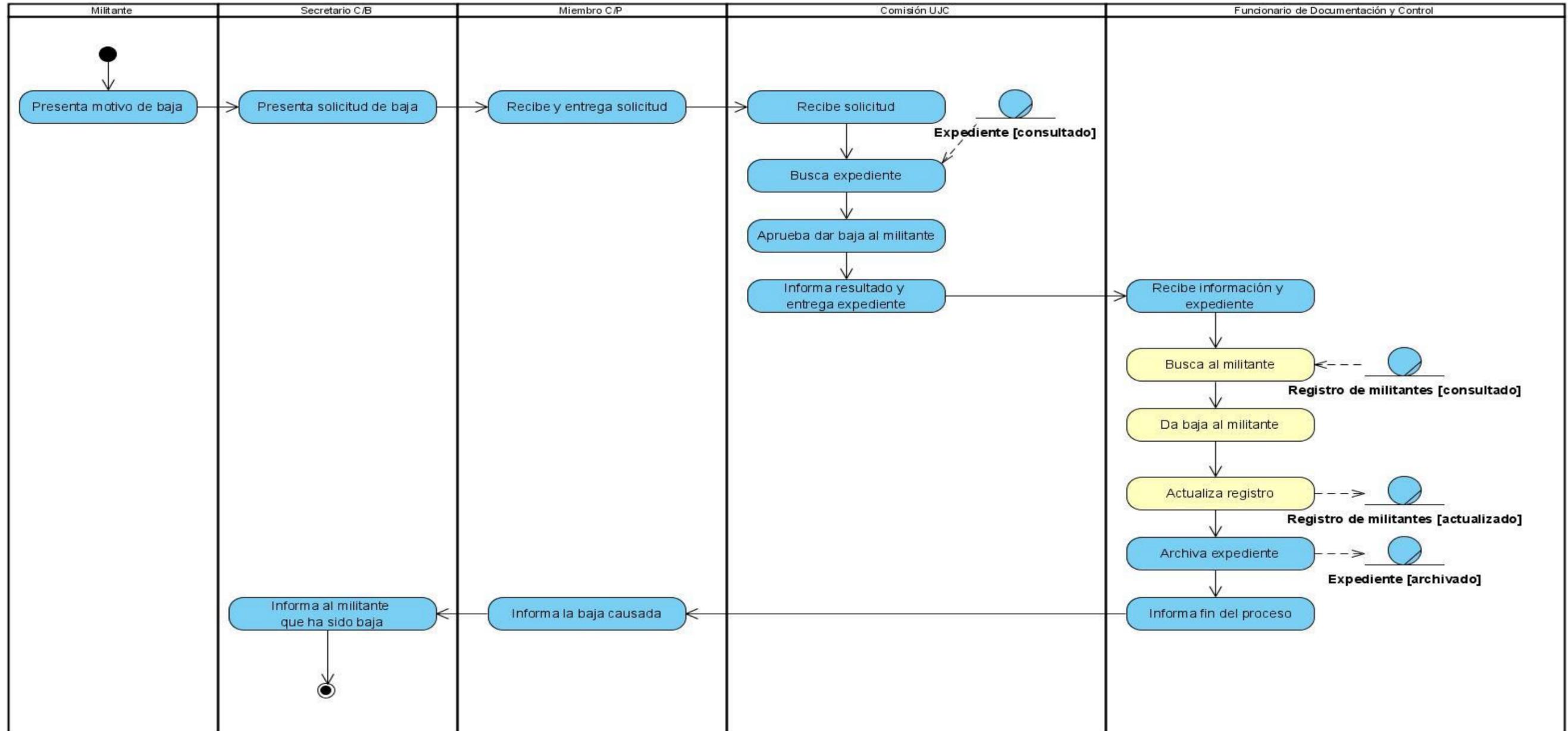


Descripción textual del CUN_ Dar Baja Natural

Caso de uso del negocio	Dar Baja Natural
Actores	Militante
Resumen	El caso de uso inicia cuando el militante presenta motivo de baja al secretario de C/B y este determina que el militante que llego a la edad límite tiene buena conducta dentro de la UJC pero presenta problemas de salud o problemas familiares que le impiden el pase al PCC y el funcionario de documentación y control le da de baja al militante.
Acción del actor	Respuesta del proceso de negocio
1. El militante presenta motivo de baja.	2. El secretario C/B presenta solicitud de baja natural para el militante al miembro C/P.
	3. El miembro C/P recibe la solicitud e informa de ella a la comisión UJC.
	4. La comisión UJC recibe la solicitud y busca el expediente del militante.
	5. La comisión UJC aprueba darle baja al militante.
	6. La comisión UJC informa al funcionario de documentación y control de la aprobación del proceso y le entrega el expediente del militante.
	7. El funcionario de documentación y control recibe la información y el expediente del militante.
	8. El funcionario de documentación y control busca al militante en el registro de militante.
	9. El funcionario de documentación y control le da de baja al militante, actualiza el registro del militante y archiva el expediente.

	10. El funcionario de documentación y control informa al miembro C/P la culminación del proceso.
	11. El miembro C/P informa al secretario C/B que el militante fue dado de baja.
	12. El secretario C/B informa a la persona que ya no es militante.
	13. Se termina el caso de uso.

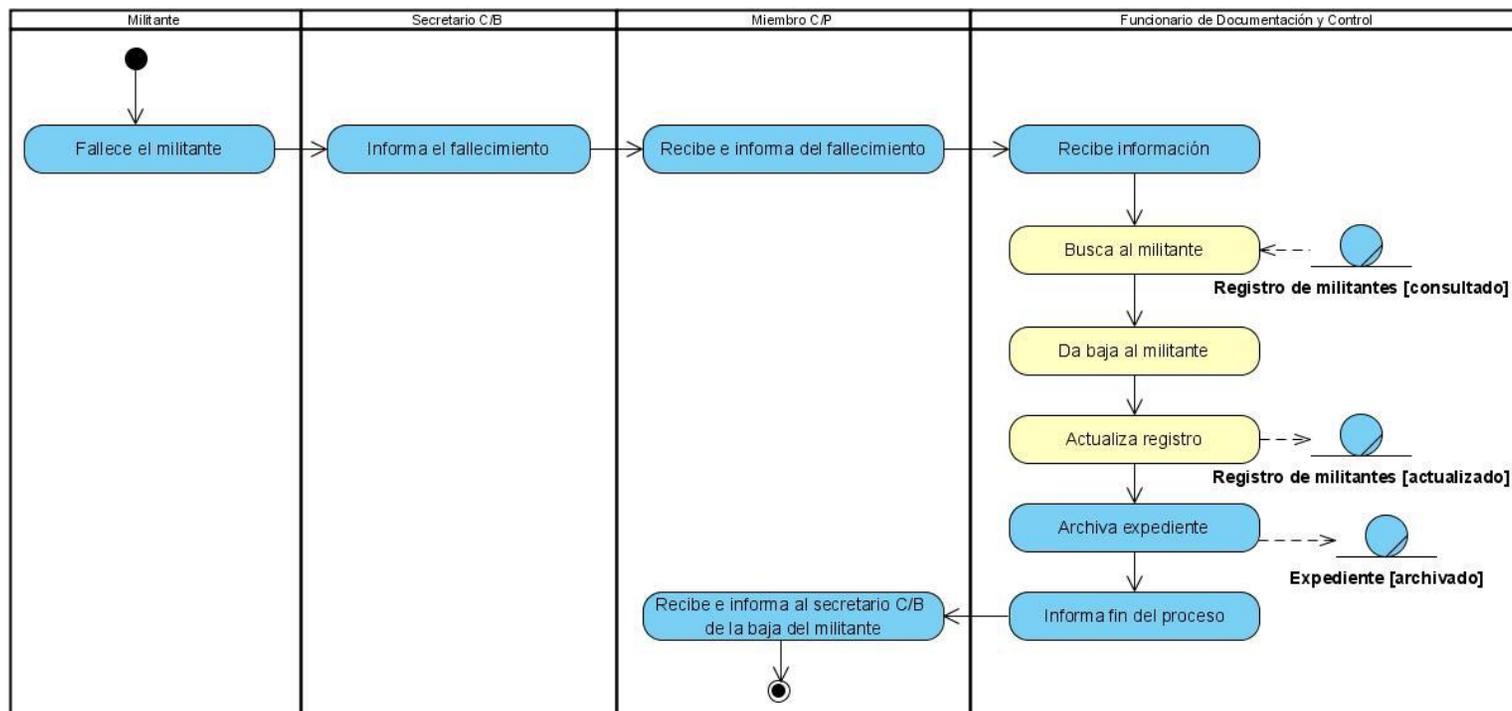
Diagrama de Actividades Caso de uso Dar Baja Natural:



Descripción textual del CUN_ Dar Baja por Defunción

Caso de uso del negocio	Dar Baja por Defunción
Actores	Militante
Resumen	El caso de uso inicia con la muerte del militante. El secretario C/B informa al miembro C/P que el militante ha fallecido y este procede a darle de baja de las filas de la UJC.
Acción del actor	Respuesta del proceso de negocio
1. El militante fallece.	2. El secretario C/B informa al miembro C/P que el militante ha fallecido.
	3. El miembro C/P informa al funcionario de documentación y control de la muerte del militante.
	4. El funcionario de documentación y control recibe información y busca al militante en el registro de militantes.
	5. El funcionario de documentación y control le da de baja al militante, actualiza el registro de militantes y archiva el expediente.
	6. El funcionario de documentación y control informa al miembro C/P la culminación del proceso.
	7. El miembro C/P recibe e informa al secretario C/B la culminación del proceso.
	8. Se termina el caso de uso.

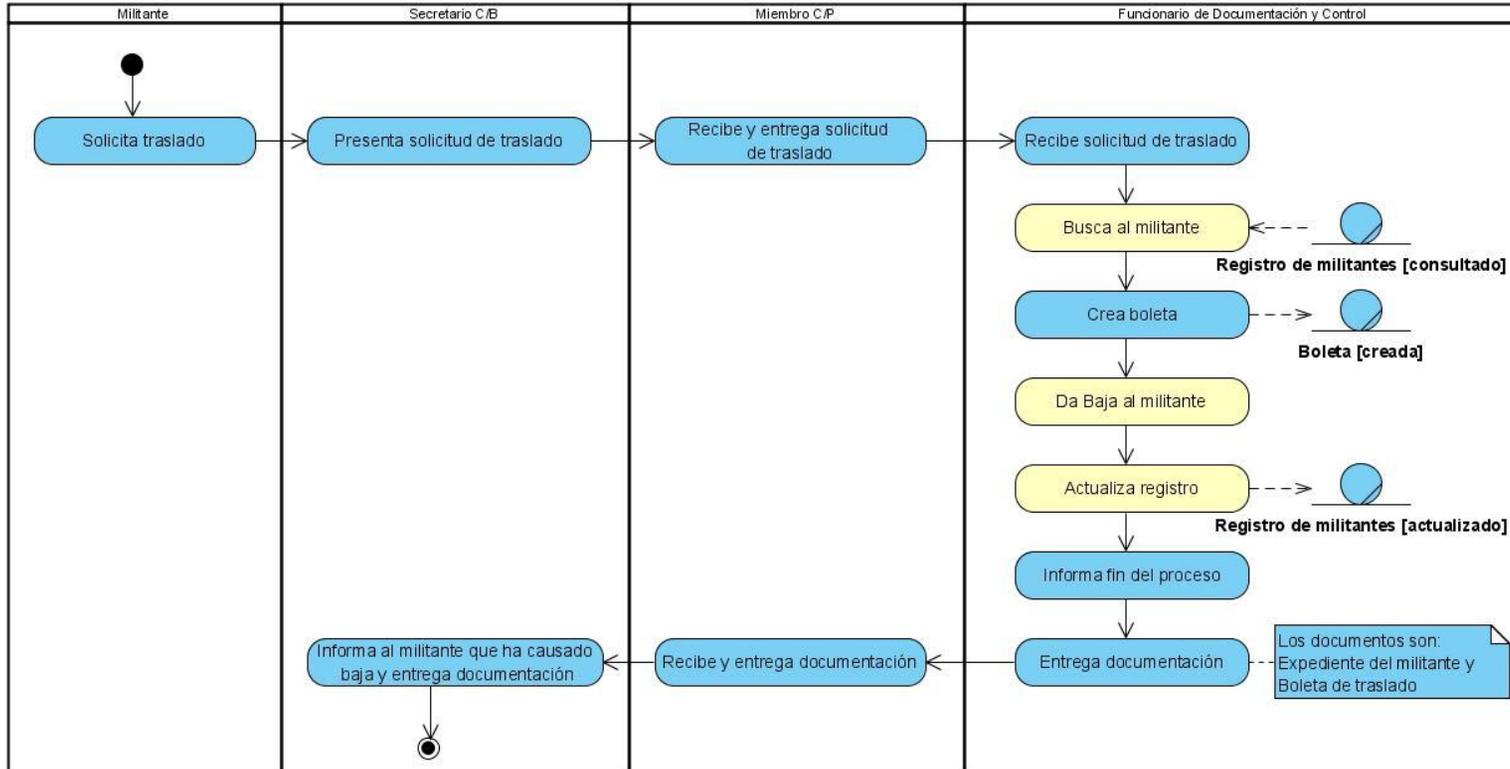
Diagrama de Actividades Caso de uso Dar Baja por Defunción:



Descripción textual del CUN_ Dar Baja por Traslado Externo

Caso de uso del negocio	Dar Baja por Traslado Externo
Actores	Militante
Resumen	El caso de uso inicia cuando el militante informa que se va de la institución. El funcionario de documentación y control le da de baja al militante de las filas de la UJC y le entrega el expediente y la boleta de traslado.
Acción del actor	Respuesta del proceso de negocio
1. El militante informa que se va de la institución y solicita un traslado externo.	2. El secretario C/B presenta al miembro C/P la solicitud de traslado externo.
	3. El miembro C/P recibe la solicitud e informa al funcionario de documentación y control sobre la solicitud del traslado externo.
	4. El funcionario de documentación y control recibe solicitud, busca al militante en el registro de militantes y crea la boleta de traslado.
	5. El funcionario de documentación y control da baja al militante y actualiza el registro de militantes.
	6. El funcionario de documentación y control informa al miembro C/P que el militante fue dado de baja y le entrega la documentación del mismo (expediente y boleta de traslado).
	7. El miembro C/P recibe e informa al secretario C/B el fin del proceso y le entrega la documentación del militante.
	8. El secretario C/B recibe e informa al militante que fue dado de baja y le entrega su documentación.
	9. Se termina el caso de uso.

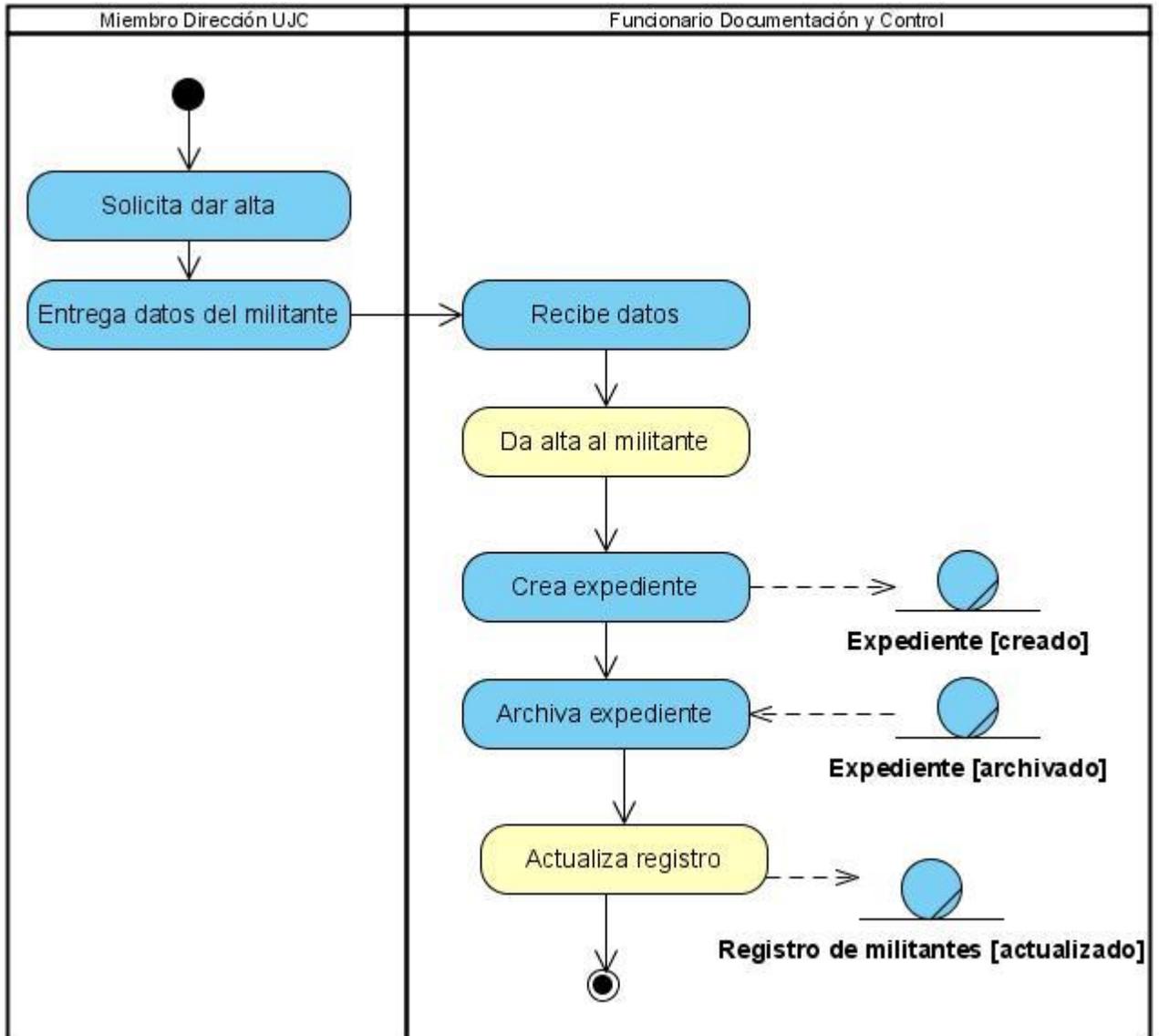
Diagrama de Actividades Caso de uso Dar Baja por Traslado Externo:



Descripción textual del CUN_ Dar Alta por Ajuste

Caso de uso del negocio	Dar Alta por Ajuste
Actores	Miembro de la Dirección UJC.
Resumen	El caso de uso inicia cuando se detectan errores de cómputo al controlar los militantes, cuando se incorpora a un militante del PCC que ostentaba la doble militancia y que por interés del PCC debe continuar con esa condición para fortalecer su C/B o cuando un militante es trasladado para un C/B y no llega a incorporarse en el mismo, regresando al C/B que lo trasladó. El funcionario de documentación y control da de alta al militante.
Acción del actor	Respuesta del proceso de negocio
1. El Miembro de la Dirección UJC informa dar alta por ajuste y entrega datos del militante.	2. El funcionario de documentación y control recibe los datos del militante.
	3. El funcionario de documentación y control da de alta al militante.
	4. El funcionario de documentación y control crea el expediente y lo archiva.
	5. El funcionario de documentación y control actualiza el registro de militante.
	6. Se termina el caso de uso.

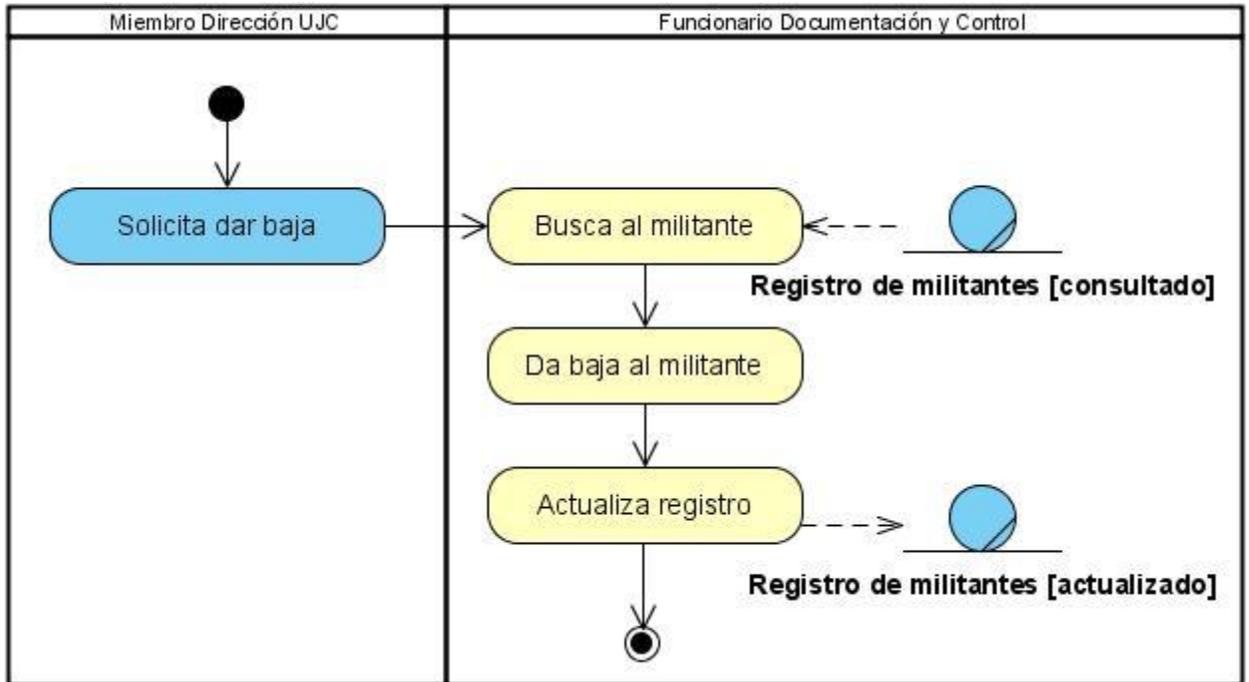
Diagrama de Actividades Caso de uso Dar Alta por Ajuste:



Descripción textual del CUN_ Dar Baja por Ajuste

Caso de uso del negocio	Dar Baja por Ajuste
Actores	Miembro de la Dirección UJC.
Resumen	El caso de uso inicia cuando el funcionario de documentación y control detecta errores de cómputo al controlar los militantes, cuando el militante que presentaba doble militancia cumple con la tarea asignada por parte del PCC y deja de pertenecer a las filas de la UJC o cuando el militante no regresa al antiguo CB y se da de alta en un nuevo comité de base. El funcionario de documentación y control da de baja al militante.
Acción del actor	Respuesta del proceso de negocio
1. El Miembro de la Dirección UJC informa dar baja por ajuste.	2. El funcionario de documentación y control busca al militante en el registro de militante.
	3. El funcionario de documentación y control da de baja al militante.
	4. El funcionario de documentación y control actualiza el registro de militante.
	5. Se termina el caso de uso.

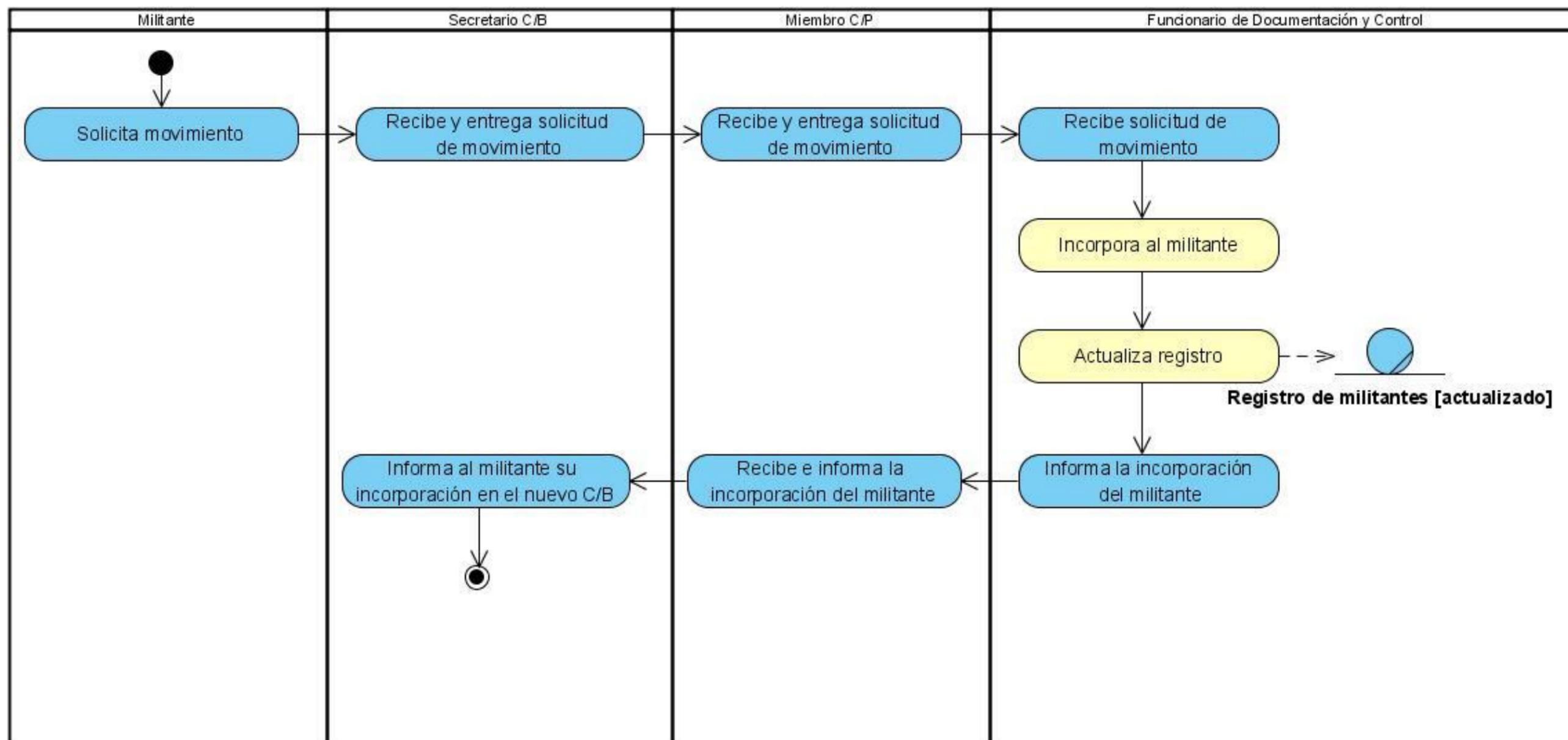
Diagrama de Actividades Caso de uso Dar Baja por Ajuste:



Descripción textual del CUN_ Dar Alta por Traslado Interno

Caso de uso del negocio	Dar Alta por Traslado Interno
Actores	Militante
Resumen	El caso de uso inicia cuando el militante solicita un movimiento. El secretario de C/B informa al miembro C/P y este a su vez al funcionario de documentación y control quien procede a darle de alta en el nuevo comité de base.
Acción del actor	Respuesta del proceso de negocio
1. El militante solicita movimiento.	2. El secretario de C/B recibe solicitud de movimiento e informa al miembro C/P.
	3. El miembro C/P recibe e informa la solicitud de movimiento al funcionario de documentación y control.
	4. El funcionario de documentación y control recibe la solicitud de movimiento, incorpora al militante en el nuevo comité de base y actualiza el registro del militante.
	5. El funcionario de documentación y control informa al miembro C/P que el militante ha sido incorporado en otro C/B.
	6. El miembro C/P recibe e informa al secretario C/B, quien le informa al militante ha sido incorporado en otro C/B.
	7. Se termina el caso de uso.

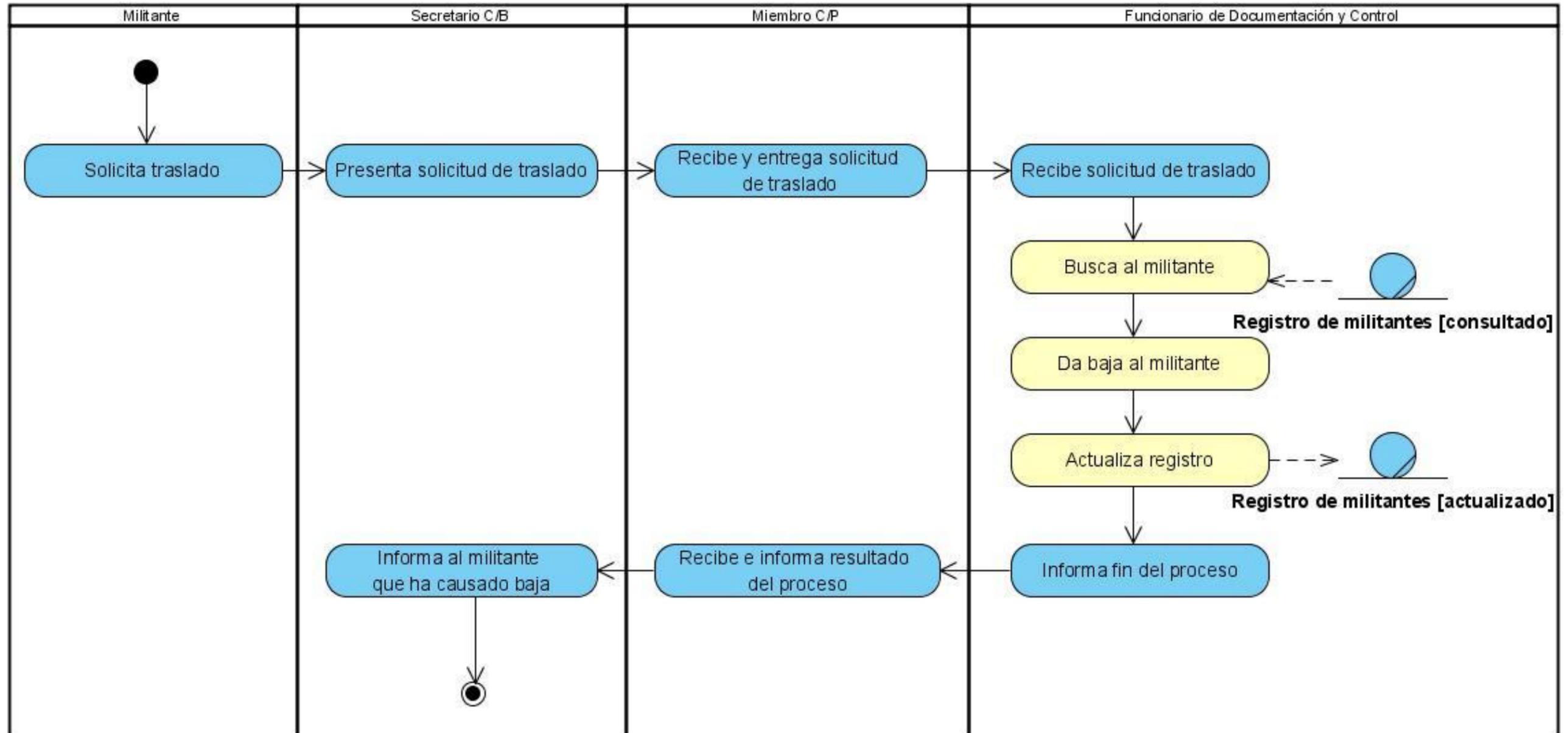
Diagrama de Actividades Caso de uso Dar Alta por Traslado Interno:



Descripción textual del CUN_ Dar Baja por Traslado Interno

Caso de uso del negocio	Dar Baja por Traslado Interno
Actores	Militante
Resumen	El caso de uso inicia cuando el militante solicita un traslado interno. El secretario C/B lo informa al miembro C/P y este a su vez al funcionario de documentación y control quien procede a efectuar la baja.
Acción del actor	Respuesta del proceso de negocio
1. El militante solicita traslado.	2. El secretario C/B presenta solicitud de traslado al miembro C/P.
	3. El miembro C/P recibe e informa al funcionario de documentación y control la solicitud de traslado.
	4. El funcionario de documentación y control recibe la solicitud y busca al militante en el registro de militantes.
	5. El funcionario de documentación y control da de baja al militante y actualiza el registro de militantes.
	6. El funcionario de documentación y control informa al miembro C/P la culminación del proceso.
	7. El miembro C/P le informa al secretario C/B la culminación del proceso.
	8. El secretario C/B le informa al militante que fue dado de baja.
	9. Se termina el caso de uso.

Diagrama de Actividades Caso de uso Dar Baja por Traslado Interno:



Anexo 3. Casos de uso por iteraciones

Ciclo	Nombre de caso de uso	Justificación de la selección
1	Gestionar Usuario	El caso de uso Gestionar Usuario permitirá crear los usuarios que trabajarán en el sistema, y a través del caso de uso Autenticar Usuario se podrá acceder a la aplicación, por su parte el caso de uso Gestionar Estructura es de vital importancia para el desarrollo del Módulo Militante, pues los militantes se agrupan en estructuras.
	Gestionar Estructura	
	Autenticar Usuario	
2	Gestionar Universo Juvenil	Son los casos de usos de prioridad crítica que responden a las funcionalidades vitales para que la organización pueda gestionar sus procesos.
	Gestionar Crecimiento	
	Dar Altas	
	Gestionar Expediente	
3	Dar Bajas	Son los casos de usos que permiten dar continuidad a los procesos de la organización y con los que se complementan las funciones del Módulo Militante.

Anexo 4. Descripciones de casos de uso del sistema

Caso de uso Gestionar Usuario:

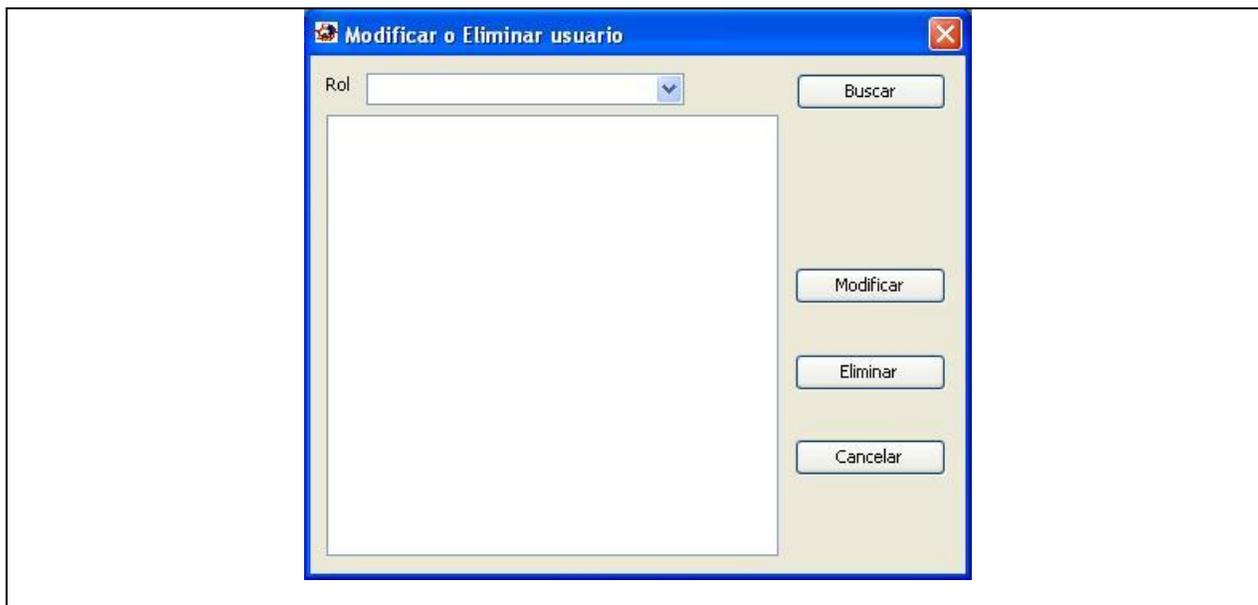
Caso de Uso:	Gestionar Usuario	
Actores:	Administrador	
Resumen:	<p>El caso de uso se inicia cuando el Administrador va a realizar alguna de las siguientes operaciones relacionadas con la gestión de los usuarios:</p> <ul style="list-style-type: none"> • Crear • Modificar • Eliminar <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.</p>	
Precondiciones:	Para crear un usuario este no debe existir y el Administrador debe haberse autenticado.	
Referencias	RF 1, RF 2, RF 3	
Prioridad	Crítica	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Administrador selecciona mediante las Acciones la opción "Usuario".	2. El sistema muestra una interfaz con las opciones Crear, Modificar o Eliminar.	
<p>3. a) Si el Administrador desea Crear un nuevo usuario ver sesión "Crear usuario".</p> <p>b) Si el Administrador desea Modificar o Eliminar un usuario existente ver sección "Modificar o Eliminar usuario".</p>		
Sección "Crear usuario"		

Acción del Actor	Respuesta del Sistema
3. El Administrador selecciona la opción Crear.	4. El sistema muestra la interfaz Crear usuario con los campos correspondientes a un usuario.
5. El Administrador llena los campos y selecciona crear el usuario mediante el botón Aceptar.	6. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos.
	7. El sistema adiciona el usuario y muestra un mensaje informando que la operación se efectuó correctamente. Finalizando así el caso de uso.
Flujo Alternativo 6	
Acción del Actor	Respuesta del Sistema
	6. El sistema detectó algún error y lo informa mediante un mensaje. Se pasa a la acción 5.
Flujo Alternativo 7	
Acción del Actor	Respuesta del Sistema
	7. No pudo efectuarse la operación pues ya existía el nombre de usuario. El sistema muestra un mensaje informando la situación al actor. Se pasa a la acción 5.
Prototipos de interfaz	

Sección “Modificar o Eliminar usuario”

Acción del Actor	Respuesta del Sistema
3. El Administrador selecciona Modificar o Eliminar usuario.	4. El sistema muestra una interfaz que le permite al Administrador buscar el usuario a través del parámetro Rol y seleccionar la opción que desee realizar.
5. El Administrador selecciona el rol del usuario que desea modificar o eliminar y presiona el botón Buscar.	6. El sistema muestra todos los usuarios que tengan el rol seleccionado por el administrador.
7. El Administrador selecciona el usuario y la opción que realizará. a) Si selecciona Modificar ver sesión “Modificar usuario”. b) Si selecciona Eliminar ver sesión “Eliminar usuario”.	

Prototipos de interfaz



Sección "Modificar usuario"

Acción del Actor	Respuesta del Sistema
7. El Administrador selecciona la opción Modificar.	8. El sistema muestra los datos del usuario dando la posibilidad de que el actor cambie los campos que desee (el nombre de usuario no puede ser cambiado).
9. El Administrador realiza los cambios en los campos que considere y los guarda mediante el botón Modificar.	10. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos.
	11. El sistema actualiza los cambios mostrando un mensaje de que todos los cambios se efectuaron satisfactoriamente. Finalizando así el caso de uso.

Flujo Alterno 10

Acción del Actor	Respuesta del Sistema
	10. El sistema encuentra algún error y lo informa mediante un mensaje. Se pasa a la acción 9.

Prototipos de interfaz

Sección "Eliminar usuario"

Acción del Actor	Respuesta del Sistema
7. El Administrador selecciona la opción Eliminar.	8. El sistema muestra un mensaje de confirmación, preguntando al administrador si está seguro de eliminar el usuario.
9. El Administrador acepta el mensaje.	10. El sistema elimina el usuario y muestra un mensaje informando que la operación tuvo éxito. Finalizando así el caso de uso.

Prototipos de interfaz

Flujo Alterno 9

Acción del Actor	Respuesta del Sistema
------------------	-----------------------

9. El Administrador no acepta el mensaje. Se pasa a la acción 6.	
Poscondiciones	Se ha agregado un usuario nuevo al sistema, modificado o eliminado uno ya existente.

Caso de uso Autenticar Usuario:

Caso de Uso:	Autenticar Usuario	
Actores:	Usuario	
Resumen:	El caso de uso inicia cuando algún Usuario desea hacer una operación en la aplicación, permite al Usuario autenticarse para tener acceso al sistema, finalizando así el caso de uso.	
Precondiciones:	Tiene el Usuario que ser usuario del sistema.	
Referencias	RF 4	
Prioridad	Crítica	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1. El Usuario solicita acceder al sistema.		2. El sistema muestra una interfaz para ingresar el nombre del usuario y su contraseña.
3. El actor introduce su usuario y contraseña y selecciona entrar mediante el botón Aceptar.		4 El sistema valida los datos y verifica que no existan campos obligatorios vacíos.
		5. El sistema muestra una interfaz según los privilegios del usuario autenticado. Finalizando así el caso de uso.
Flujo Alterno 4		
Acción del Actor		Respuesta del Sistema
		4. El sistema detecta algún error y lo informa mediante un mensaje. Se pasa a la acción 3.

Prototipos de interfaz	
	
	
Poscondiciones	El Usuario entró al sistema

Caso de uso Gestionar Estructura:

Caso de Uso:	Gestionar Estructura
Actores:	Miembro del buró UJC
Resumen:	El caso de uso se inicia cuando el Miembro del buró UJC va a realizar alguna de

	<p>las siguientes operaciones relacionadas con la gestión de las estructuras:</p> <ul style="list-style-type: none"> • Crear • Modificar • Listar <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.</p>
Precondiciones:	El usuario debe estar autenticado como Miembro del buró UJC.
Referencias	RF 5, RF 6, RF 7
Prioridad	Crítica
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Miembro del buró UJC selecciona mediante Acciones la opción Comité.	2. El sistema muestra una interfaz con las opciones crear, modificar y listar comité.
<p>3. a) Si el actor desea crear un comité ver sesión "Crear comité".</p> <p>b) Si el actor desea modificar un comité existente ver sección "Modificar comité".</p> <p>c) Si el actor desea listar un comité ver sección "Listar comité".</p>	
Sección "Crear comité"	
Acción del Actor	Respuesta del Sistema
3. El Miembro del buró UJC selecciona la opción Crear.	4. El sistema muestra la interfaz crear un nuevo comité.
5. El Miembro del buró UJC entra los datos necesarios para crear un comité nuevo y presiona el botón Crear, solicitando de esta forma la acción.	6. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos.

	7. El sistema crea el comité mostrando un mensaje informando que la operación tuvo éxito. Finalizando así el caso de uso.
Flujo Alterno 6	
Acción del Actor	Respuesta del Sistema
	6. El sistema detecta algún error y lo informa mediante un mensaje. Se pasa a la acción 5.
Flujo Alterno 7	
Acción del Actor	Respuesta del Sistema
	7. No pudo efectuarse la operación. El sistema muestra un mensaje informando la situación al actor. Se pasa a la acción 5.
Prototipos de interfaz	
	
Sección "Modificar comité"	
Acción del Actor	Respuesta del Sistema

3. El Miembro del buró UJC selecciona la opción modificar comité.	4. El sistema muestra una interfaz que permite seleccionar el tipo de comité.
5. El Miembro del buró UJC selecciona el tipo de comité que desea modificar.	6. El sistema muestra una interfaz que le permite al Miembro del buró UJC seleccionar el comité específico que desea modificar mediante el nombre.
7. El Miembro del buró UJC selecciona el comité que modificará y presiona el botón Modificar.	8. El sistema muestra la interfaz con los datos correspondientes del comité seleccionado.
9. El Miembro del buró UJC modifica los datos del comité seleccionado y presiona el botón Aceptar para continuar.	10. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos.
	11. El sistema realiza la modificación y lo informa mediante un mensaje. Finalizando así el caso de uso.
Flujo Alterno 10	
Acción del Actor	Respuesta del Sistema
	10. El sistema detecta algún error y lo informa mediante un mensaje. Se pasa a la acción 9.
Prototipos de interfaz	



	
Sección “Listar comité”	
Acción del Actor	Respuesta del Sistema
3. El Miembro del buró UJC selecciona la opción listar	4. El sistema muestra la interfaz correspondiente, permitiéndole al actor visualizar los comités creados hasta ese momento. Finalizando así el caso de uso.
Prototipos de interfaz	

Poscondiciones	Se creó un nuevo comité o se listaron o modificaron uno de los ya existentes.
-----------------------	---

Caso de Uso Buscar Militante:

Caso de Uso:	Buscar Militante
Actores:	Funcionario de Control y Documentación
Resumen:	Permite la búsqueda de un militante. El caso de uso inicia cuando el Funcionario de Control y Documentación desea buscar un militante, se introduce el parámetro de búsqueda y el sistema busca los militantes que coincidan con el parámetro entrado, el actor en cuestión selecciona el militante deseado finalizando así el caso de uso.
Precondiciones:	El Funcionario de Control y Documentación debe estar autenticado en el sistema.

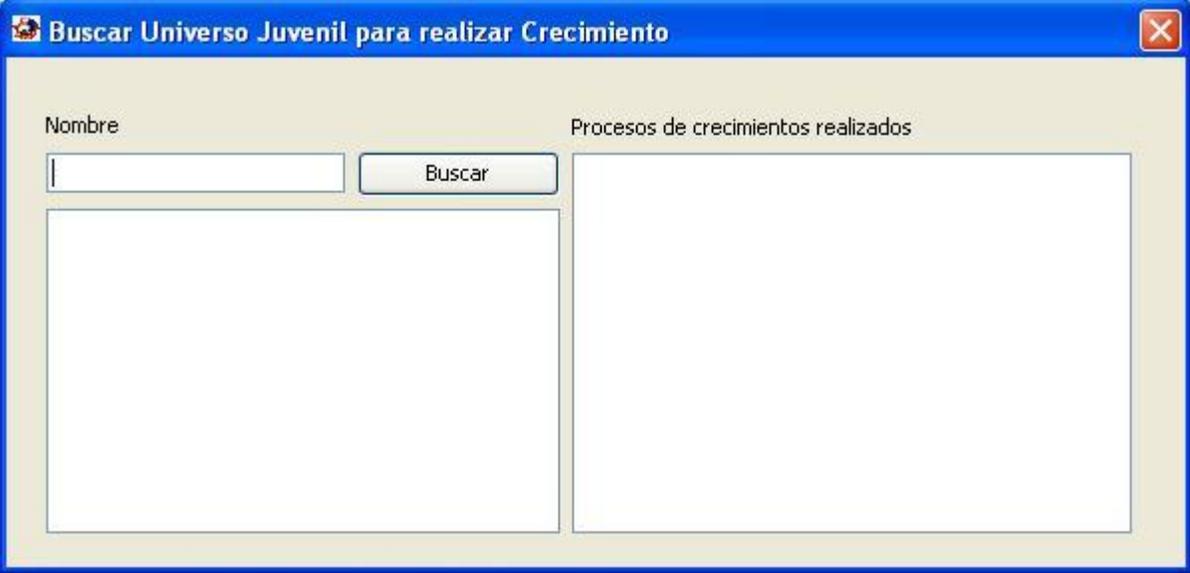
Referencias	RF 8
Prioridad	Crítica
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El actor entra el parámetro de búsqueda por el cual desea buscar y selecciona la opción Buscar.	2. El sistema muestra un listado con todos los nombres de los militantes que coinciden con el parámetro entrado.
3. El actor selecciona el militante deseado.	4. El sistema muestra la interfaz que corresponda. Finalizando así el caso de uso.
Flujo Alternativo 2	
Acción del Actor	Respuesta del Sistema
	2. La búsqueda no produjo resultado. Finalizando así el caso de uso.
Prototipos de interfaz	
	
Poscondiciones	El sistema muestra todos los militantes que coincidan con el parámetro entrado.

Caso de uso Gestionar Crecimiento:

Caso de Uso:	Gestionar Crecimiento
Actores:	Funcionario de Procesos Políticos
Resumen:	El caso de uso inicia cuando se comienza el crecimiento de un miembro del universo juvenil, en el se recogen los pasos que se siguen en el crecimiento y su fecha de cumplimiento. El caso de uso finaliza cuando son completados todos los pasos (aprobándose o no su ingreso a la UJC) o cuando se acoge al principio de voluntariedad (en este caso no ingresa a la UJC).
Precondiciones:	El Funcionario de Procesos Políticos debe estar autenticado. Para modificar o eliminar un paso, el proceso de crecimiento debe tener al menos un paso completado.
Referencias	RF 12
Prioridad	Crítica
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Funcionario de Procesos Políticos selecciona Crecimiento UJC a través del menú Vida Interna.	2. El sistema muestra la opción: Gestionar crecimiento UJC.
3. El Funcionario de Procesos Políticos selecciona la opción.	4. El sistema muestra una interfaz en la que permite buscar (a través del parámetro Nombre) el universo juvenil al cual le iniciará o modificará el proceso de crecimiento (Ver caso de uso Buscar UJ) y además le permitirá conocer si el universo juvenil tiene algún proceso de crecimiento iniciado o no.
	5. Después de seleccionado el universo juvenil (este no tiene proceso de crecimiento iniciado) el sistema muestra un mensaje informando de ello y da la posibilidad de crear un nuevo proceso.
6. El actor acepta el mensaje seleccionando crear un nuevo proceso.	7. El sistema muestra la interfaz correspondiente al primer paso del proceso de crecimiento.

8. El actor selecciona los datos y la opción guardar.	9. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos.
	10. El sistema guarda los datos y muestra un mensaje informando que la operación tuvo éxito.
11. El actor acepta el mensaje.	12. El sistema muestra una interfaz en la que muestra una tabla con los pasos completados y las opciones de modificar y agregar un paso.
13. a) Si el actor desea modificar un paso ver sección "Modificar". b) Si el actor desea agregar un paso ver sección "Agregar".	
Flujo Alterno 5	
Acción del Actor	Respuesta del Sistema
	5. Después de seleccionado el universo juvenil el sistema muestra todos los procesos de crecimiento que tiene realizados hasta el momento.
5.1 El actor selecciona el primer proceso de la lista (este sería el último en ser realizado).	5.2. El sistema muestra en la parte inferior de la misma interfaz una tabla con los pasos realizados hasta el momento y la opción de modificar cada uno de los pasos, o adicionar uno nuevo. Se pasa a la acción 12. Además de la opción de crear un nuevo proceso.
5.3. Si el actor desea crear un nuevo proceso ver sección "Crear nuevo proceso".	
Flujo Alterno 9	
Acción del Actor	Respuesta del Sistema
	9. El sistema detecta algún error y lo informa mediante un mensaje.
Sección "Agregar"	

Acción del Actor	Respuesta del Sistema
13. El actor selecciona la opción agregar paso.	14. El sistema muestra la interfaz correspondiente al segundo o al tercer paso en dependencia del paso que tenga realizado, si tiene realizado el tercer paso el sistema muestra un mensaje de error. Finalizando así el caso de uso.
Sección "Modificar"	
Acción del Actor	Respuesta del Sistema
13. El actor selecciona la opción modificar correspondiente al paso que desea modificar.	14. El sistema muestra una interfaz con los datos del paso a modificar.
15. El actor modifica los datos y selecciona la opción Aceptar.	16. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos.
	17. El sistema guarda los datos y muestra un mensaje informando el éxito de la operación.
18. El actor acepta el mensaje.	19. El sistema muestra la interfaz correspondiente a los pasos completados. Finalizando así el caso de uso.
Flujo Alterno 16	
Acción del Actor	Respuesta del Sistema
	16. El sistema detecta algún error y lo informa mediante un mensaje.
Sección "Crear nuevo proceso"	
Acción del Actor	Respuesta del Sistema
6. El actor selecciona la opción crear nuevo proceso.	7. El sistema verifica que el universo juvenil no tenga un proceso sin terminar y muestra la interfaz correspondiente al primer paso.
8. El actor introduce los datos.	9. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos.

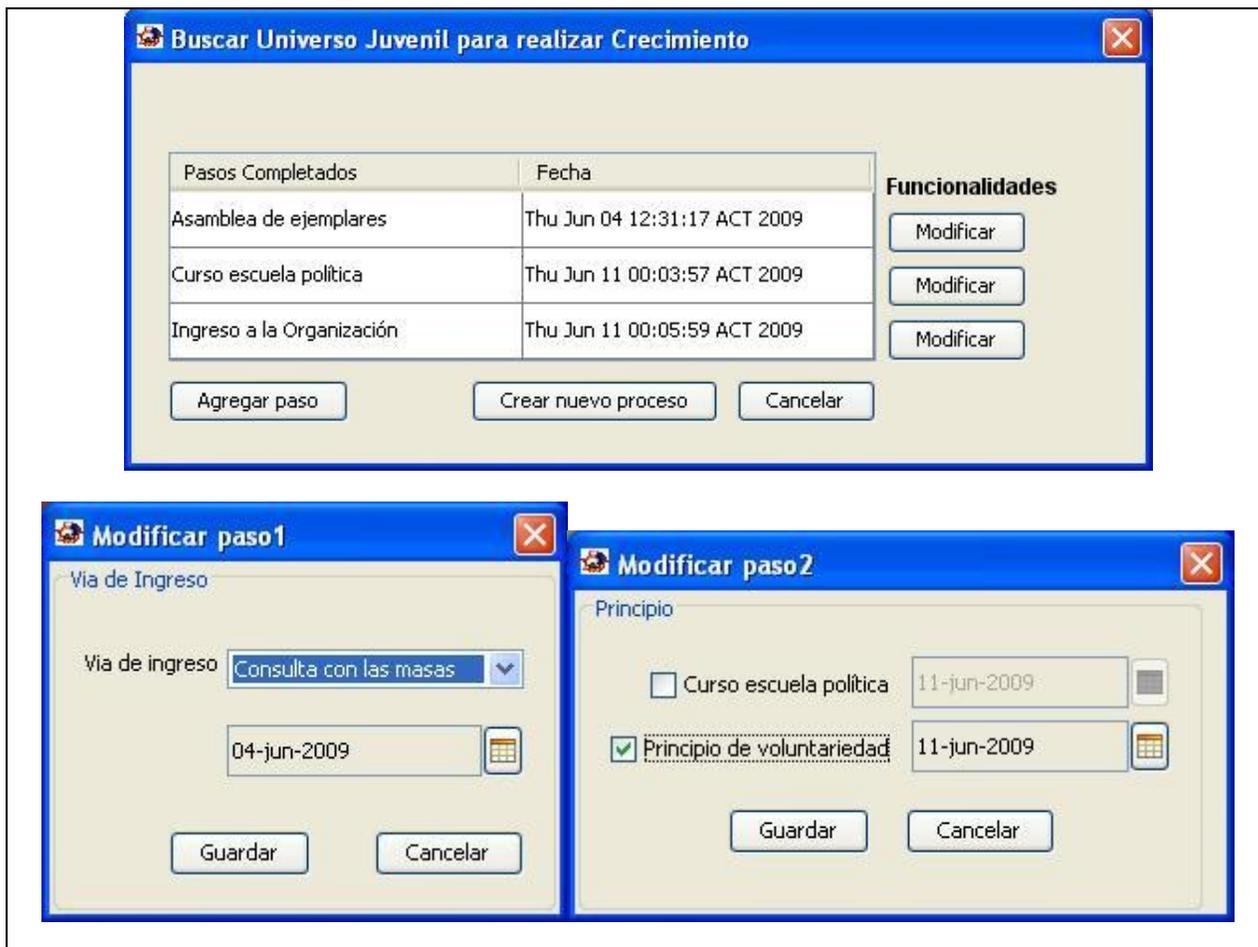
	10. El sistema guarda los datos y muestra un mensaje informando que la operación tuvo éxito. Finalizando así el caso de uso.
Flujo Alternativo 7	
Acción del Actor	Respuesta del Sistema
	7. El sistema detecta que existe un proceso no terminado y muestra un mensaje informando el error.
Flujo Alternativo 9	
Acción del Actor	Respuesta del Sistema
	9. El sistema detecta algún error y lo informa mediante un mensaje.
Prototipos de Interfaz	
	

The image displays three overlapping dialog boxes, each representing a step in a data entry process. Each dialog box has a blue title bar with a close button (X) in the top right corner.

Paso 1: Via de Ingreso
This dialog box contains two input fields. The first is a dropdown menu labeled "Via de ingreso" with the value "Asamblea de ejem...". The second is a date field labeled "Fecha de inicio del proceso" with the value "04-jun-2009". At the bottom, there are two buttons: "Guardar" and "Cancelar".

Paso 2: Principio
This dialog box contains two checked checkboxes. The first is "Curso escuelas politicas" with a date field set to "11-jun-2009". The second is "Principio de voluntariedad" with a date field set to "11-jun-2009". At the bottom, there are two buttons: "Guardar" and "Cancelar".

Paso 3: Ingreso
This dialog box contains several input fields. It starts with an unchecked checkbox "Principio de voluntariedad" and a date field "11-jun-2009". Below that is a radio button group for "Ingreso a la organización" with "Ingresa" selected. The "Fecha de ingreso" field is "11-jun-2009". The "Organizacion que realizó el proceso" field contains "CB Sandino". The "Provincia donde se realizó el proceso" field contains "Pinar del Río". The "Municipio donde se realizó el proceso" field contains "Sandino". At the bottom, there are two buttons: "Guardar" and "Cancelar".



	
Poscondiciones	Se agregó un nuevo paso, modificó uno ya existente o se inició un nuevo proceso.

Caso de Uso Crear Expediente:

Caso de Uso:	Crear Expediente
Actores:	Funcionario de Control y Documentación
Resumen:	<p>El caso de uso se inicia cuando el funcionario de Control y Documentación va a crear un expediente:</p> <ul style="list-style-type: none"> • Crear <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta la acción necesaria. El caso de uso finaliza cuando se ejecuta la acción solicitada.</p>
Precondiciones:	El funcionario de Control y Documentación debe haberse autenticado correctamente en el sistema.

Referencias	RF 13
Prioridad	Crítica
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El funcionario de Control y Documentación selecciona la opción crear expediente, mediante el vínculo de la página correspondiente.	2. El sistema muestra la primera sección del expediente correspondiente a los datos personales.
3. El actor llena los campos correspondientes y selecciona continuar con el proceso mediante el botón Siguiente.	4. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos.
	5. El sistema mostrará la siguiente sección del expediente hasta haber llenado la última sección (las siguientes secciones del expediente son: datos laborales, datos familiares, proceso de ingreso, datos de interés, asignar comité).
6. Después de haber llenado la última sección del expediente el actor selecciona la opción Finalizar.	7. El sistema guardará todos los datos. Finalizado así el caso de uso.
Flujo Alternativo 4	
Acción del Actor	Respuesta del Sistema
	4. El sistema detecta algún error y lo informa mediante un mensaje. Se pasa a la acción 3.
Prototipos de interfaz	

Expediente del militante: Datos generales

Datos Personales

* Nombre * Carnet de Identidad
 2do Nombre * Número de Carnet de Militante
 * Apellido * Color de Piel
 * 2do Apellido * Sexo

Dirección Particular

*Dirección Particular
 * Provincia
 * Municipio
 Teléfono

Otros Datos

* Nivel cultural vencido
 Ciudadanía
 Estudia
 Año que cursa
 Cumplió con el Servicio Militar
 Lugar de nacimiento

Nota: Los campos que están señalizados con * son obligatorios

Expediente del militante: Datos familiares

Todos estos campos son opcionales, para continuar de Click en Siguiete

Parentesco Parentesco
 Nombre Nombre
 2do Nombre 2do Nombre
 Apellido Apellido
 2do Apellido 2do Apellido

Expediente del militante: Datos Laborales

* Clasificador general: MILITANTES OBREROS

* Clasificador ocupacional: Militantes Obreros con Calif. Tec. Med. o Prof

Labor: cualquiera

Nota: Los campos que están señalizados con * son obligatorios

Anterior Siguiente Cancelar

Expediente del militante: Proceso de ingreso

* Vía de ingreso utilizada: Asamblea de ejemplares

* Organización que desarrolló el proceso: jTextField1

Provincia donde se realizó el proceso: Pinar del Río

Municipio donde se realizó el proceso: Sandino

Escuelas políticas cursadas: 11-jun-2009

Escuelas administrativas cursadas: 11-jun-2009

* Fecha de ingreso al UJC: 11-jun-2009

Fecha de ingreso al PCC: 11-jun-2009

Nota: Los campos que están señalizados con * son obligatorios

Anterior Siguiente Cancelar

Expediente del militante: Datos de interés ✖

Seleccione la(s) misiones internacionalistas

Venezuela - 12 días - Militar Adicionar Misión al militante

Nueva misión internacionalista Ver Detalles de la Misión

Tipo de misión Militar

Duración 12 días

País Venezuela

Fecha 11-jun-2009 

Motivo

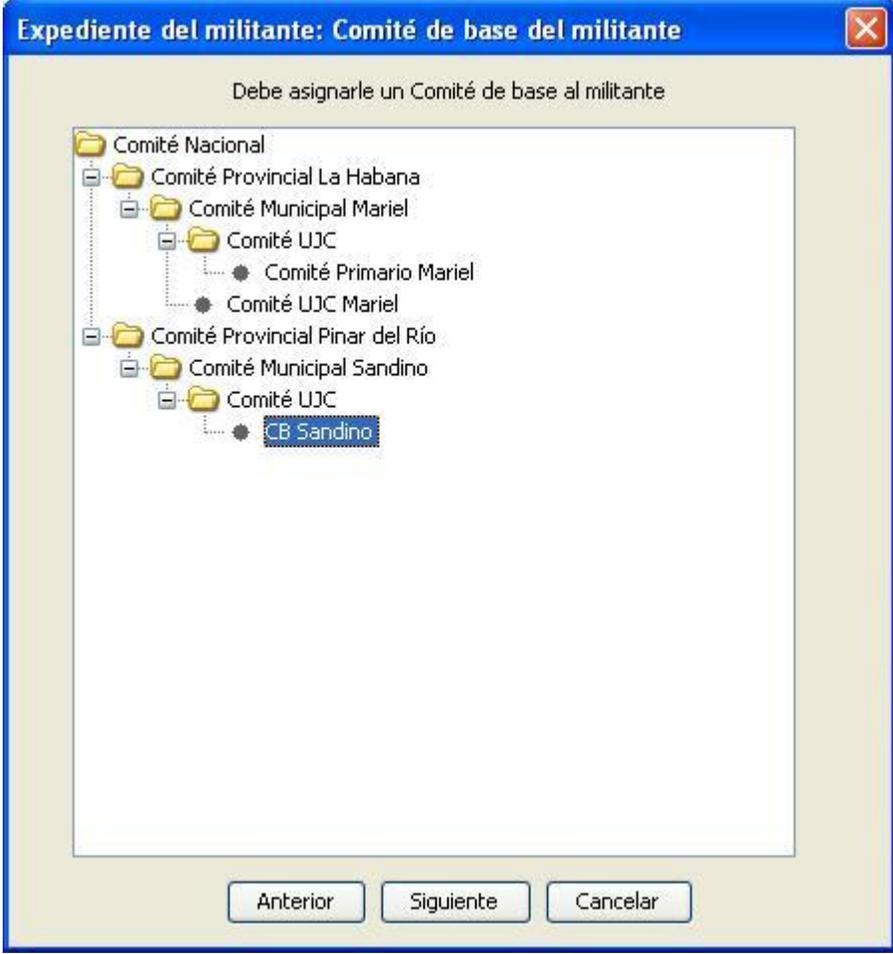
Adicionar Misión

Misiones del militante

Cargo Secretario General

Año 2007

Anterior Finalizar Cancelar

	
Poscondiciones	El expediente del militante fue creado de forma correcta.

Caso de Uso Gestionar Expediente:

Caso de Uso:	Gestionar Expediente	
Actores:	Funcionario de Control y Documentación	
Resumen:	<p>El caso de uso se inicia cuando el funcionario de Control y Documentación va a realizar alguna de las siguientes operaciones relacionadas con la gestión del expediente:</p> <ul style="list-style-type: none"> • Modificar • Eliminar <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.</p>	
Precondiciones:	El funcionario de Control y Documentación debe haberse autenticado correctamente en el sistema.	
Referencias	RF 14, RF 15	
Prioridad	Crítica	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. El funcionario de Control y Documentación selecciona Expediente a través del menú Estado de la Militancia.	2. El sistema muestra la opción Modificar o Eliminar Expediente.
	3. El funcionario de control y documentación selecciona la opción.	4. El sistema muestra una interfaz para realizar la búsqueda de los militantes que tengan expediente creado. Ver caso de uso Buscar Militante y la operación que desee realizar el actor.
		5. Después de seleccionado el militante y la opción que se realizará el sistema muestra la interfaz correspondiente.
	6. a) Si el funcionario de Control y Documentación desea modificar ver sección "Modificar Expediente".	

b) Si el funcionario de Control y Documentación desea eliminar ver sección "Eliminar Expediente".	
Sección "Modificar Expediente"	
Acción del Actor	Respuesta del Sistema
6. El funcionario de Control y Documentación selecciona la opción modificar.	7. El sistema muestra una interfaz para que el actor seleccione la sección del expediente que desea modificar.
8. El funcionario de Control y Documentación selecciona una de las secciones y continúa con la operación mediante el botón Aceptar.	9. El sistema muestra la sección del expediente seleccionada por el funcionario de Control y Documentación con los datos correspondientes.
10. El funcionario de Control y Documentación realiza los cambios pertinentes.	11. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos.
	12. El sistema realiza los cambios. Finalizando así el caso de uso.
Flujo Alterno 11	
Acción del Actor	Respuesta del Sistema
	11. El sistema encuentra algún error y lo informa mediante un mensaje. Se pasa a la acción 10.
Prototipos de interfaz	

	
Sección “Eliminar Expediente”	
Acción del Actor	Respuesta del Sistema
6. El funcionario de Control y Documentación selecciona la opción eliminar.	7. El sistema muestra un mensaje de confirmación preguntando si está seguro de eliminar el expediente.
8. El funcionario de Control y Documentación acepta el mensaje.	9. El sistema verifica que el militante seleccionado haya causado baja hace cinco años y elimina el expediente mostrando un mensaje que la operación tuvo éxito. Finalizando así el caso de uso.
Flujo Alterno 8	
Acción del Actor	Respuesta del Sistema
8. El funcionario de Control y Documentación no acepta el mensaje. Se pasa a la acción 6.	
Flujo Alterno 9	
Acción del Actor	Respuesta del Sistema
	9. El sistema verifica que el expediente no puede ser eliminado y lo informa mediante un mensaje.
9.1. El funcionario de Control y Documentación acepta el mensaje. Se pasa a la acción 6.	
Poscondiciones	El expediente del militante fue modificado o eliminado de forma correcta.

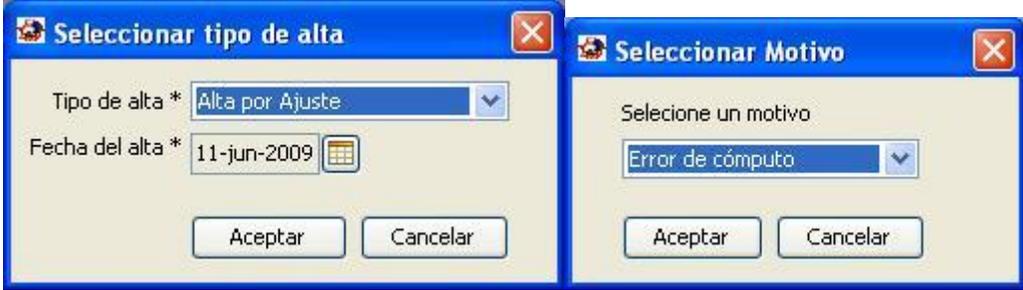
Caso de Uso Realizar Alta por Crecimiento:

Caso de Uso:	Realizar Alta por Crecimiento	
Actores:	Funcionario de Control y Documentación	
Resumen:	El caso de uso inicia cuando el funcionario de Control y Documentación selecciona realizar alta por crecimiento, el sistema permite insertar un nuevo militante en el sistema y crear el expediente correspondiente. Finalizando así el caso de uso.	
Precondiciones:	El funcionario de Control y Documentación debe estar autenticado.	
Referencias	RF 16	
Prioridad	Crítica	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. El funcionario de Control y Documentación selecciona la opción Alta a través del menú Estado de la Militancia.	2. El sistema muestra una interfaz que le permite al usuario seleccionar el tipo de alta que desee realizar.
	3. El funcionario de Control y Documentación selecciona realizar alta por crecimiento.	4. El sistema muestra una interfaz que le permite al actor buscar los miembros del universo juvenil que tengan los pasos del crecimiento completo. Ver caso de uso Buscar UJ.
	5. Después de seleccionada la persona el actor selecciona la opción Crear Expediente. Ver caso de uso Crear Expediente.	
Prototipos de interfaz		

	
Poscondiciones	El sistema dio de alta a un nuevo militante en el sistema.

Caso de Uso Realizar Alta por Ajuste:

Caso de Uso:	Realizar Alta por Ajuste
Actores:	Funcionario de Control y Documentación
Resumen:	El caso de uso inicia cuando el funcionario de Control y Documentación selecciona realizar alta por ajuste, el sistema permite realizar el alta, actualizar el expediente del militante y de ser necesario crear el expediente correspondiente. Finalizando así el caso de uso.
Precondiciones:	El funcionario de Control y Documentación debe estar autenticado.
Referencias	RF 17
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El funcionario de Control y Documentación selecciona la opción Alta a través del menú Estado de la Militancia.	2. El sistema muestra una interfaz que le permite al usuario seleccionar el tipo de alta que desee realizar.
3. El funcionario de Control y Documentación selecciona realizar alta por ajuste.	4. El sistema muestra una interfaz que le permite al actor seleccionar el tipo de ajuste por el cual se dará una nueva alta en el sistema.

<p>5. El funcionario de Control y Documentación selecciona el tipo de ajuste y selecciona la opción Aceptar para continuar con la acción.</p> <p>a) Si el actor selecciona la opción error de cómputo, ver sección “Error de Cómputo”.</p> <p>b) Si el actor selecciona la opción doble militancia, ver sección “Doble Militancia”.</p> <p>c) Si el actor selecciona la opción regreso al antiguo comité de base, ver sección “Regreso al antiguo CB”.</p>	
Prototipos de interfaz	
	
Sección “Error de Cómputo”	
Acción del Actor	Respuesta del Sistema
<p>5. El funcionario de Control y Documentación selecciona error de cómputo y la opción Aceptar. Ver caso de uso Crear Expediente.</p>	
Sección “Doble Militancia”	
Acción del Actor	Respuesta del Sistema
<p>5. El funcionario de Control y Documentación selecciona doble militancia y la opción Aceptar. Ver caso de uso Crear Expediente.</p>	
Sección “Regreso al antiguo CB”	
Acción del Actor	Respuesta del Sistema

5. El funcionario de Control y Documentación selecciona regreso al antiguo CB y la opción Aceptar.	6. El sistema muestra una interfaz que le permite al actor buscar los militantes que hallan sido dados de baja por traslado externo.
7. Después de seleccionada la persona el actor selecciona la opción Aceptar.	8. El sistema muestra la primera sección del expediente el cual debe tener todos los datos del militante.
9. El actor modificará los campos que considere necesarios.	10. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos.
	11. Guarda los datos. Finalizando así el caso de uso.
Flujo Alterno 10	
Acción del Actor	Respuesta del Sistema
	10. El sistema detecta algún error y lo informa mediante un mensaje.
Poscondiciones	El sistema dio de alta a un nuevo militante en el sistema.

Caso de Uso Realizar Traslado Interno:

Caso de Uso:	Realizar Traslado Interno
Actores:	Funcionario de Control y Documentación
Resumen:	El caso de uso inicia cuando el funcionario de Control y Documentación selecciona realizar un traslado interno, el sistema da baja al militante del comité de base en el que se encuentre militando e inmediatamente lo incorpora en un nuevo comité de base. Finalizando así el caso de uso.
Precondiciones:	El funcionario de Control y Documentación debe estar autenticado.
Referencias	RF 18
Prioridad	Crítica
Flujo Normal de Eventos	

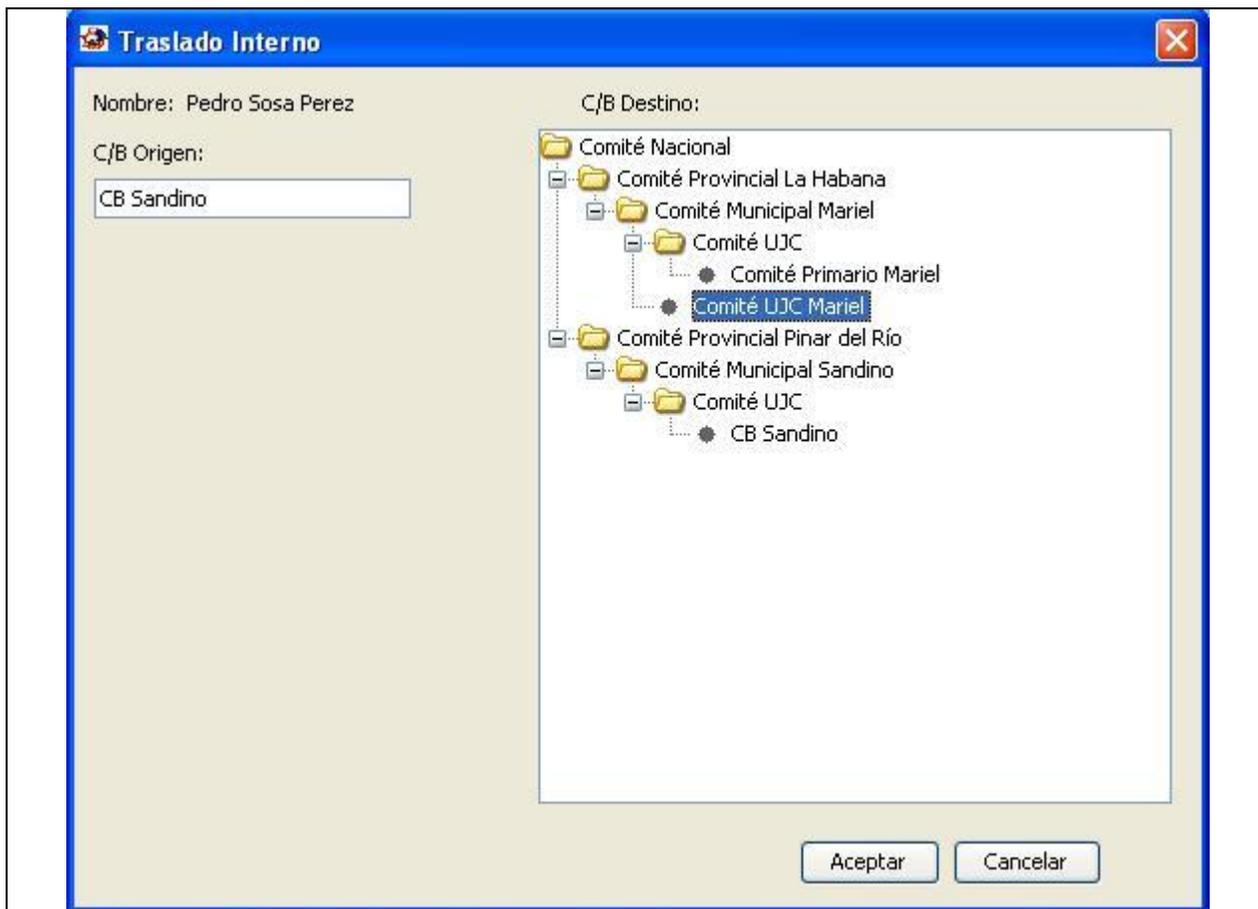
Acción del Actor	Respuesta del Sistema
1. El funcionario de Control y Documentación selecciona la opción Alta o Baja a través del menú Estado de la Militancia.	2. El sistema muestra una interfaz que le permite al usuario seleccionar el tipo de alta o baja que desee realizar.
3. El funcionario de Control y Documentación selecciona realizar traslado interno.	4. El sistema muestra una interfaz que le permite al actor buscar los militantes que coincidan con el parámetro entrado. Ver caso de uso Buscar Militante.
	5. Después de seleccionada la persona el sistema muestra una interfaz con los datos necesarios para realizar el traslado del militante.
6. El actor introduce los datos y selecciona la opción Aceptar.	7. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos.
	8. El sistema realiza el movimiento. Finalizando así el caso de uso.

Flujo Alternativo 7

Acción del Actor	Respuesta del Sistema
	7. El sistema encuentra algún error y lo informa mediante un mensaje. Se pasa a la acción 6.

Prototipos de interfaz





Poscondiciones	El sistema dio de baja a un militante de un comité de base y lo incorporó en otro.
-----------------------	--

Caso de Uso Realizar Alta por Traslado Externo:

Caso de Uso:	Realizar Alta por Traslado Externo
Actores:	Funcionario de Control y Documentación
Resumen:	El caso de uso inicia cuando el funcionario de Control y Documentación selecciona realizar alta por traslado externo, el sistema permite realizar el alta y crear el expediente correspondiente.
Precondiciones:	El funcionario de Control y Documentación debe estar autenticado.

Referencias	RF 19
Prioridad	Crítica
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El funcionario de Control y Documentación selecciona la opción Alta a través del menú Estado de la Militancia.	2. El sistema muestra una interfaz que le permite al usuario seleccionar el tipo de alta que desee realizar.
3. El funcionario de Control y Documentación selecciona realizar alta por traslado externo.	4. El sistema muestra una interfaz que le permite al actor seleccionar el tipo de traslado externo por el cual se dará una nueva alta en el sistema.
<p>5. El funcionario de Control y Documentación selecciona el tipo de traslado externo y selecciona la opción Aceptar para continuar con la acción.</p> <p>a) Si el actor selecciona la opción Provincia, ver sección “Traslado entre Provincias”.</p> <p>b) Si el actor selecciona la Municipio, ver sección “Traslado entre Municipios”.</p> <p>c) Si el actor selecciona la opción FAR, ver sección “Traslado desde las FAR”.</p> <p>d) Si el actor selecciona la opción MININT, ver sección “Traslado desde el MININT”.</p> <p>e) Si el actor selecciona la opción Exterior, ver sección “Traslado desde el Exterior”.</p>	
Prototipos de interfaz	



Sección “Traslado entre Provincias”

Acción del Actor	Respuesta del Sistema
5. El funcionario de Control y Documentación selecciona Provincia y la opción Aceptar.	6. El sistema muestra una interfaz con los datos necesarios para el traslado entre provincias.
7. El funcionario de Control y Documentación llena los campos y selecciona Aceptar para continuar con la operación.	8. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos. Para continuar con la operación ver caso de uso Crear Expediente.

Flujo Alterno 8

Acción del Actor	Respuesta del Sistema
	8. El sistema encuentra algún error y lo informa mediante un mensaje. Se pasa a la acción 7.

Sección “Traslado entre Municipios”

Acción del Actor	Respuesta del Sistema
5. El funcionario de Control y Documentación selecciona Municipio y la opción Aceptar.	6. El sistema muestra una interfaz con los datos necesarios para el traslado entre municipios.
7. El funcionario de Control y Documentación llena los campos y selecciona Aceptar para continuar con la operación.	8. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos. Para continuar con la operación ver caso de uso Crear Expediente.

Flujo Alterno 8

Acción del Actor	Respuesta del Sistema
	8. El sistema encuentra algún error y lo informa

	mediante un mensaje. Se pasa a la acción 7.
Sección “Traslado desde las FAR”	
Acción del Actor	Respuesta del Sistema
5. El funcionario de Control y Documentación selecciona FAR y la opción Aceptar.	6. El sistema muestra una interfaz con los datos necesarios para el traslado desde las FAR.
7. El funcionario de Control y Documentación llena los campos y selecciona Aceptar para continuar con la operación.	8. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos. Para continuar con la operación ver caso de uso Crear Expediente.
Flujo Alterno 8	
Acción del Actor	Respuesta del Sistema
	8. El sistema encuentra algún error y lo informa mediante un mensaje. Se pasa a la acción 7.
Sección “Traslado desde el MININT”	
Acción del Actor	Respuesta del Sistema
5. El funcionario de Control y Documentación selecciona MININT y la opción Aceptar.	6. El sistema muestra una interfaz con los datos necesarios para el traslado desde el MININT.
7. El funcionario de Control y Documentación llena los campos y selecciona Aceptar para continuar con la operación.	8. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos. Para continuar con la operación ver caso de uso Crear Expediente.
Flujo Alterno 8	
Acción del Actor	Respuesta del Sistema
	8. El sistema encuentra algún error y lo informa mediante un mensaje. Se pasa a la acción 7.
Sección “Traslado desde el Exterior”	
Acción del Actor	Respuesta del Sistema

5. El funcionario de Control y Documentación selecciona Exterior y la opción Aceptar.	6. El sistema muestra una interfaz con los datos necesarios para el traslado desde el Exterior.
7. El funcionario de Control y Documentación llena los campos y selecciona Aceptar para continuar con la operación.	8. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos. Para continuar con la operación ver caso de uso Crear Expediente.
Flujo Alterno 8	
Acción del Actor	Respuesta del Sistema
	8. El sistema encuentra algún error y lo informa mediante un mensaje. Se pasa a la acción 7.
Poscondiciones	El sistema dio de alta a un nuevo militante en el sistema.

Caso de Uso Realizar Baja por Traslado Externo:

Caso de Uso:	Realizar Alta por Traslado Externo
Actores:	Funcionario de Control y Documentación
Resumen:	El caso de uso inicia cuando el funcionario de Control y Documentación selecciona realizar baja por traslado externo, el sistema da baja al militante. Finalizando así el caso de uso.
Precondiciones:	El funcionario de Control y Documentación debe estar autenticado.
Referencias	RF 20
Prioridad	Crítica
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El funcionario de Control y Documentación selecciona la opción Baja a través del menú Estado de la Militancia.	2. El sistema muestra una interfaz que le permite al usuario seleccionar el tipo de baja que desee realizar.
3. El funcionario de Control y Documentación selecciona realizar baja por traslado externo.	4. El sistema muestra una interfaz que le permite al actor seleccionar el tipo de traslado externo por el cual se dará una baja en el

	sistema.
<p>5. El funcionario de Control y Documentación selecciona el tipo de traslado externo y selecciona la opción Aceptar para continuar con la acción.</p> <p>a) Si el actor selecciona la opción Provincia, ver sección “Traslado entre Provincias”.</p> <p>b) Si el actor selecciona la Municipio, ver sección “Traslado entre Municipios”.</p> <p>c) Si el actor selecciona la opción FAR, ver sección “Traslado desde las FAR”.</p> <p>d) Si el actor selecciona la opción MININT, ver sección “Traslado desde el MININT”.</p> <p>e) Si el actor selecciona la opción Exterior, ver sección “Traslado desde el Exterior”.</p>	
Prototipos de interfaz	
	
Sección “Traslado entre Provincias”	
Acción del Actor	Respuesta del Sistema
5. El funcionario de Control y Documentación selecciona Provincia y la opción Aceptar.	6. El sistema muestra una interfaz que le permite al actor buscar el militante. Ver caso de uso Buscar Militante.
	7. Después de seleccionada la persona el sistema muestra una interfaz con los campos necesarios para darle baja al militante.

8. El actor introduce los datos y selecciona la opción Aceptar.	9. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos y da baja al militante del sistema. Finalizando así el caso de uso.
Flujo Alterno 9	
Acción del Actor	Respuesta del Sistema
	9. El sistema encuentra algún error y lo informa mediante un mensaje. Se pasa a la acción 8.
Sección "Traslado entre Municipios"	
Acción del Actor	Respuesta del Sistema
5. El funcionario de Control y Documentación selecciona Municipio y la opción Aceptar.	6. El sistema muestra una interfaz que le permite al actor buscar el militante. Ver caso de uso Buscar Militante.
	7. Después de seleccionada la persona el sistema muestra una interfaz con los campos necesarios para darle baja al militante.
8. El actor introduce los datos y selecciona la opción Aceptar.	9. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos y da baja al militante del sistema. Finalizando así el caso de uso.
Flujo Alterno 9	
Acción del Actor	Respuesta del Sistema
	9. El sistema encuentra algún error y lo informa mediante un mensaje. Se pasa a la acción 8.
Sección "Traslado desde las FAR"	
Acción del Actor	Respuesta del Sistema
5. El funcionario de Control y Documentación selecciona FAR y la opción Aceptar.	6. El sistema muestra una interfaz que le permite al actor buscar el militante. Ver caso de uso Buscar Militante.
	7. Después de seleccionada la persona el sistema muestra una interfaz con los campos

	necesarios para darle baja al militante.
8. El actor introduce los datos y selecciona la opción Aceptar.	9. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos y da baja al militante del sistema. Finalizando así el caso de uso.
Flujo Alternativo 9	
Acción del Actor	Respuesta del Sistema
	9. El sistema encuentra algún error y lo informa mediante un mensaje. Se pasa a la acción 8.
Sección "Traslado desde el MININT"	
Acción del Actor	Respuesta del Sistema
5. El funcionario de Control y Documentación selecciona MININT y la opción Aceptar.	6. El sistema muestra una interfaz que le permite al actor buscar el militante. Ver caso de uso Buscar Militante.
	7. Después de seleccionada la persona el sistema muestra una interfaz con los campos necesarios para darle baja al militante.
8. El actor introduce los datos y selecciona la opción Aceptar.	9. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos y da baja al militante del sistema. Finalizando así el caso de uso.
Flujo Alternativo 9	
Acción del Actor	Respuesta del Sistema
	9. El sistema encuentra algún error y lo informa mediante un mensaje. Se pasa a la acción 8.
Sección "Traslado desde el Exterior"	
Acción del Actor	Respuesta del Sistema
5. El funcionario de Control y Documentación	6. El sistema muestra una interfaz que le

selecciona Exterior y la opción Aceptar.	permite al actor buscar el militante. Ver caso de uso Buscar Militante.
	7. Después de seleccionada la persona el sistema muestra una interfaz con los campos necesarios para darle baja al militante.
8. El actor introduce los datos y selecciona la opción Aceptar.	9. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos y da baja al militante del sistema. Finalizando así el caso de uso.
Flujo Alterno 9	
Acción del Actor	Respuesta del Sistema
	9. El sistema encuentra algún error y lo informa al actor mediante un mensaje. Se pasa a la acción 8.
Poscondiciones	El sistema dio de baja a un militante del sistema.

Caso de Uso Realizar Baja por Ajuste:

Caso de Uso:	Realizar Baja por Ajuste
Actores:	Funcionario de Control y Documentación
Resumen:	El caso de uso inicia cuando el funcionario de Control y Documentación selecciona realizar baja por ajuste, el sistema da baja al militante. Finalizando así el caso de uso.
Precondiciones:	El funcionario de Control y Documentación debe estar autenticado.
Referencias	RF 21
Prioridad	Crítica
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

1. El funcionario de Control y Documentación selecciona la opción Baja a través del menú Estado de la Militancia.	2. El sistema muestra una interfaz que le permite al usuario seleccionar el tipo de baja que desee realizar.
3. El funcionario de Control y Documentación selecciona realizar baja por ajuste.	4. El sistema muestra una interfaz que le permite al actor buscar los militantes con doble militancia. Se sigue el mismo procedimiento del caso de uso Buscar Militante.
	5. Después de seleccionada la persona el sistema muestra una interfaz con los campos necesarios para darle baja al militante.
6. El actor introduce los datos y selecciona la opción Aceptar.	7. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos y da baja al militante del sistema. Finalizando así el caso de uso.
Flujo Alterno 7	
Acción del Actor	Respuesta del Sistema
	7. El sistema encuentra algún error y lo informa mediante un mensaje. Se pasa a la acción 6.
Prototipos de interfaz	
	
Poscondiciones	El sistema dio de baja a un militante del sistema.

Caso de Uso Realizar Baja Natural:

Caso de Uso:	Realizar Baja Natural	
Actores:	Funcionario de Control y Documentación	
Resumen:	El caso de uso inicia cuando el funcionario de Control y Documentación selecciona realizar baja natural, el sistema da baja al militante. Finalizando así el caso de uso.	
Precondiciones:	El funcionario de Control y Documentación debe estar autenticado.	
Referencias	RF 22	
Prioridad	Crítica	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1. El funcionario de Control y Documentación selecciona la opción Baja a través del menú Estado de la Militancia.		2. El sistema muestra una interfaz que le permite al usuario seleccionar el tipo de baja que desee realizar.
3. El funcionario de Control y Documentación selecciona realizar baja natural.		4. El sistema muestra una interfaz que le permite al actor buscar los militantes que tengan la edad límite para militar en las filas de la UJC. Ver caso de uso Buscar Militante.
		5. Después de seleccionada la persona el sistema muestra una interfaz con los campos necesarios para darle baja al militante.
6. El actor introduce los datos y selecciona la opción Aceptar.		7. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos y da baja al militante del sistema. Finalizando así el caso de uso.
Flujo Alterno 7		
Acción del Actor		Respuesta del Sistema
		7. El sistema encuentra algún error y lo informa

	mediante un mensaje. Se pasa a la acción 6.
Prototipos de interfaz	
	
Poscondiciones	El sistema dio de baja a un militante del sistema.

Caso de Uso Realizar Baja por Defunción:

Caso de Uso:	Realizar Baja por Defunción
Actores:	Funcionario de Control y Documentación
Resumen:	El caso de uso inicia cuando el funcionario de Control y Documentación selecciona realizar baja por defunción, el sistema da baja al militante. Finalizando así el caso de uso.
Precondiciones:	El funcionario de Control y Documentación debe estar autenticado.
Referencias	RF 23
Prioridad	Crítica
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El funcionario de Control y Documentación selecciona la opción Baja a través del menú Estado de la Militancia.	2. El sistema muestra una interfaz que le permite al usuario seleccionar el tipo de baja que desee realizar.
3. El funcionario de Control y Documentación	4. El sistema muestra una interfaz que le permite al actor buscar los militantes que

selecciona realizar baja por defunción.	coincidan con el parámetro entrado. Ver caso de uso Buscar Militante.
	5. Después de seleccionada la persona el sistema muestra una interfaz con los campos necesarios para darle baja al militante.
6. El actor introduce los datos y selecciona la opción Aceptar.	7. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos y da baja al militante del sistema. Finalizando así el caso de uso.
Flujo Alternativo 7	
Acción del Actor	Respuesta del Sistema
	7. El sistema encuentra algún error y lo informa mediante un mensaje. Se pasa a la acción 6.
Prototipos de interfaz	
	
Poscondiciones	El sistema dio de baja a un militante del sistema.

Caso de uso Gestionar Centro:

Caso de Uso:	Gestionar Centro
Actores:	Miembro de buró UJC
Resumen:	<p>El caso de uso se inicia cuando el Miembro de buró UJC va a realizar alguna de las siguientes operaciones relacionadas con la gestión del centro.</p> <ul style="list-style-type: none"> • Crear • Modificar

	El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.	
Precondiciones:	El usuario debe estar autenticado como Miembro del buró UJC.	
Referencias	RF 24, RF 25	
Prioridad	Crítica	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1. El Miembro de buró UJC selecciona mediante las Acciones la opción "Centro".		2 El sistema muestra una interfaz con las opciones Crear y Modificar.
3. a) Si el Miembro de buró UJC desea Crear un nuevo centro ver sesión "Crear centro". b) Si el Miembro de buró UJC desea Modificar un centro existente ver sección "Modificar centro".		
Sección "Crear centro"		
Acción del Actor		Respuesta del Sistema
3. El Miembro de buró UJC selecciona la opción Crear.		4. El sistema muestra la interfaz Crear centro con los campos correspondientes.
5. El Miembro de buró UJC llena los campos y selecciona crear el centro mediante el botón Crear.		6. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos.
		7. El sistema adiciona un nuevo centro y muestra un mensaje informando que la operación se efectuó correctamente. Finalizando así el caso de uso.
Flujo Alterno 6		
Acción del Actor		Respuesta del Sistema

	6. En sistema detecta algún error y lo informa mediante un mensaje. Se pasa a la acción 5.
Prototipos de interfaz	
	
Sección "Modificar centro"	
Acción del Actor	Respuesta del Sistema
3. El Miembro de buró UJC selecciona la opción Modificar.	4. El sistema muestra una interfaz que le permite al actor seleccionar el tipo de centro.
5. El actor selecciona el tipo de centro.	6. El sistema muestra todos los nombres de centros del tipo de centro seleccionado.
7. El Miembro de buró UJC selecciona el centro y presiona el botón Modificar.	8. El sistema muestra una interfaz con los datos del centro seleccionado.
9. El Miembro del buró UJC realiza los cambios en los campos que desee.	10. El sistema verifica que sean válidos los datos escritos en cada campo y que no existan campos obligatorios vacíos.
	11. El sistema actualiza los cambios mostrando un mensaje informando que la operación tuvo éxito. Finalizando así el caso de uso.
Flujo Alterno 10	
Acción del Actor	Respuesta del Sistema
	10. El sistema encuentra algún error y lo informa mediante un mensaje. Se pasa a la acción 9.
Prototipos de interfaz	

	
<p>Poscondiciones</p>	<p>Se ha agregado un centro nuevo al sistema o modificado uno ya existente.</p>

Caso de Uso Buscar UJ:

<p>Caso de Uso:</p>	<p>Buscar UJ</p>
<p>Actores:</p>	<p>Funcionario de Procesos Políticos.</p>
<p>Resumen:</p>	<p>Permite la búsqueda de un miembro del universo juvenil. El caso de uso inicia cuando el actor desea buscar un miembro del universo juvenil, se introduce el parámetro de búsqueda y el sistema busca los miembros del universo juvenil que coincidan con el parámetro entrado, el actor en cuestión selecciona el miembro del universo juvenil deseado finalizando así el caso de uso.</p>

Precondiciones:	El Funcionario de Procesos Políticos debe estar autenticado en el sistema.	
Referencias	RF 26	
Prioridad	Crítica	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1. El actor entra el parámetro de búsqueda por el cual desea buscar y selecciona la opción Buscar.		2. El sistema muestra un listado con todos los nombres de los miembros del universo juvenil que coinciden con el parámetro entrado.
3. El actor selecciona el miembro del universo juvenil deseado.		4. El sistema muestra la interfaz que corresponda. Finalizando así el caso de uso.
Flujo Alterno 2		
Acción del Actor		Respuesta del Sistema
		2. La búsqueda no produjo resultados. Finalizando así el caso de uso.
Poscondiciones	El sistema muestra todos los miembros del universo juvenil que coincidan con el parámetro entrado.	

Caso de Uso Buscar Centro:

Caso de Uso:	Buscar Centro
Actores:	Miembro del Buró UJC.
Resumen:	Permite la búsqueda de un centro. El caso de uso inicia cuando el Miembro del Buró UJC desea buscar un centro, se introduce el parámetro de búsqueda y el sistema busca los centros que coincidan con el parámetro entrado, el actor en cuestión selecciona el centro deseado finalizando así el caso de uso.
Precondiciones:	El Miembro del Buró UJC debe estar autenticado en el sistema.
Referencias	RF 27
Prioridad	Crítica

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El actor entra el parámetro de búsqueda por el cual desea buscar y selecciona la opción Buscar.	2. El sistema muestra un listado con todos los nombres de los centros que coinciden con el parámetro entrado.
3. El actor selecciona el centro deseado.	4. El sistema muestra la interfaz que corresponda. Finalizando así el caso de uso.
Flujo Alternativo 2	
Acción del Actor	Respuesta del Sistema
	2. La búsqueda no produjo resultados. Finalizando así el caso de uso.
Poscondiciones	El sistema muestra todos los centros que coincidan con el parámetro entrado.

Caso de Uso Buscar Comité:

Caso de Uso:	Buscar Comité
Actores:	Miembro del Buró UJC.
Resumen:	Permite la búsqueda de un comité. El caso de uso inicia cuando el Miembro del Buró UJC desea buscar un comité, se introduce el parámetro de búsqueda y el sistema busca los comité que coincidan con el parámetro entrado, el actor en cuestión selecciona el comité deseado finalizando así el caso de uso.
Precondiciones:	El Miembro del Buró UJC debe estar autenticado en el sistema.
Referencias	RF 28
Prioridad	Crítica
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

1. El actor entra el parámetro de búsqueda por el cual desea buscar y selecciona la opción Buscar.	2. El sistema muestra un listado con todos los nombres de los comités que coinciden con el parámetro entrado.
3. El actor selecciona el comité deseado.	4. El sistema muestra la interfaz que corresponda. Finalizando así el caso de uso.
Flujo Alterno 2	
Acción del Actor	Respuesta del Sistema
	2. La búsqueda no produjo resultados. Finalizando así el caso de uso.
Poscondiciones	El sistema muestra todos los comités que coincidan con el parámetro entrado.

Anexo 5. Descripción de las clases del diseño

Nombre: VialIngreso	
Tipo de clase: Nomencladora	
Atributo	Tipo
idVialIngreso	Integer
nombre	String
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: VialIngresoDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarVialIngreso(VialIngreso vialIngreso)
Nombre:	obtenerTodasViasIngreso()
Nombre:	actualizarVialIngreso(VialIngreso vialIngreso)
Nombre:	obtenerVialIngreso(Integer idVialIngreso)
Nombre:	eliminarVialIngreso(VialIngreso vialIngreso)

Nombre: VialIngresoDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarVialIngreso(VialIngreso vialIngreso)
Descripción:	Guarda en la base de datos una vía de ingreso dada.
Nombre:	obtenerTodasViasIngreso()
Descripción:	Obtiene un listado de todas las vías de ingreso existentes en la base de datos.
Nombre:	actualizarVialIngreso(VialIngreso vialIngreso)
Descripción:	Actualiza los datos de una vía de ingreso dada.
Nombre:	obtenerVialIngreso(Integer idVialIngreso)
Descripción:	Obtiene una vía de ingreso dado su identificador.
Nombre:	eliminarVialIngreso(VialIngreso vialIngreso)
Descripción:	Elimina de la base de datos una vía de ingreso dada.

Nombre: ProcesoIngreso	
Tipo de clase: Entidad	
Atributo	Tipo
idProcesoIngreso	String
persona	Persona
municipio	Municipio
vialIngreso	VialIngreso
fechaIngresoUJC	Date
organizacionDesarrolloProc	String

escuelaPoliticaCurs	Date
escAdm	Date
insertado	Boolean
actualizado	Boolean
eliminado	Boolean
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: ProcesoIngresoDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	obtenerTodosProcesosIngresoPorPersonas()
Nombre:	obtenerTodasFechasIngresoPorPersonas()

Nombre: ProcesoIngresoDaoImpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarProcesoIngreso(ProcesoIngreso procesoIngreso)
Descripción:	Guarda en la base de datos un proceso de ingreso dado.

Nombre:	obtenerTodosProcesosIngreso()
Descripción:	Obtiene un listado de todos los procesos de ingreso existentes en la base de datos.
Nombre:	actualizarProcesoIngreso(ProcesoIngreso procesoIngreso)
Descripción:	Actualiza los datos de un proceso de ingreso dado.
Nombre:	obtenerProcesoIngreso(String idProcesoIngreso)
Descripción:	Obtiene un proceso de ingreso dado su identificador.
Nombre:	eliminarProcesoIngreso(ProcesoIngreso procesoIngreso)
Descripción:	Elimina de la base de datos un proceso de ingreso dado.

Nombre: Municipio	
Tipo de clase: Nomencladora	
Atributo	Tipo
idMunicipio	Integer
municipio	String
provincia	Provincia
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: MunicipioDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarMunicipio(Municipio municipio)
Nombre:	obtenerTodosMunicipios()
Nombre:	obtenerTodosMunicipiosProvincia(Integer idProvincia)
Nombre:	actualizarMunicipio(Municipio municipio)
Nombre:	obtenerMunicipio(Integer idMunicipio)
Nombre:	obtenerProvinciaPertenece(Integer idMunicipio)
Nombre:	obtenerTodosMunicipiosProvincia(String provincia)
Nombre:	eliminarMunicipio(Municipio municipio)

Nombre: MunicipioDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarMunicipio(Municipio municipio)
Descripción:	Guarda en la base de datos un municipio dado.
Nombre:	obtenerTodosMunicipios()
Descripción:	Obtiene un listado de todos los municipios existentes en la base de datos.
Nombre:	obtenerTodosMunicipiosProvincia(Integer idProvincia)
Descripción:	Obtiene un listado de todos los municipios pertenecientes a una provincia dada.
Nombre:	actualizarMunicipio(Municipio municipio)

Descripción:	Actualiza los datos de un municipio dado.
Nombre:	obtenerMunicipio(Integer idMunicipio)
Descripción:	Obtiene un municipio dado su identificador.
Nombre:	obtenerProvinciaPertenece(Integer idMunicipio)
Descripción:	Obtiene la provincia a la que pertenece un municipio dado el identificador de este.
Nombre:	obtenerTodosMunicipiosProvincia(String provincia)
Descripción:	Obtiene dado el nombre de una provincia un listado con los municipios de esta.
Nombre:	eliminarMunicipio(Municipio municipio)
Descripción:	Elimina de la base de datos un municipio dado.

Nombre: Provincia	
Tipo de clase: Nomencladora	
Atributo	Tipo
idProvincia	Integer
provincia	String
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: ProvinciaDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarProvincia(Provincia provincia)
Nombre:	obtenerTodasProvincias()
Nombre:	actualizarProvincia(Provincia provincia)
Nombre:	obtenerProvincia(Integer idProvincia)
Nombre:	eliminarProvincia(Provincia provincia)

Nombre: ProvinciaDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarProvincia(Provincia provincia)
Descripción:	Guarda en la base de datos una provincia dada.
Nombre:	obtenerTodasProvincias()
Descripción:	Obtiene un listado de todas las provincias existentes en la base de datos.
Nombre:	actualizarProvincia(Provincia provincia)
Descripción:	Actualiza los datos de una provincia dada.
Nombre:	obtenerProvincia(Integer idProvincia)
Descripción:	Obtiene una provincia dado su identificador.
Nombre:	eliminarProvincia(Provincia provincia)
Descripción:	Elimina de la base de datos una provincia dada.

Nombre: AnnoCursa	
Tipo de clase: Nomencladora	
Atributo	Tipo
idAnnoCursa	Integer
nombre	String
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: AnnoCursaDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarAnno(AnnoCursa anno)
Nombre:	obtenerTodosAnnos()
Nombre:	actualizarAnno(AnnoCursa anno)
Nombre:	obtenerAnno(Integer idAnno)
Nombre:	eliminarAnno(AnnoCursa anno)

Nombre: AnnoCursaDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarAnno(AnnoCursa anno)
Descripción:	Guarda el año que cursa el militante y que es pasado por parámetro en la base de datos.
Nombre:	obtenerTodosAnnos()
Descripción:	Obtiene una lista de todos los años existentes en la base de datos.
Nombre:	actualizarAnno(AnnoCursa anno)
Descripción:	Actualiza los datos del año que cursa el militante, este dato es pasado por parámetro.
Nombre:	obtenerAnno(Integer idAnno)
Descripción:	Obtiene un año dado su identificador.
Nombre:	eliminarAnno(AnnoCursa anno)
Descripción:	Elimina de la base de datos un año dado.

Nombre: Direccion	
Tipo de clase: Entidad	
Atributo	Tipo
idDireccion	String
direccionParticular	String (Contiene la especificación de la dirección (calle, entre calle, número....))
municipio	Municipio
Para cada responsabilidad:	

Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: DireccionDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarDireccion(Direccion direccion)
Nombre:	obtenerTodasDirecciones()
Nombre:	actualizarDireccion(Direccion direccion)
Nombre:	obtenerDireccion(String idDireccion)

Nombre: DireccionDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarDireccion(Direccion direccion)
Descripción:	Guarda una dirección dada en la base de datos.
Nombre:	obtenerTodasDirecciones()
Descripción:	Obtiene una lista de todas las direcciones existentes en la base de datos.
Nombre:	actualizarDireccion(Direccion direccion)
Descripción:	Actualiza los datos de una dirección dada.

Nombre:	obtenerDireccion(String idDireccion)
Descripción:	Obtiene una dirección dado su identificador.

Nombre: Rama	
Tipo de clase: Nomencladora	
Atributo	Tipo
idRama	Integer
nombre	String
sector	Sector
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: RamaDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarRama(Rama rama)
Nombre:	obtenerTodasRamas()
Nombre:	actualizarRama(Rama rama)
Nombre:	obtenerRama(Integer idRama)
Nombre:	eliminarRama(Rama rama)

Nombre:	obtenerTodasRamasSector(Integer idSector)
Nombre:	obtenerSectorPertenece(Integer idRama)

Nombre: RamaDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarRama(Rama rama)
Descripción:	Guarda en la base de datos una rama dada.
Nombre:	obtenerTodasRamas()
Descripción:	Obtiene un listado de todas las ramas existentes en la base de datos.
Nombre:	actualizarRama(Rama rama)
Descripción:	Actualiza los datos de una rama dada.
Nombre:	obtenerRama(Integer idRama)
Descripción:	Obtiene una rama dado su identificador.
Nombre:	eliminarRama(Rama rama)
Descripción:	Elimina de la base de datos una rama dada.
Nombre:	obtenerTodasRamasSector(Integer idSector)
Descripción:	Obtiene un listado de todas las ramas que pertenecen a un sector dado el identificador de este último.
Nombre:	obtenerSectorPertenece(Integer idRama)
Descripción:	Obtiene el sector al que pertenece una rama dado el identificador de esta última.

Nombre: Sector	
Tipo de clase: Nomencladora	
Atributo	Tipo
idSector	Integer
nombre	String
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: SectorDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarSector(Sector sector)
Nombre:	obtenerTodosSectores()
Nombre:	actualizarSector(Sector sector)
Nombre:	obtenerSector(Integer idSector)
Nombre:	eliminarSector(Sector sector)

Nombre: SectorDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarSector(Sector sector)
Descripción:	Guarda en la base de datos un sector dado.
Nombre:	obtenerTodosSectores()
Descripción:	Obtiene un listado de todos los sectores existentes en la base de datos.
Nombre:	actualizarSector(Sector sector)
Descripción:	Actualiza los datos de un sector dado.
Nombre:	obtenerSector(Integer idSector)
Descripción:	Obtiene un sector dado su identificador.
Nombre:	eliminarSector(Sector sector)
Descripción:	Elimina de la base de datos un sector dado.

Nombre: Centro	
Tipo de clase: Entidad	
Atributo	Tipo
idCentro	String
nombre	String
tipoCentro	TipoCentro
insertado	Boolean
actualizado	Boolean
eliminado	Boolean

Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: CentroDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarCentro(Centro centro)
Nombre:	obtenerTodosCentros()
Nombre:	actualizarCentro(Centro centro)
Nombre:	obtenerCentro(String idCentro)
Nombre:	eliminarCentro(Centro centro)
Nombre:	obtenerCentrosPorTipo(TipoCentro tipoCentro)

Nombre: CentroDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarCentro(Centro centro)
Descripción:	Guarda un centro dado en la base de datos.
Nombre:	obtenerTodosCentros()

Descripción:	Obtiene un listado de todos los centros existentes en la base de datos.
Nombre:	actualizarCentro(Centro centro)
Descripción:	Actualiza los datos de un centro dado.
Nombre:	obtenerCentro(String idCentro)
Descripción:	Obtiene un centro dado su identificador.
Nombre:	eliminarCentro(Centro centro)
Descripción:	Elimina de la base de datos un centro dado.
Nombre:	obtenerCentrosPorTipo(TipoCentro tipoCentro)
Descripción:	Obtiene una lista de los centros pertenecientes a un tipo especificado.

Nombre: Ciudadania	
Tipo de clase: Nomencladora	
Atributo	Tipo
idCiudadania	Integer
nombre	String
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.
Descripción:	

Nombre: CiudadaniaDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarCiudadania(Ciudadania ciudadania)
Nombre:	obtenerTodasCiudadanias()
Nombre:	actualizarCiudadania(Ciudadania ciudadania)
Nombre:	obtenerCiudadania(Integer idCiudadania)
Nombre:	eliminarCiudadania(Ciudadania ciudadania)

Nombre: CiudadaniaDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarCiudadania(Ciudadania ciudadania)
Descripción:	Guarda una ciudadanía dada en la base de datos.
Nombre:	obtenerTodasCiudadanias()
Descripción:	Obtiene un listado de todas las ciudadanía existentes en la base de datos.
Nombre:	actualizarCiudadania(Ciudadania ciudadania)
Descripción:	Actualiza los datos de una ciudadanía dada.
Nombre:	obtenerCiudadania(Integer idCiudadania)
Descripción:	Obtiene una ciudadanía dado su identificador.
Nombre:	eliminarCiudadania(Ciudadania ciudadania)
Descripción:	Elimina de la base de datos una ciudadanía dada.

Nombre: TipoCentro	
Tipo de clase: Nomencladora	
Atributo	Tipo
idTipoCentro	Integer
tipoCentro	String
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: TipoCentroDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarTipoCentro(TipoCentro tipoCentro)
Nombre:	obtenerTodosTiposCentro()
Nombre:	actualizarTipoCentro(TipoCentro tipoCentro)
Nombre:	obtenerTipoCentro(Integer idTipoCentro)
Nombre:	eliminarTipoCentro(TipoCentro tipoCentro)

Nombre: TipoCentroDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarTipoCentro(TipoCentro tipoCentro)
Descripción:	Guarda en la base de datos un tipo de centro dado.
Nombre:	obtenerTodosTiposCentro()
Descripción:	Obtiene un listado de todos los tipos de centros existentes en la base de datos.
Nombre:	actualizarTipoCentro(TipoCentro tipoCentro)
Descripción:	Actualiza los datos de un tipo de centro dado.
Nombre:	obtenerTipoCentro(Integer idTipoCentro)
Descripción:	Obtiene un tipo de centro dado su identificador.
Nombre:	eliminarTipoCentro(TipoCentro tipoCentro)
Descripción:	Elimina de la base de datos un tipo de centro dado.

Nombre: Mision	
Tipo de clase: Entidad	
Atributo	Tipo
idMision	String
tipoMision	TipoMision
fecha	Date
pais	String
motivo	String
duracion	String

insertado	Boolean
actualizado	Boolean
eliminado	Boolean
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: MisionDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarMision(Mision mision)
Nombre:	obtenerTodasMisiones()
Nombre:	actualizarMision(Mision mision)
Nombre:	obtenerMision(Integer idMision)

Nombre: MisionDaoImpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarMision(Mision mision)
Descripción:	Guarda una misión dada en la base de datos.

Nombre:	obtenerTodasMisiones()
Descripción:	Obtiene una lista de todas las misiones existentes en la base de datos.
Nombre:	actualizarMision(Mision mision)
Descripción:	Actualiza los datos de una misión dada.
Nombre:	obtenerMision(Integer idMision)
Descripción:	Obtiene una misión dado su identificador.

Nombre: Telefono	
Tipo de clase: Entidad	
Atributo	Tipo
idTelefono	String
numero	String (es único)
descripcion	String
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: PeridoDesactivacion	
Tipo de clase: Entidad	
Atributo	Tipo
idPeriodoDesactivacion	String

idComite	String
Desactivacion	Date
Activacion	Date
Insercion	Boolean
Actualizacion	Boolean
Eliminacion	Boolean
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.
Nombre:	esIguar(Telefono telefono)
Descripción:	Dado un teléfono determinado sabe si hay uno igual en la base de datos.

Nombre: PeriodoDesactivacionDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarPeriodoDesactivacion(PeriodoDesactivacion periodoDesactivacion)
Nombre:	obtenerTodosPeriodosDesactivacion()
Nombre:	actualizarPeriodoDesactivacion(PeriodoDesactivacion periodoDesactivacion)
Nombre:	obtenerPeriodoDesactivacion(Integer idPeriodoDesactivacion)
Nombre:	eliminarPeriodoDesactivacion(PeriodoDesactivacion periodoDesactivacion)

Nombre: PeriodoDesactivacionDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarPeriodoDesactivacion(PeriodoDesactivacion periodoDesactivacion)
Descripción:	Guarda en la base de datos un periodo de desactivación dado.
Nombre:	obtenerTodosPeriodosDesactivacion()
Descripción:	Obtiene un listado de todos los periodos de desactivación existentes en la base de datos.
Nombre:	actualizarPeriodoDesactivacion(PeriodoDesactivacion periodoDesactivacion)
Descripción:	Actualiza los datos de un periodo de desactivación dado.
Nombre:	obtenerPeriodoDesactivacion(Integer idPeriodoDesactivacion)
Descripción:	Obtiene un periodo de desactivación dado su identificador.
Nombre:	eliminarPeriodoDesactivacion(PeriodoDesactivacion periodoDesactivacion)
Descripción:	Elimina de la base de datos un periodo de desactivación dado.

Nombre: DobleMilitancia	
Tipo de clase: Entidad	
Atributo	Tipo
militante	Militante
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: DobleMilitanciaDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarDobleMilitancia(DobleMilitancia dobleMilitancia)
Nombre:	obtenerTodasDobleMilitancias()
Nombre:	actualizarDobleMilitancia(DobleMilitancia dobleMilitancia)
Nombre:	obtenerDobleMilitancia(String idDobleMilitancia)

Nombre: Parentesco	
Tipo de clase: Nomencladora	
Atributo	Tipo
idParentesco	Integer
tipo	String
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: ParentescoDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarParentesco(Parentesco parentesco)
Nombre:	obtenerTodosParentescos()
Nombre:	actualizarParentesco(Parentesco parentesco)
Nombre:	obtenerParentesco(Integer idParentesco)
Nombre:	eliminarParentesco(Parentesco parentesco)

Nombre: ParentescoDaoImpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarParentesco(Parentesco parentesco)
Descripción:	Guarda un parentesco dado en la base de datos.
Nombre:	obtenerTodosParentescos()
Descripción:	Obtiene un listado de todos los parentescos existentes en la base de datos.
Nombre:	actualizarParentesco(Parentesco parentesco)
Descripción:	Actualiza los datos de un parentesco dado.
Nombre:	obtenerParentesco(Integer idParentesco)
Descripción:	Obtiene un parentesco dado su identificador.
Nombre:	eliminarParentesco(Parentesco parentesco)

Descripción:	Elimina de la base de datos un parentesco dado.
---------------------	---

Nombre: Usuario	
Tipo de clase: Entidad	
Atributo	Tipo
usuario	String
password	String
activo	Boolean
nombreCompleto	String
listaRoles	List<Rol>
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo.

Nombre: UsuarioDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarUsuario(Usuario usuario)
Nombre:	eliminarUsuario(Usuario usuario)
Nombre:	actualizarUsuario(Usuario usuario)
Nombre:	obtenerUsuario(String usuario)

Nombre:	obtenerTodosUsuarios()
Nombre:	autenticarUsuario(String usuario, String password)
Nombre:	obtenerUsuariosConRol(Rol rol)

Nombre: UsuarioDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarUsuario(Usuario usuario)
Descripción:	Guarda un usuario dado en la base de datos.
Nombre:	eliminarUsuario(Usuario usuario)
Descripción:	Elimina un usuario dado de la base de datos.
Nombre:	actualizarUsuario(Usuario usuario)
Descripción:	Modifica los datos de un usuario dado en la base de datos.
Nombre:	obtenerUsuario(String usuario)
Descripción:	Obtiene dado el nombre de usuario el objeto usuario.
Nombre:	obtenerTodosUsuarios()
Descripción:	Obtiene todos los usuarios del sistema.
Nombre:	autenticarUsuario(String usuario, String password)
Descripción:	Permite dado el usuario y la contraseña la entrada de un usuario al sistema con los permisos correspondientes.
Nombre:	obtenerUsuariosConRol(Rol rol)
Descripción:	Obtiene dado un rol los usuarios que tengan ese rol.

Nombre: Rol	
Tipo de clase: Entidad	
Atributo	Tipo
idRol	Integer
rol	String
descripcion	String
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set)
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo.

Nombre: RolDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarRol(Rol rol)
Nombre:	eliminarRol(Rol rol)
Nombre:	actualizarRol(Rol rol)
Nombre:	obtenerRol(Integer idRol)
Nombre:	obtenerTodosRoles()
Nombre:	eliminarTodosRoles()
Nombre:	obtenerRol(String rol)

Nombre: RolDaoImpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarRol(Rol rol)
Descripción:	Guarda un rol dado en la base de datos.
Nombre:	eliminarRol(Rol rol)
Descripción:	Elimina un rol dado de la base de datos.
Nombre:	actualizarRol(Rol rol)
Descripción:	Modifica un rol dado en la base de datos.
Nombre:	obtenerRol(Integer idRol)
Descripción:	Obtiene un rol dado su id.
Nombre:	obtenerTodosRoles()
Descripción:	Obtiene todos los roles que existan en la base de datos.
Nombre:	eliminarTodosRoles()
Descripción:	Elimina todos los roles de la base de datos.
Nombre:	obtenerRol(String rol)
Descripción:	Obtiene un rol dado su nombre.

Nombre: Cargo	
Tipo de clase: Entidad	
Atributo	Tipo
idCargo	String
cargo	Cargo

annoEmpezocargo	Integer
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: TipoCargo	
Tipo de clase: Nomencladora	
Atributo	Tipo
idTipoCargo	Integer
nombreTipoCargo	String
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: TipoCargoDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarTipoCargo(TipoCargo tipoCargo)
Nombre:	obtenerTodosTiposCargos()
Nombre:	actualizarTipoCargo(TipoCargo tipoCargo)

Nombre:	obtenerTipoCargo(Integer idTipoCargo)
Nombre:	eliminarTipoCargo(TipoCargo tipoCargo)

Nombre: TipoCargoDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarTipoCargo(TipoCargo tipoCargo)
Descripción:	Guarda en la base de datos un tipo de cargo dado.
Nombre:	obtenerTodosTiposCargos()
Descripción:	Obtiene un listado de todos los tipos de cargos existentes en la base de datos.
Nombre:	actualizarTipoCargo(TipoCargo tipoCargo)
Descripción:	Actualiza los datos de un tipo de cargo dado.
Nombre:	obtenerTipoCargo(Integer idTipoCargo)
Descripción:	Obtiene un tipo de cargo dado su identificador.
Nombre:	eliminarTipoCargo(TipoCargo tipoCargo)
Descripción:	Elimina de la base de datos un tipo de cargo dado.

Nombre: MotivoTipoAlta	
Tipo de clase: Nomencladora	
Atributo	Tipo
idMotivoTipoAlta	Integer

nombre	String
tipoAlta	TipoAlta
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: MotivoTipoAltaDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarMotivoTipoAlta(MotivoTipoAlta motivoTipoAlta)
Nombre:	obtenerTodosMotivosTipoAlta()
Nombre:	actualizarMotivoTipoAlta(MotivoTipoAlta motivoTipoAlta)
Nombre:	obtenerMotivoTipoAlta(Integer idMotivoTipoAlta)
Nombre:	eliminarMotivoTipoAlta(MotivoTipoAlta motivoTipoAlta)

Nombre: MotivoTipoAltaDaoImpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarMotivoTipoAlta(MotivoTipoAlta motivoTipoAlta)
Descripción:	Guarda en la base de datos un motivo tipo de alta dado.

Nombre:	obtenerTodosMotivosTipoAlta()
Descripción:	Obtiene un listado de todos los motivos tipo de alta existentes en la base de datos.
Nombre:	actualizarMotivoTipoAlta(MotivoTipoAlta motivoTipoAlta)
Descripción:	Actualiza los datos de un motivo tipo de alta dado.
Nombre:	obtenerMotivoTipoAlta(Integer idMotivoTipoAlta)
Descripción:	Obtiene un motivo tipo de alta dado su identificador.
Nombre:	eliminarMotivoTipoAlta(MotivoTipoAlta motivoTipoAlta)
Descripción:	Elimina de la base de datos un motivo tipo de alta dado.

Nombre: MotivoTipoBaja	
Tipo de clase: Nomencladora	
Atributo	Tipo
idMotivoTipoBaja	Integer
tipoBaja	TipoBaja
nombre	String
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: MotivoTipoBajaDao
Tipo de clase: Interfaz

Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarMotivoTipoBaja(MotivoTipoBaja motivoTipoBaja)
Nombre:	obtenerTodosMotivosTipoBaja()
Nombre:	actualizarMotivoTipoBaja(MotivoTipoBaja motivoTipoBaja)
Nombre:	obtenerMotivoTipoBaja(Integer idMotivoTipoBaja)
Nombre:	eliminarMotivoTipoBaja(MotivoTipoBaja motivoTipoBaja)

Nombre: MotivoTipoBajaDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarMotivoTipoBaja(MotivoTipoBaja motivoTipoBaja)
Descripción:	Guarda en la base de datos un motivo tipo de baja dado.
Nombre:	obtenerTodosMotivosTipoBaja()
Descripción:	Obtiene un listado de todos los motivos tipo de bajas existentes en la base de datos.
Nombre:	actualizarMotivoTipoBaja(MotivoTipoBaja motivoTipoBaja)
Descripción:	Actualiza los datos de un motivo tipo de baja dado.
Nombre:	obtenerMotivoTipoBaja(Integer idMotivoTipoBaja)
Descripción:	Obtiene un motivo tipo de baja dado su identificador.
Nombre:	eliminarMotivoTipoBaja(MotivoTipoBaja motivoTipoBaja)
Descripción:	Elimina de la base de datos un motivo tipo de baja dado.

Nombre: TipoMision	
Tipo de clase: Nomencladora	
Atributo	Tipo
idTipoMision	Integer
nombre	String
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: TipoMisionDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarTipoMision(TipoMision tipoMision)
Nombre:	obtenerTodosTiposMisiones()
Nombre:	actualizarTipoMision(TipoMision tipoMision)
Nombre:	obtenerTipoMision(Integer idTipoMision)
Nombre:	eliminarTipoMision(TipoMision tipoMision)

Nombre: TipoMisionDaoImpl	
Tipo de clase:	
Atributo	Tipo

Para cada responsabilidad:	
Nombre:	guardarTipoMision(TipoMision tipoMision)
Descripción:	Guarda en la base de datos un tipo de misión dada.
Nombre:	obtenerTodosTiposMisiones()
Descripción:	Obtiene un listado de todos los tipos de misiones existentes en la base de datos.
Nombre:	actualizarTipoMision(TipoMision tipoMision)
Descripción:	Actualiza los datos de un tipo de misión dada.
Nombre:	obtenerTipoMision(Integer idTipoMision)
Descripción:	Obtiene un tipo de misión dado su identificador.
Nombre:	eliminarTipoMision(TipoMision tipoMision)
Descripción:	Elimina de la base de datos un tipo de misión dada.

Nombre: NivelCultural	
Tipo de clase: Nomencladora	
Atributo	Tipo
idNivelCultural	Integer
nombre	String
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: NivelCulturalDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarNivelCultural(NivelCultural nivelCultural)
Nombre:	obtenerTodosNivelesCulturales()
Nombre:	actualizarNivelCultural(NivelCultural nivelCultural)
Nombre:	obtenerNivelCultural(Integer idNivelCultural)
Nombre:	eliminarNivelCultural(NivelCultural nivelCultural)

Nombre: NivelCulturalDaoImpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarNivelCultural(NivelCultural nivelCultural)
Descripción:	Guarda en la base de datos un nivel cultural dado.
Nombre:	obtenerTodosNivelesCulturales()
Descripción:	Obtiene un listado de todos los niveles culturales existentes en la base de datos.
Nombre:	actualizarNivelCultural(NivelCultural nivelCultural)
Descripción:	Actualiza los datos de un nivel cultural dado.
Nombre:	obtenerNivelCultural(Integer idNivelCultural)
Descripción:	Obtiene un nivel cultural dado su identificador.
Nombre:	eliminarNivelCultural(NivelCultural nivelCultural)

Descripción:	Elimina de la base de datos un nivel cultural dado.
---------------------	---

Nombre: ComiteBase	
Tipo de clase: Entidad	
Atributo	Tipo
centro	Centro
rama	Rama
sector	Sector
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: ComiteBaseDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarComiteBase(ComiteBase comiteBase)
Nombre:	eliminarComiteBase(ComiteBase comiteBase)
Nombre:	actualizarComiteBase(ComiteBase comiteBase)
Nombre:	obtenerComiteBase(String idComiteBase)

Nombre:	obtenerComiteBaseNombre(String nombre)
Nombre:	obtenerComitesBasesTipoComiteBase(Estructura tipoComiteBase)
Nombre:	obtenerTodosComitesBases()
Nombre:	obtenerTodosComitesBasesNoActivos()
Nombre:	obtenerComitesBasesNivelSuperior(Integer nivel)
Nombre:	cantidadMilitantes()
Nombre:	cantidadUniversoJuvenil()
Nombre:	obtenerTodosComiteBasesMunicipio(Municipio municipio)
Nombre:	obtenerComiteBase(ComiteBase comiteBase)
Nombre:	obtenerSectorComiteBase()

Nombre: ComiteBaseDaImpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarComiteBase(ComiteBase comiteBase)
Descripción:	Guarda un comité de base dado en la base de datos.
Nombre:	eliminarComiteBase(ComiteBase comiteBase)
Descripción:	Elimina de la base de datos un comité de base dado.
Nombre:	actualizarComiteBase(ComiteBase comiteBase)
Descripción:	Actualiza los datos de un comité de base dado.
Nombre:	obtenerComiteBase(String idComiteBase)
Descripción:	Obtiene un comité de base dado su identificador.

Nombre:	obtenerComiteBaseNombre(String nombre)
Descripción:	Obtiene un comité de base dado su nombre.
Nombre:	obtenerComitesBasesTipoComiteBase(Estructura tipoComiteBase)
Descripción:	Obtiene un listado de los comités de base de una estructura especificada.
Nombre:	obtenerTodosComitesBases()
Descripción:	Obtiene un listado de todos los comités de base existentes en la base de datos.
Nombre:	obtenerTodosComitesBasesNoActivos()
Descripción:	Obtiene un listado de los comités de base no activos.
Nombre:	obtenerComitesBasesNivelSuperior(Integer nivel)
Descripción:	Obtiene un listado de los comités de base que están por encima del nivel indicado.
Nombre:	cantidadMilitantes()
Descripción:	Obtiene la cantidad de militantes que están registrados en la base de datos.
Nombre:	cantidadUniversoJuvenil()
Descripción:	Obtiene la cantidad de miembros del universo juvenil que están registrados en la base de datos.
Nombre:	obtenerTodosComiteBasesMunicipio(Municipio municipio)
Descripción:	Obtiene una lista de los comités de base de un municipio dado.
Nombre:	obtenerComiteBase(ComiteBase comiteBase)
Descripción:	Obtiene un comité de base dado un objeto de tipo ComiteBase.
Nombre:	obtenerSectorComiteBase()
Descripción:	Obtiene una lista de los comités de base que pertenecen a un sector.

Nombre: Comite	
Tipo de clase: Entidad	
Atributo	Tipo
idComite	String
estructura	Estructura
comiteSuperior	Comite
municipio	Municipio
provincia	Provincia
nombre	String
esActivo	Boolean
insercion	Boolean
actualizacion	Boolean
eliminacion	Boolean
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: ComiteDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarComite(Comite comite)

Nombre:	eliminarComite(Comite comite)
Nombre:	actualizarComite(Comite comite)
Nombre:	obtenerComite(String idComite)
Nombre:	obtenerComiteNombre(String nombre)
Nombre:	obtenerComitesTipoComite(Estructura tipoComite)
Nombre:	obtenerTodosComites()
Nombre:	obtenerTodosComitesNoActivos()
Nombre:	obtenerComitesNivelSuperior(Integer nivel)
Nombre:	cantidadMilitantes()
Nombre:	cantidadUniversoJuvenil()
Nombre:	obtenerComite(Comite comite)
Nombre:	obtenerComitesNivel(Integer nivel)
Nombre:	obtenerComitesSubordinadosA(Comite comite)
Nombre:	obtenerTodosComitesMunicipio(Municipio municipio)

Nombre: ComiteDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarComite(Comite comite)
Descripción:	Guarda un comité dado en la base de datos.
Nombre:	eliminarComite(Comite comite)
Descripción:	Elimina un comité dado de la base de datos.

Nombre:	actualizarComite(Comite comite)
Descripción:	Actualiza los datos de un comité dado.
Nombre:	obtenerComite(String idComite)
Descripción:	Obtiene un comité dado su identificador.
Nombre:	obtenerComiteNombre(String nombre)
Descripción:	Obtiene un comité dado su nombre.
Nombre:	obtenerComitesTipoComite(Estructura tipoComite)
Descripción:	Obtiene una lista con los comités que pertenezcan a la estructura especificada.
Nombre:	obtenerTodosComites()
Descripción:	Obtiene una lista con todos los comités existentes en la base de datos.
Nombre:	obtenerTodosComitesNoActivos()
Descripción:	Obtiene una lista de todos los comités no activos.
Nombre:	obtenerComitesNivelSuperior(Integer nivel)
Descripción:	Obtiene un listado de los comités que están en el nivel superior al especificado.
Nombre:	cantidadMilitantes()
Descripción:	Obtiene la cantidad de militantes que están registrados en la base de datos.
Nombre:	cantidadUniversoJuvenil()
Descripción:	Obtiene la cantidad de miembros del universo juvenil que están registrados en la base de datos.
Nombre:	obtenerComite(Comite comite)
Descripción:	Obtiene un comité dado un objeto de tipo Comite.
Nombre:	obtenerComitesNivel(Integer nivel)
Descripción:	Obtiene una lista de los comités que se encuentran en el nivel especificado.
Nombre:	obtenerComitesSubordinadosA(Comite comite)

Descripción:	Obtiene un listado de los comités que estén subordinados al comité especificado.
Nombre:	obtenerTodosComitesMunicipio(Municipio municipio)
Descripción:	Obtiene un listado de los comités que existen en un municipio especificado.

Nombre: ColorPiel	
Tipo de clase: Nomencladora	
Atributo	Tipo
idColorPiel	Integer
nombre	String
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: ColorPielDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarColorPiel(ColorPiel colorPiel)
Nombre:	obtenerTodosColoresPiel()
Nombre:	actualizarColorPiel(ColorPiel colorPiel)
Nombre:	obtenerColorPiel(Integer idColorPiel)

Nombre:	eliminarColorPiel(ColorPiel colorPiel)
----------------	--

Nombre: ColorPielDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarColorPiel(ColorPiel colorPiel)
Descripción:	Guarda en la base de datos un color de piel dado.
Nombre:	obtenerTodosColoresPiel()
Descripción:	Obtiene un listado con los colores de piel existentes en la base de datos.
Nombre:	actualizarColorPiel(ColorPiel colorPiel)
Descripción:	Actualiza los datos de un color de piel dado.
Nombre:	obtenerColorPiel(Integer idColorPiel)
Descripción:	Obtiene un color de piel dado su identificador.
Nombre:	eliminarColorPiel(ColorPiel colorPiel)
Descripción:	Elimina de la base de datos un color de piel dado.

Nombre: ClasificadorOcupacional	
Tipo de clase: Nomencladora	
Atributo	Tipo
idClasificadorOcupacional	Integer
nombre	String
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: ClasificadorOcupacionalDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarClasificadorOcupacional(ClasificadorOcupacional clasificadorOcupacional)
Nombre:	obtenerTodosClasificadoresOcupacionales()
Nombre:	actualizarClasificadorOcupacional(ClasificadorOcupacional clasificadorOcupacional)
Nombre:	obtenerClasificadorOcupacional(Integer idClasificadorOcupacional)
Nombre:	eliminarClasificadorOcupacional(ClasificadorOcupacional clasificadorOcupacional)

Nombre: ClasificadorOcupacionalDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarClasificadorOcupacional(ClasificadorOcupacional clasificadorOcupacional)
Descripción:	Guarda en la base de datos un clasificador ocupacional dado.
Nombre:	obtenerTodosClasificadoresOcupacionales()
Descripción:	Obtiene un listado de todos los clasificadores ocupacionales existentes en la base de datos.
Nombre:	actualizarClasificadorOcupacional(ClasificadorOcupacional clasificadorOcupacional)
Descripción:	Actualiza los datos de un clasificador ocupacional dado.
Nombre:	obtenerClasificadorOcupacional(Integer idClasificadorOcupacional)
Descripción:	Obtiene un clasificador ocupacional dado su identificador.
Nombre:	eliminarClasificadorOcupacional(ClasificadorOcupacional clasificadorOcupacional)
Descripción:	Elimina de la base de datos un clasificador ocupacional dado.

Nombre: ClasificadorGeneral	
Tipo de clase: Nomencladora	
Atributo	Tipo
idClasificadorGeneral	Integer

nombre	String
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: ClasificadorGeneralDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarClasificadorGeneral(ClasificadorGeneral clasificadorGeneral)
Nombre:	obtenerTodosClasificadoresGenerales()
Nombre:	actualizarClasificadorGeneral(ClasificadorGeneral clasificadorGeneral)
Nombre:	obtenerClasificadorGeneral(Integer idClasificadorGeneral)
Nombre:	eliminarClasificadorGeneral(ClasificadorGeneral clasificadorGeneral)

Nombre: ClasificadorGeneralDaoImpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarClasificadorGeneral(ClasificadorGeneral clasificadorGeneral)

Descripción:	Guarda en la base de datos un clasificador general dado.
Nombre:	obtenerTodosClasificadoresGenerales()
Descripción:	Obtiene un listado de todos los clasificadores generales existentes en la base de datos.
Nombre:	actualizarClasificadorGeneral(ClasificadorGeneral clasificadorGeneral)
Descripción:	Actualiza los datos de un clasificador general dado.
Nombre:	obtenerClasificadorGeneral(Integer idClasificadorGeneral)
Descripción:	Obtiene un clasificador general dado su identificador.
Nombre:	eliminarClasificadorGeneral(ClasificadorGeneral clasificadorGeneral)
Descripción:	Elimina de la base de datos un clasificador general dado.

Nombre: TipoAlta	
Tipo de clase: Nomencladora	
Atributo	Tipo
idTipoAlta	Integer
nombre	String
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: TipoAltaDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarTipoAlta(TipoAlta tipoAlta)
Nombre:	obtenerTodosTiposAlta()
Nombre:	actualizarTipoAlta(TipoAlta tipoAlta)
Nombre:	obtenerTipoAlta(Integer idTipoAlta)
Nombre:	eliminarTipoAlta(TipoAlta tipoAlta)

Nombre: TipoAltaDaoImpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarTipoAlta(TipoAlta tipoAlta)
Descripción:	Guarda en la base de datos un tipo de alta dado.
Nombre:	obtenerTodosTiposAlta()
Descripción:	Obtiene un listado de todos los tipos de altas existentes en la base de datos.
Nombre:	actualizarTipoAlta(TipoAlta tipoAlta)
Descripción:	Actualiza los datos de un tipo de alta dado.
Nombre:	obtenerTipoAlta(Integer idTipoAlta)
Descripción:	Obtiene un tipo de alta dado su identificador.
Nombre:	eliminarTipoAlta(TipoAlta tipoAlta)

Descripción:	Elimina de la base de datos un tipo de alta dado.
---------------------	---

Nombre: TipoBaja	
Tipo de clase: Nomencladora	
Atributo	Tipo
idTipoBaja	Integer
nombre	String
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: TipoBajaDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarTipoBaja(TipoBaja tipoBaja)
Nombre:	obtenerTodosTiposBaja()
Nombre:	actualizarTipoBaja(TipoBaja tipoBaja)
Nombre:	obtenerTipoBaja(Integer idTipoBaja)
Nombre:	eliminarTipoBaja(TipoBaja tipoBaja)

Nombre: TipoBajaDaolmpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarTipoBaja(TipoBaja tipoBaja)
Descripción:	Guarda en la base de datos un tipo de baja dado.
Nombre:	obtenerTodosTiposBaja()
Descripción:	Obtiene un listado de todos los tipos de baja existentes en la base de datos.
Nombre:	actualizarTipoBaja(TipoBaja tipoBaja)
Descripción:	Actualiza los datos de un tipo de baja dado.
Nombre:	obtenerTipoBaja(Integer idTipoBaja)
Descripción:	Obtiene un tipo de baja dado su identificador.
Nombre:	eliminarTipoBaja(TipoBaja tipoBaja)
Descripción:	Elimina de la base de datos un tipo de baja dado.

Nombre: Estructura	
Tipo de clase: Nomencladora	
Atributo	Tipo
idEstructura	Integer
nombre	String

estructuraSuperior	Estructura
nivel	Integer
Para cada responsabilidad:	
Nombre:	Para cada atributo los métodos de acceso correspondientes (Get y Set).
Descripción:	Los métodos de acceso Get se utilizan para obtener el atributo y los métodos de acceso Set se utilizan para cambiar el valor del atributo por uno que sea indicado.

Nombre: EstructuraDao	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarEstructura(Estructura estructura)
Nombre:	eliminarEstructura(Estructura estructura)
Nombre:	actualizarEstructura(Estructura estructura)
Nombre:	obtenerEstructura(Integer idEstructura)
Nombre:	obtenerEstructuraNombre(String nombreEstructura)
Nombre:	obtenerTodasEstructuras()

Nombre: EstructuraDaoImpl	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	guardarEstructura(Estructura estructura)

Descripción:	Guarda en la base de datos una estructura dada.
Nombre:	eliminarEstructura(Estructura estructura)
Descripción:	Elimina de la base de datos una estructura dada.
Nombre:	actualizarEstructura(Estructura estructura)
Descripción:	Actualiza los datos de una estructura dada.
Nombre:	obtenerEstructura(Integer idEstructura)
Descripción:	Obtiene una estructura dado su identificador.
Nombre:	obtenerEstructuraNombre(String nombreEstructura)
Descripción:	Obtiene una estructura dado su nombre.
Nombre:	obtenerTodasEstructuras()
Descripción:	Obtiene un listado de todas las estructuras existentes en la base de datos.

Anexo 6. Descripción de las clases persistentes

Nombre: TipoAlta		
Descripción: Recoge los datos relacionados con el tipo de alta.		
Atributo	Tipo	Descripción
idTipoAlta	Integer	Identificador principal del tipo de alta.
nombre	String	Nombre del alta.

Nombre: TipoBaja		
Descripción: Recoge los datos relacionados con el tipo de baja.		
Atributo	Tipo	Descripción
idTipoBaja	Integer	Identificador principal del tipo de baja.
nombre	String	Nombre de la baja.

Nombre: MotivoTipoAlta		
Descripción: Recoge los datos relacionados con el motivo del tipo de alta.		
Atributo	Tipo	Descripción
idMotivoTipoAlta	Integer	Identificador principal del motivo del tipo de alta.
tipoAlta	TipoAlta	Hace referencia al tipo de alta.
nombre	String	Nombre del motivo del tipo de alta.

Nombre: MotivoTipoBaja		
Descripción: Recoge los datos relacionados con el motivo del tipo de baja.		
Atributo	Tipo	Descripción
idMotivoTipoBaja	Integer	Identificador principal del motivo del tipo de baja.
tipoBaja	TipoBaja	Hace referencia al tipo de baja.
nombre	String	Nombre del motivo del tipo de baja.

Nombre: Ciudadania		
Descripción: Recoge los datos relacionados con la ciudadanía.		
Atributo	Tipo	Descripción
idCiudadania	Integer	Identificador principal de la ciudadanía.
nombre	String	Nombre de la ciudadanía.

Nombre: Familiar		
Descripción: Recoge los datos relacionados con familiares de los militantes.		
Atributo	Tipo	Descripción
idFamiliar	String	Identificador principal del familiar.
militante	Militante	Hace referencia al militante.
parentesco	Parentesco	Hace referencia al parentesco.
primerNombre	String	Primer nombre del familiar.
segundoNombre	String	Segundo nombre del familiar
primerApellido	String	Primer apellido del familiar
segundoApellido	String	Segundo apellido del familiar.

insertado	Boolean	Toma valor verdadero si es insertado un familiar.
actualizado	Boolean	Toma valor verdadero si es actualizado un familiar.
eliminado	Boolean	Toma valor verdadero si es eliminado un familiar.

Nombre: Parentesco		
Descripción: Recoge los datos relacionados con el parentesco.		
Atributo	Tipo	Descripción
idParentesco	Integer	Identificador principal del parentesco.
tipo	String	Tipo de parentesco.

Nombre: VialIngreso		
Descripción: Recoge los datos relacionados con la vía por la que ingresa un militante.		
Atributo	Tipo	Descripción
idVialIngreso	Integer	Identificador principal de la vía de ingreso utilizada.
nombre	String	Hace referencia al nombre de la vía por la que ingresa.

Nombre: ProcesosIngreso		
Descripción: Recoge los datos relacionados con el proceso de ingreso a las filas de la UJC.		
Atributo	Tipo	Descripción
idProcesosIngreso	String	Identificador principal del proceso de ingreso.
idpersona	String	Hace referencia a la persona a la que se le está realizando el proceso de ingreso.
idMunicipio	Integer	Hace referencia al municipio al que pertenece la persona

		que se le está realizando el proceso.
idVialIngreso	Integer	Hace referencia a la vía mediante la cual se realiza el proceso.
fechaIngresoUJC	Date	Hace referencia a la fecha de ingreso a la Unión de Jóvenes Comunistas.
organizacionDesarrolloProc	String	Describe cómo se desarrolló el proceso.
escPoliCurs	Boolean	Devuelve verdadero si cursó escuelas políticas.
escAdm	Boolean	Devuelve verdadero si cursó escuelas administrativas.
Insercion	Boolean	Devuelve verdadero si es insertado.
Actualización	Boolean	Devuelve verdadero si es actualizado.
Eliminación	Boolean	Devuelve verdadero si es eliminado.

Nombre: DobleMilitancia		
Descripción: Recoge los datos relacionados con los militantes que tienen doble militancia.		
Atributo	Tipo	Descripción
militante	Militante	Identificador principal de la persona que tenga la doble militancia.
fechaIngresoPCC	Date	Hace referencia a la fecha de ingreso al PCC.

Nombre: ColorPiel		
Descripción: Recoge los datos relacionados con el color de la piel de una persona.		
Atributo	Tipo	Descripción
idColorPiel	Integer	Identificador principal del color de la piel.

nombre	String	Hace referencia al nombre del color de la piel.
--------	--------	---

Nombre: NivelCultural		
Descripción: Recoge los datos relacionados con el nivel cultural de una persona.		
Atributo	Tipo	Descripción
idNivelCultural	Integer	Identificador principal del nivel cultural.
nombre	String	Hace referencia al nombre del nivel cultural.

Nombre: ClasificadorOcupacional		
Descripción: Recoge los datos relacionados con la clasificación ocupacional de una persona.		
Atributo	Tipo	Descripción
idClasificadorOcupacional	Integer	Identificador principal del clasificador ocupacional.
clasificadorGeneral	ClasificadorGeneral	
nombre	String	Hace referencia al nombre del clasificador ocupacional.

Nombre: Usuario		
Descripción: Recoge los datos relacionados con un usuario.		
Atributo	Tipo	Descripción
usuario	String	Hace referencia al nombre con que el usuario puede acceder al sistema y es el identificador principal del usuario.
password	String	Hace referencia a la contraseña del usuario para acceder al sistema.
activo	Boolean	Indica el estado del usuario, o

		sea, si esta activo o no.
nombreCompleto	String	Hace referencia al nombre del usuario.
listaRoles	List<Rol>	Indica los roles que desempeña el usuario y por los cuales tendrá determinados permisos en la aplicación.

Nombre: Rol		
Descripción: Recoge los datos relacionados con un Rol.		
Atributo	Tipo	Descripción
idRol	Integer	Identificador principal del rol.
rol	String	Hace referencia a un rol en específico.
descripcion	String	Indica la descripción del rol.

Nombre: Cargo		
Descripción: Recoge los datos relacionados con los cargos del militante.		
Atributo	Tipo	Descripción
idMilitanteCargo	String	Identificador principal del cargo del militante.
cargo	Cargo	Hace referencia al cargo.
annoEmpezocargo	Integer	Indica el año en que el militante comenzó a desempeñar el cargo.

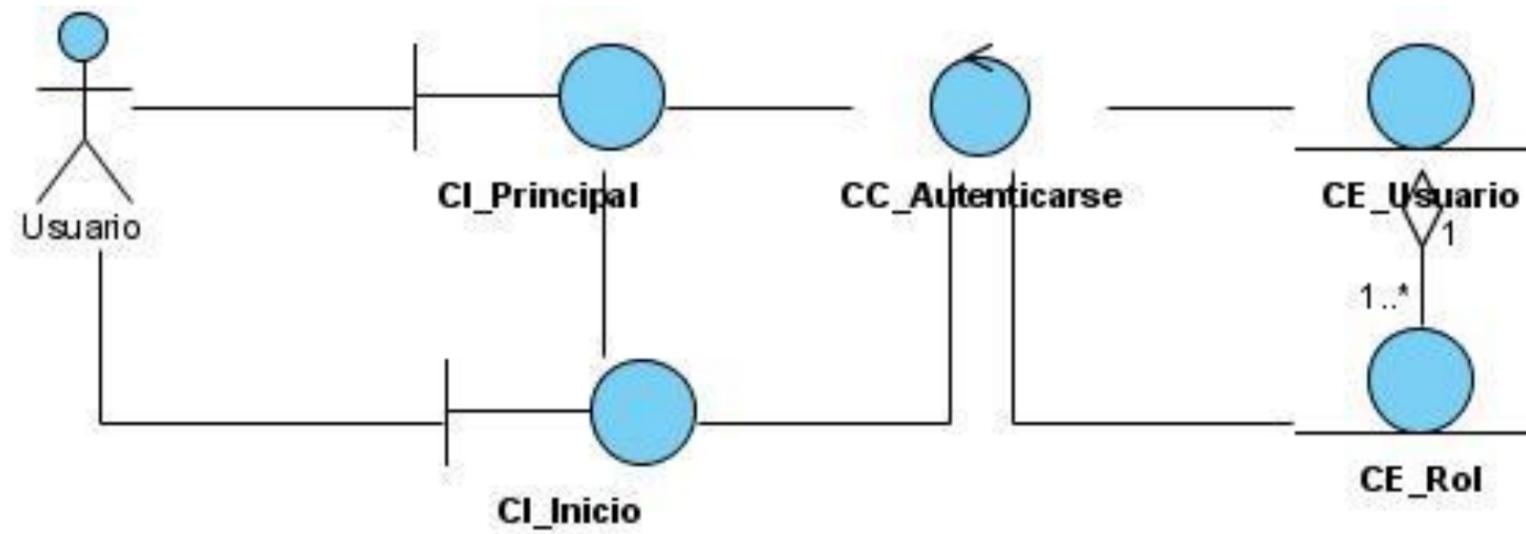
Nombre: PeriodoDesactivacion		
Descripción: Recoge los datos relacionados con el periodo de desactivación.		
Atributo	Tipo	Descripción
idPeriodoDesactivacion	String	Identificador principal del periodo de desactivación.
idComite	String	Hace referencia al comité al que pertenece.
Desactivacion	Date	Fecha en que se desactiva el comité.
Activacion	Date	Fecha en que activa el comité.
Insercion	Boolean	Toma valor verdadero si es insertado.
Actualizacion	Boolean	Toma valor verdadero si es actualizado.
Eliminacion	Boolean	Toma valor verdadero si es eliminado.

Nombre: TipoCargo		
Descripción: Recoge los datos relacionados con los tipos de cargos.		
Atributo	Tipo	Descripción
idTipoCargo	Integer	Identificador principal del tipo de cargo.
nombreTipoCargo	String	Hace referencia al nombre del tipo de cargo.

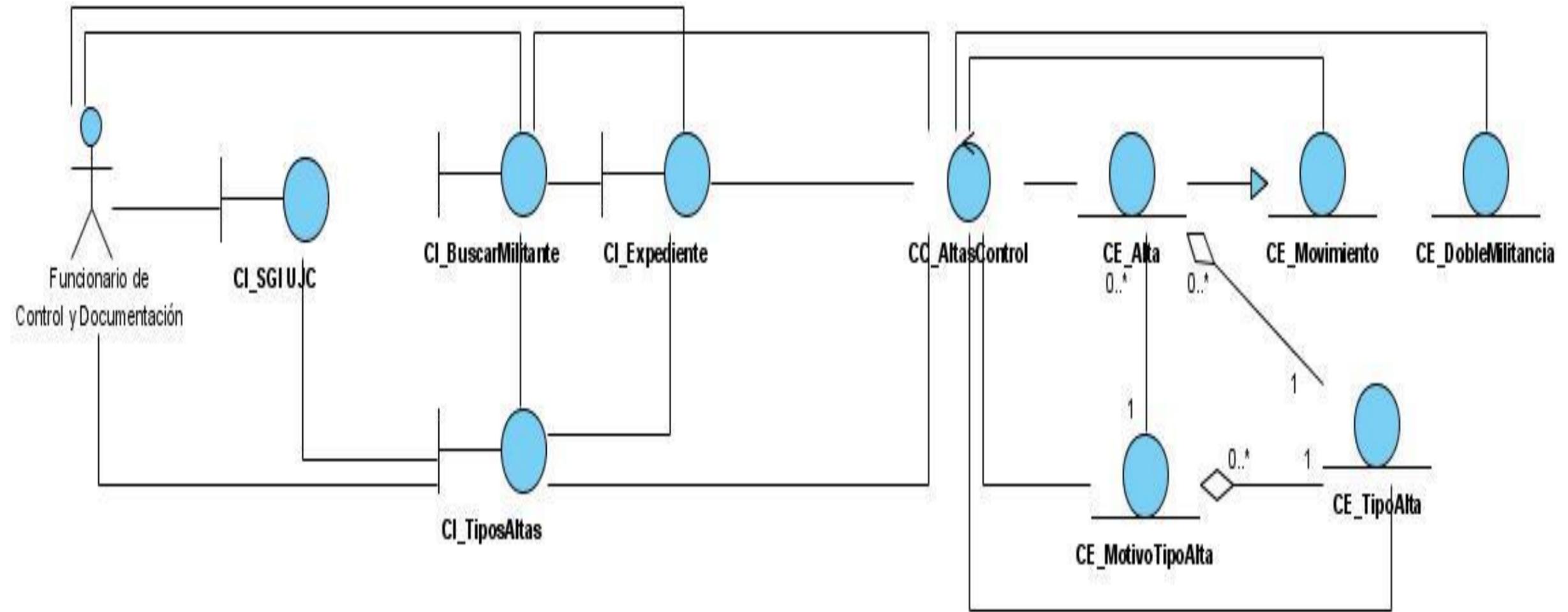
Nombre: ClasificadorGeneral		
Descripción: Recoge los datos relacionados con el clasificador general.		
Atributo	Tipo	Descripción
idClasificadorGeneral	Integer	Identificador principal del clasificador general.
nombre	String	Hace referencia al nombre del clasificador general.

Anexo 7. Diagramas de clases del análisis

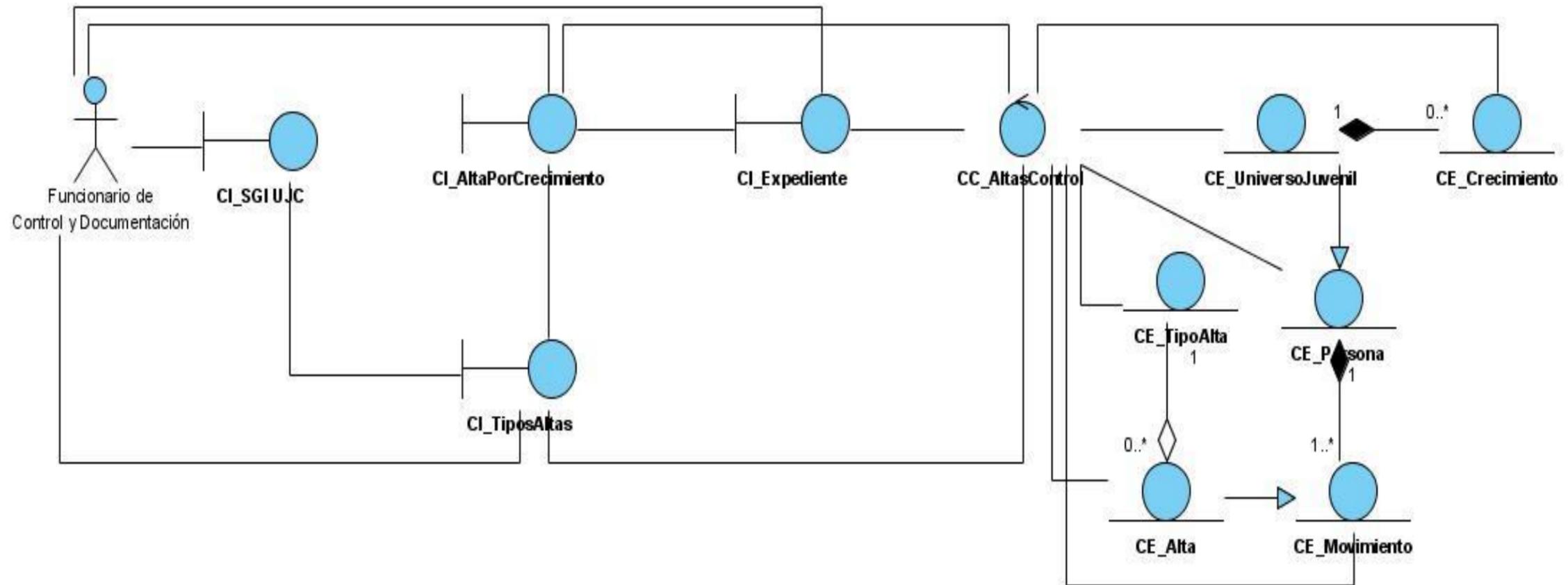
DCA CU_ Autenticar Usuario



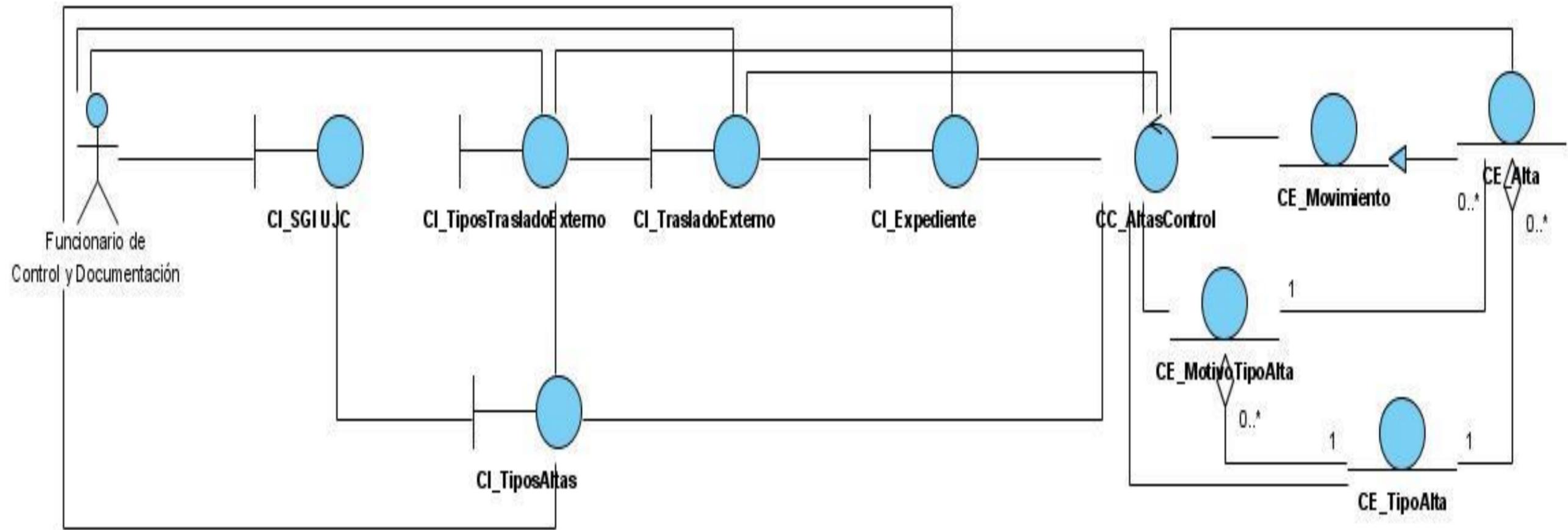
DCA CU_ Alta por Ajuste



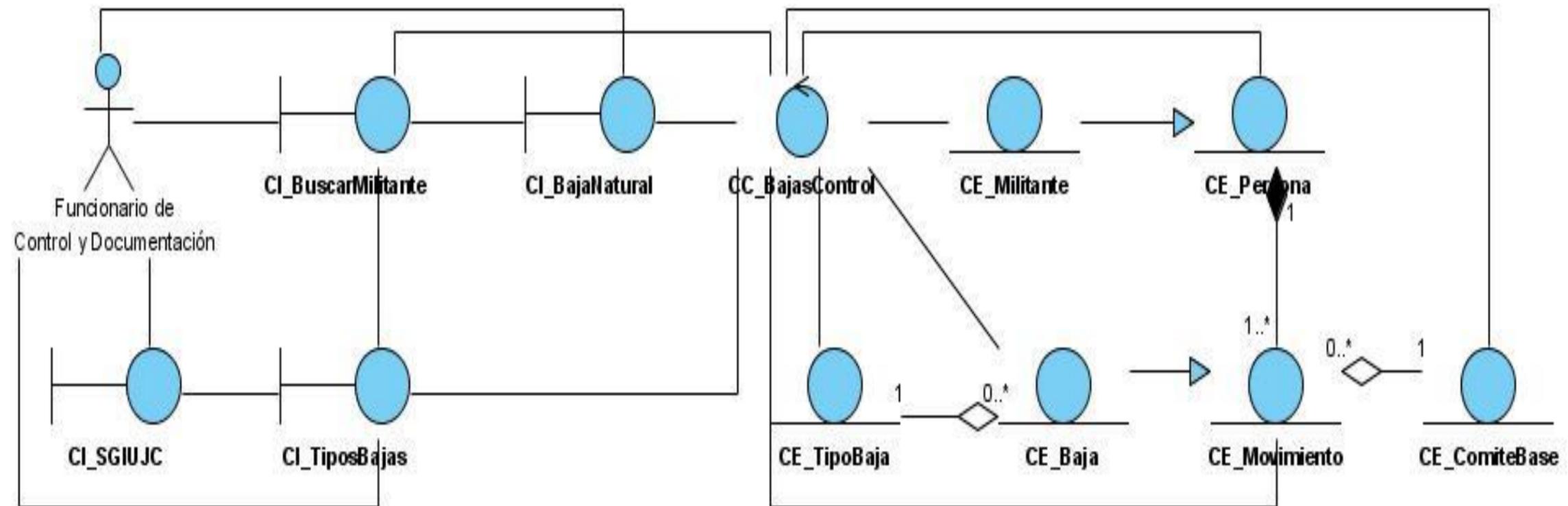
DCA CU_ Alta por Crecimiento



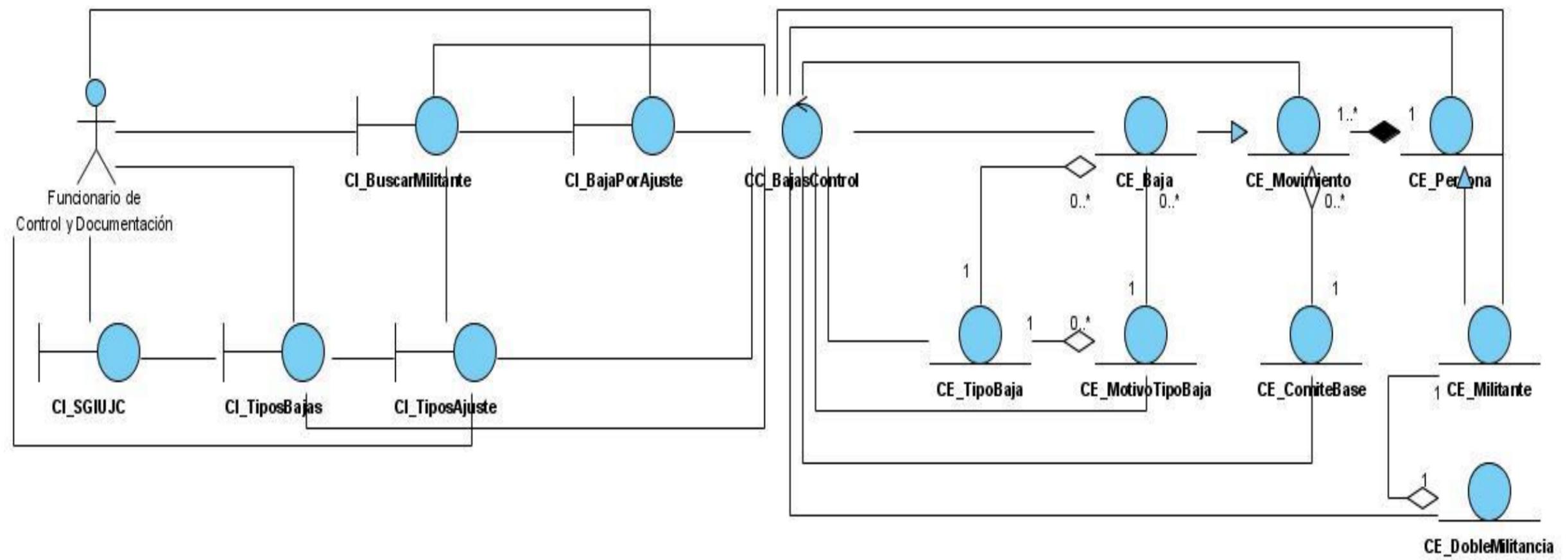
DCA CU_ Alta por Traslado Externo



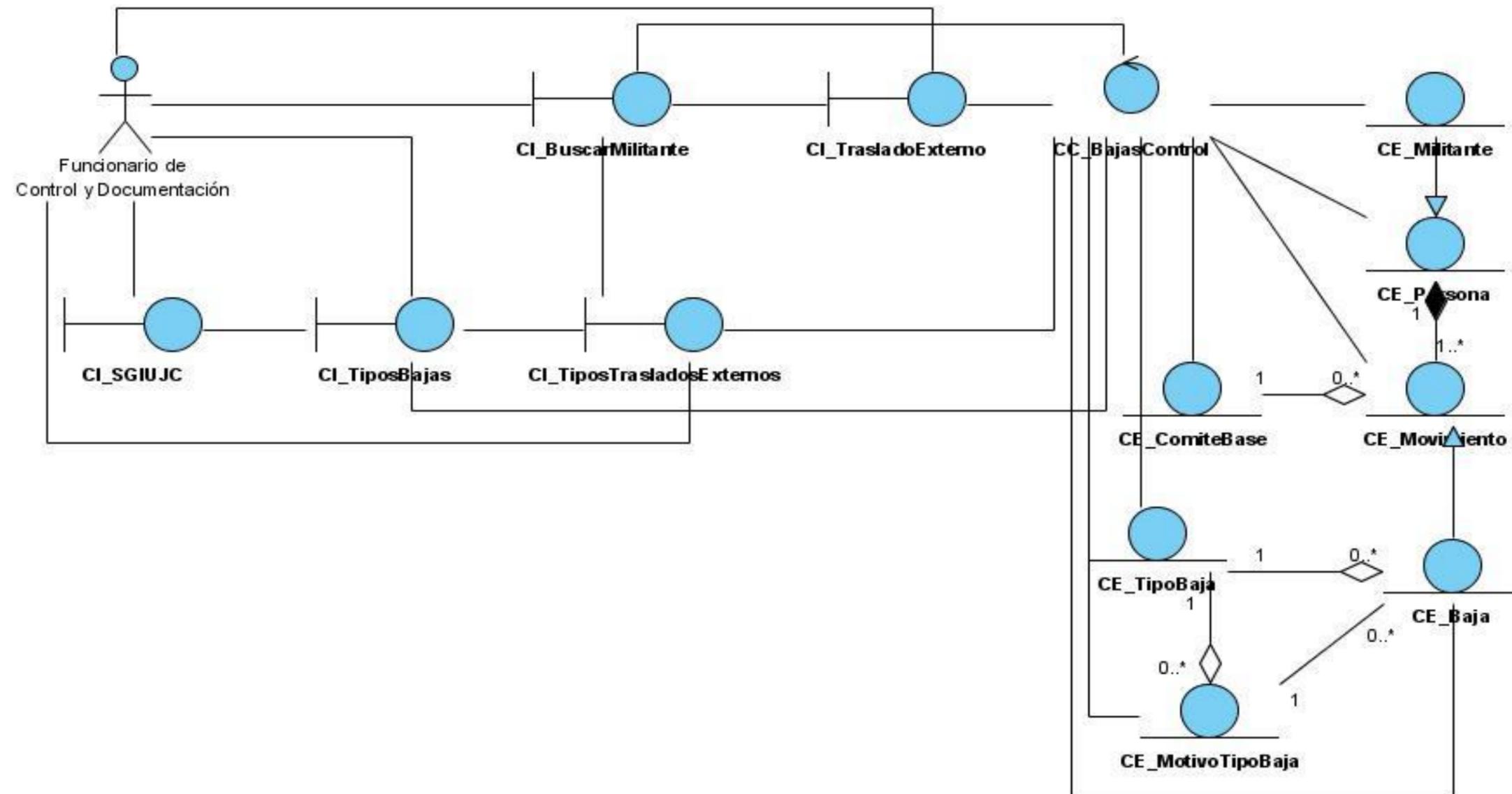
DCA CU_ Baja Natural y CU_ Baja por Defunción



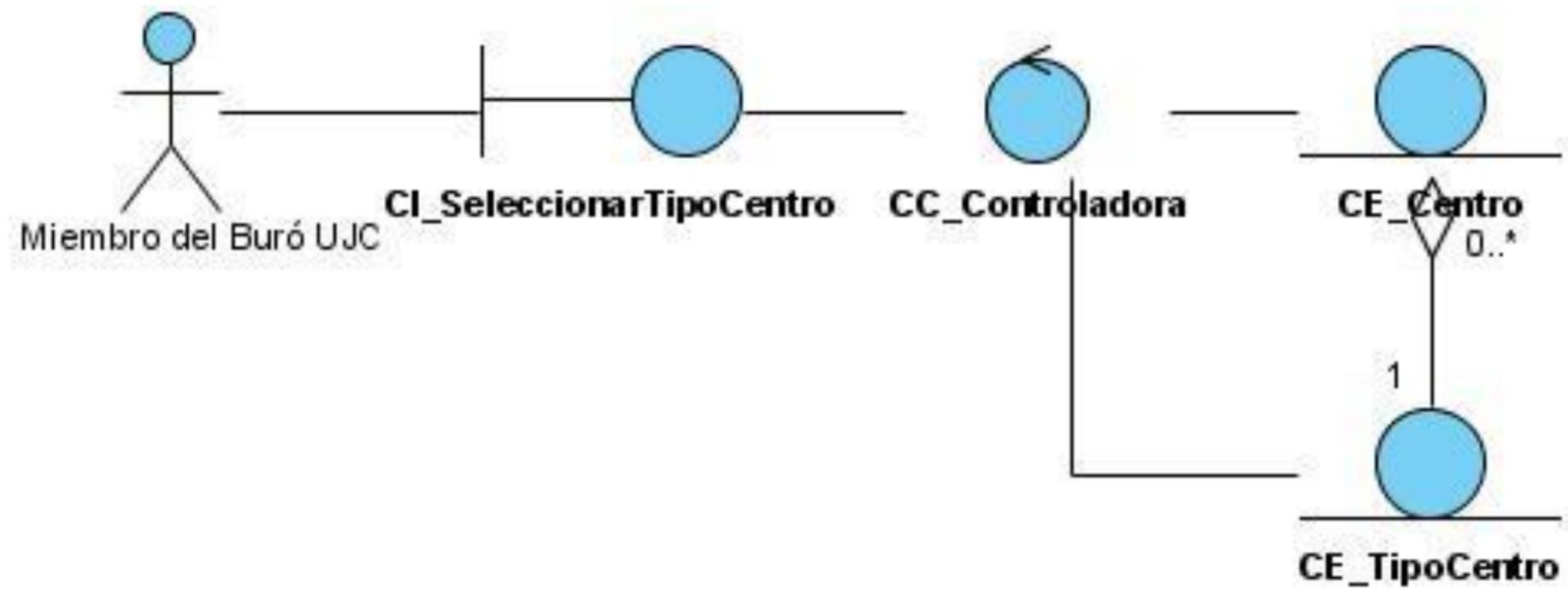
DCA CU_ Baja por Ajuste



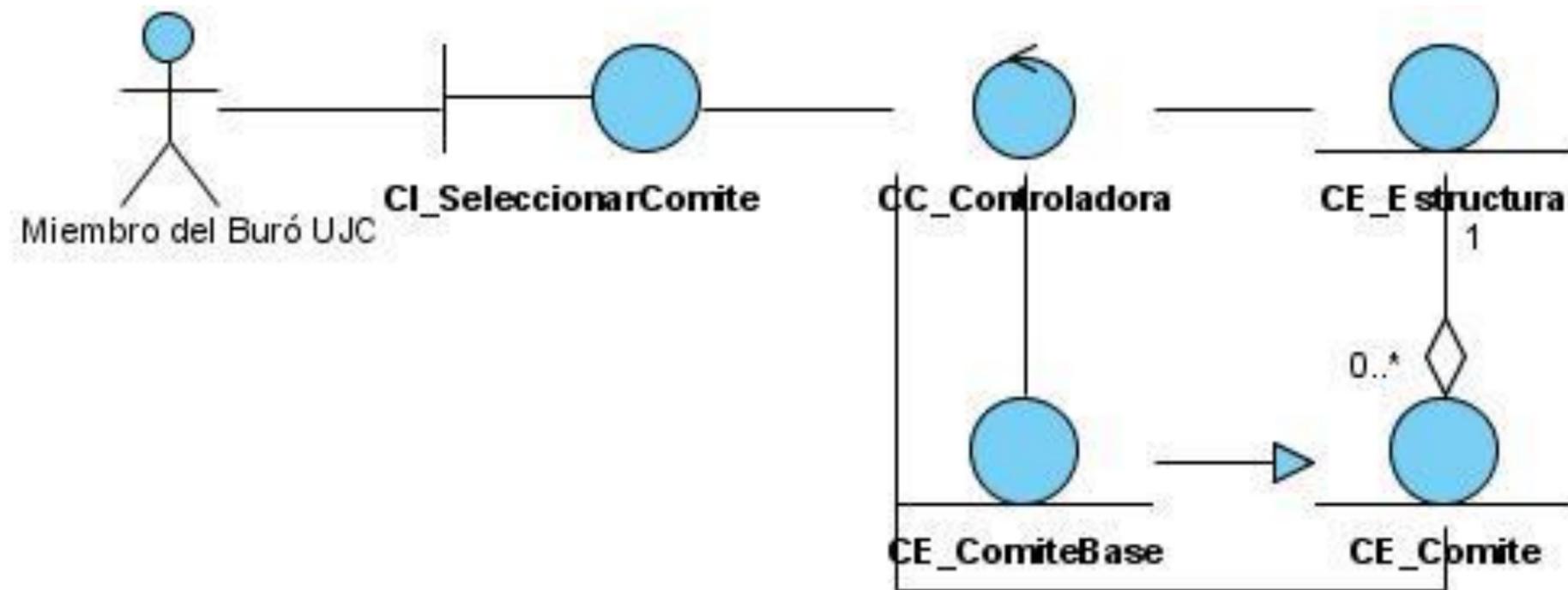
DCA CU_ Baja por Traslado Externo



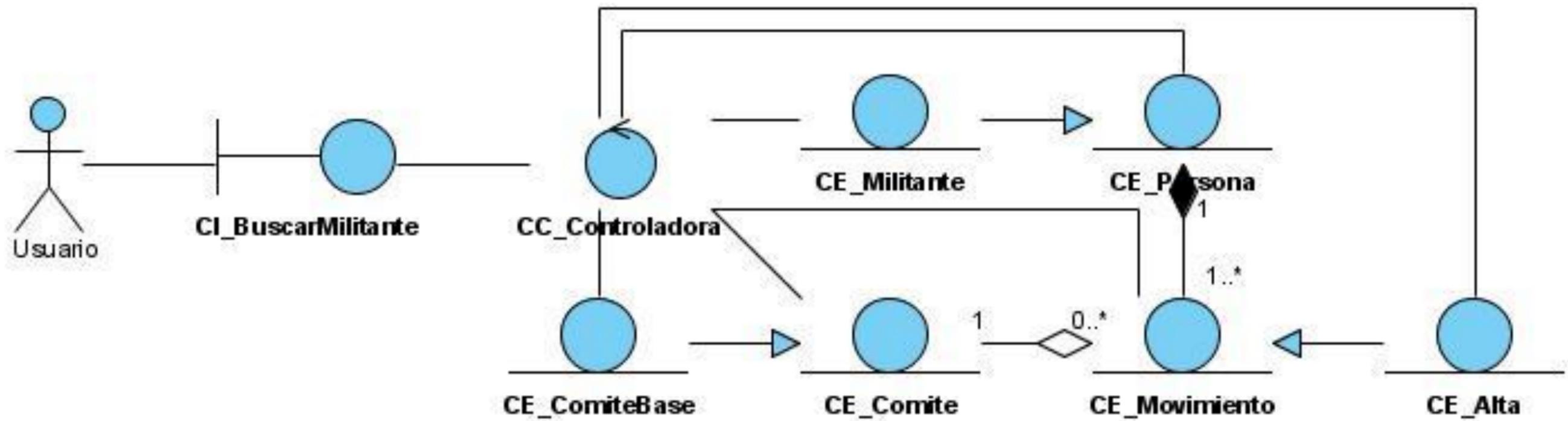
DCA CU_ Buscar Centro



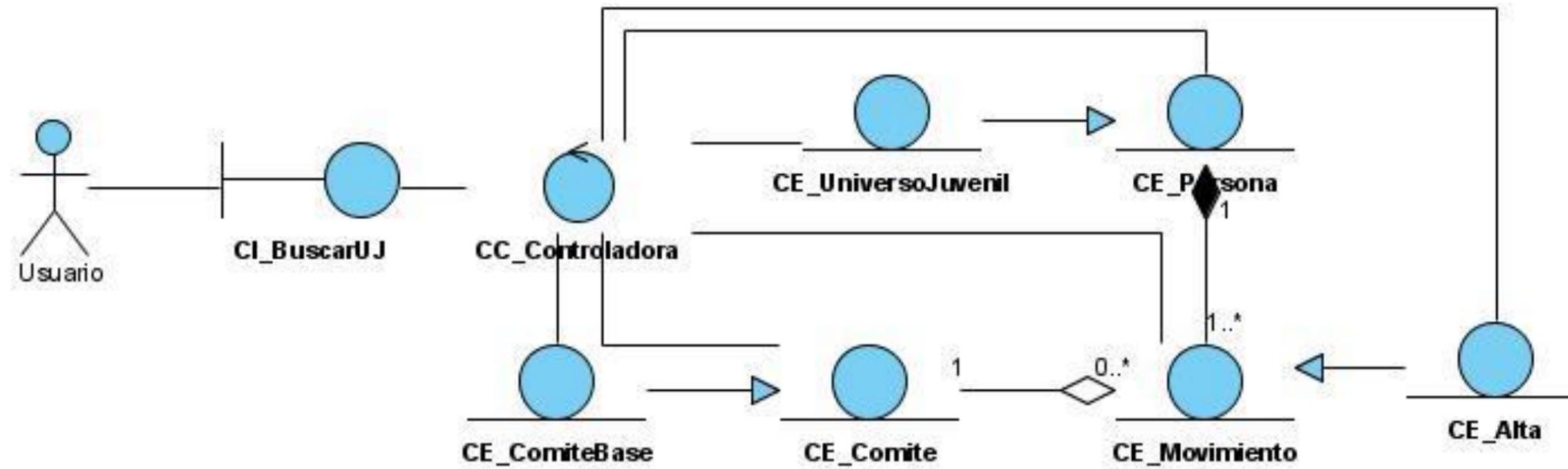
DCA CU_ Buscar Estructura



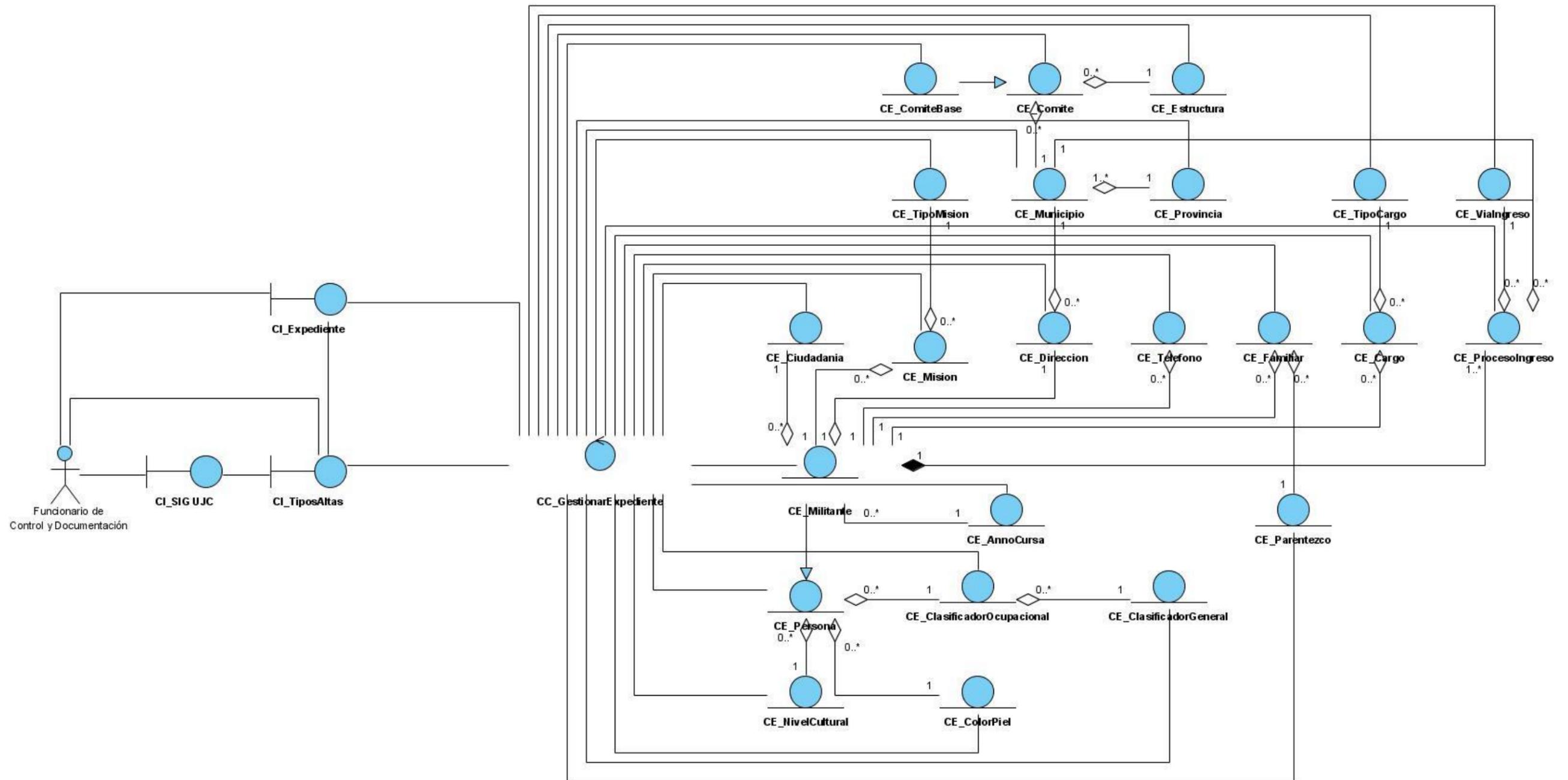
DCA CU_ Buscar Militante



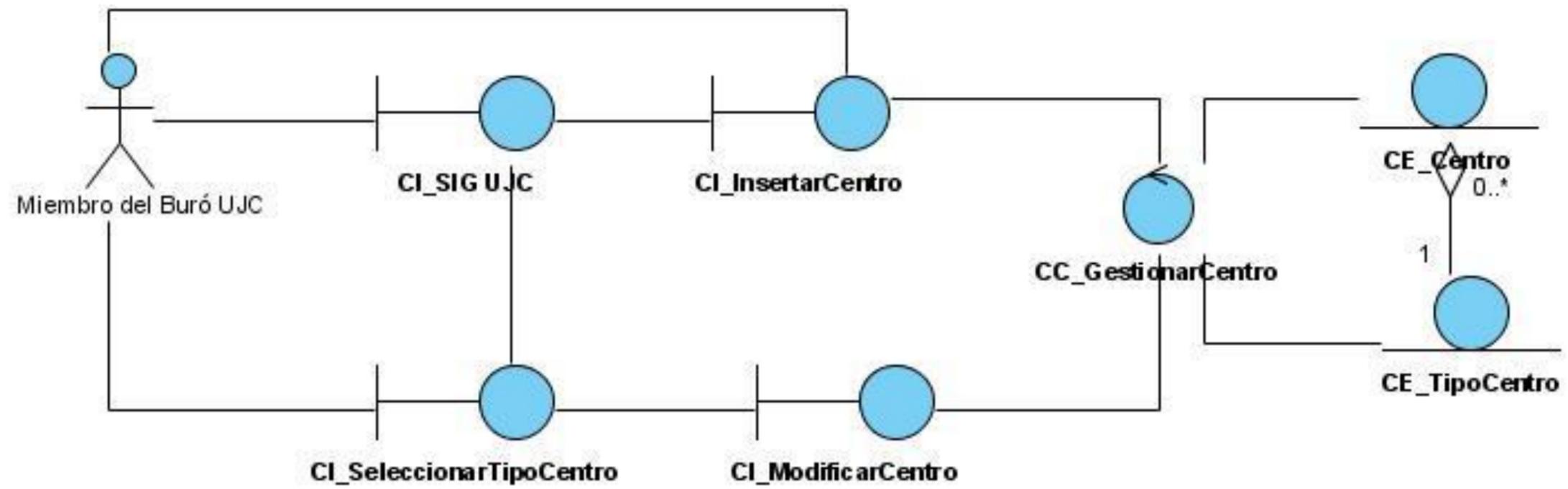
DCA CU_ Buscar UJ



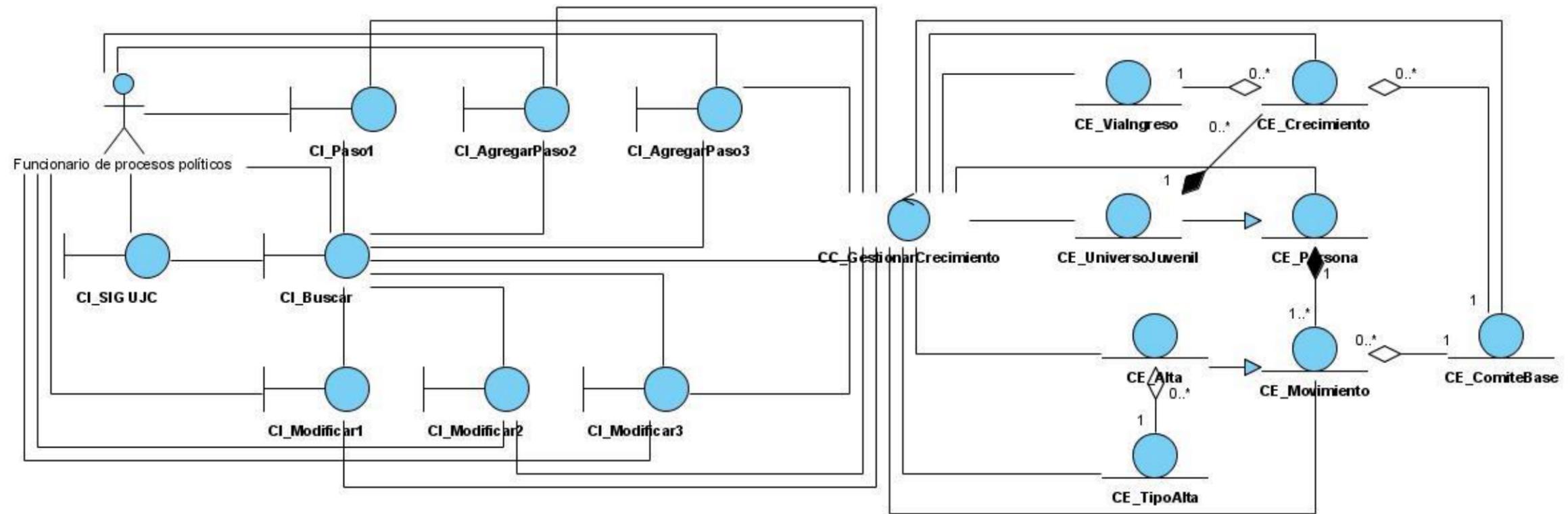
DCA CU_ Crear Expediente



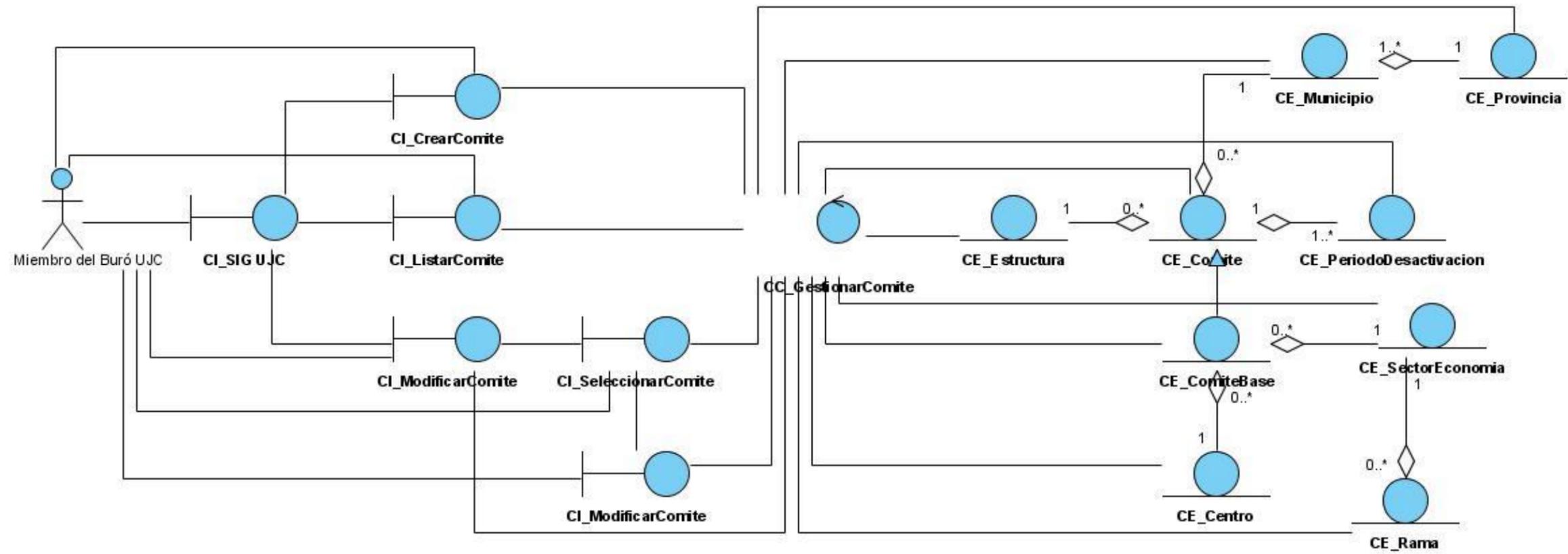
DCA CU_ Gestionar Centro



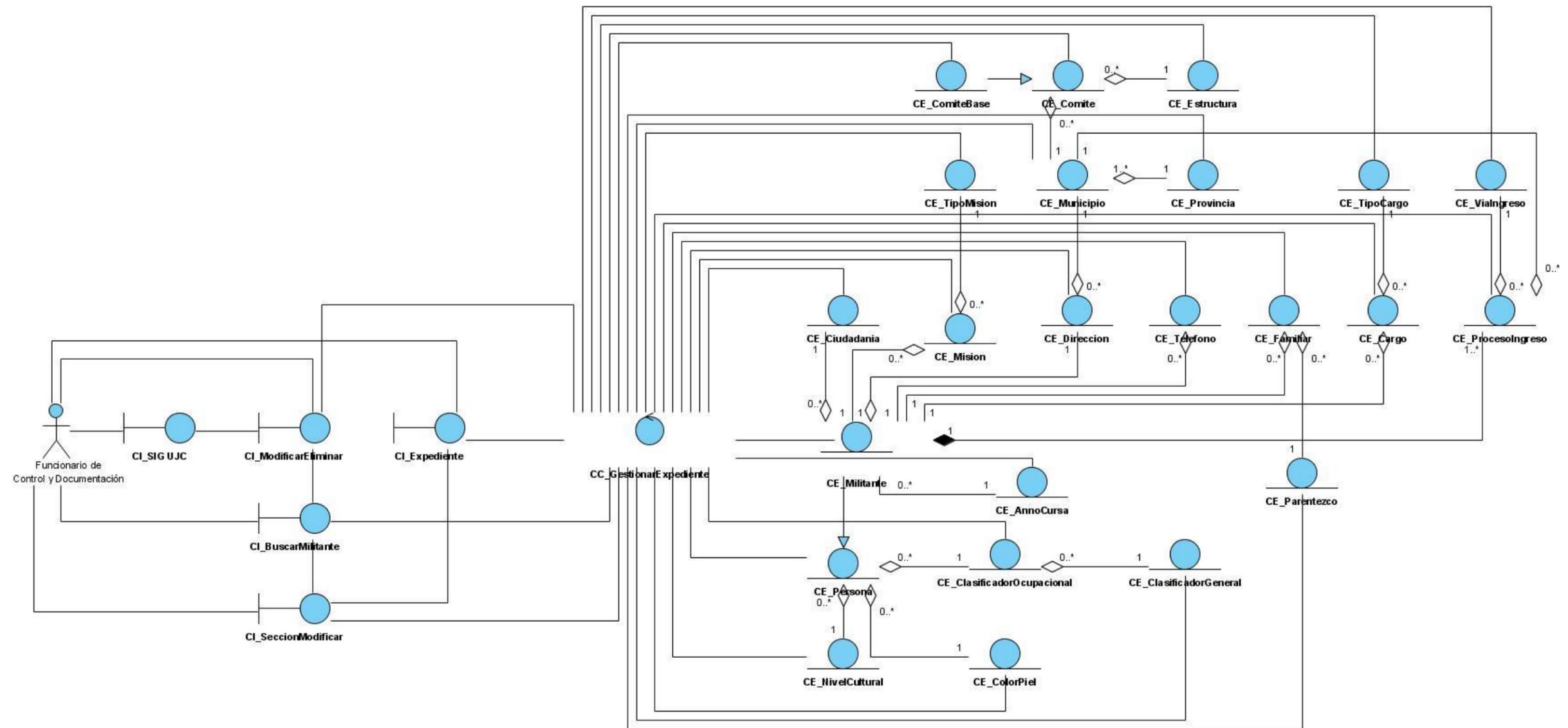
DCA CU_ Gestionar Crecimiento



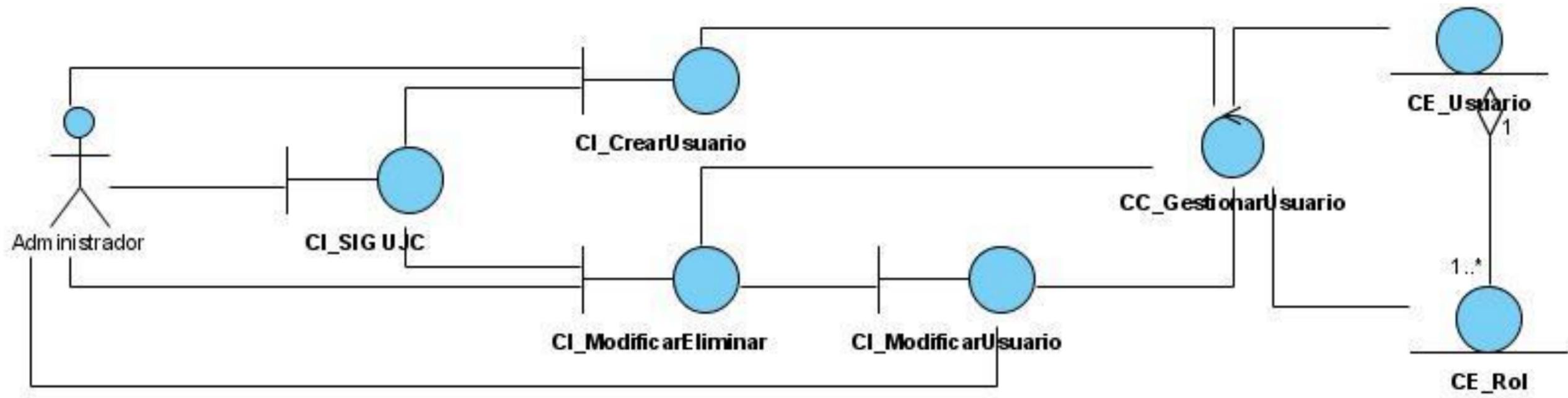
DCA CU_ Gestionar Estructura



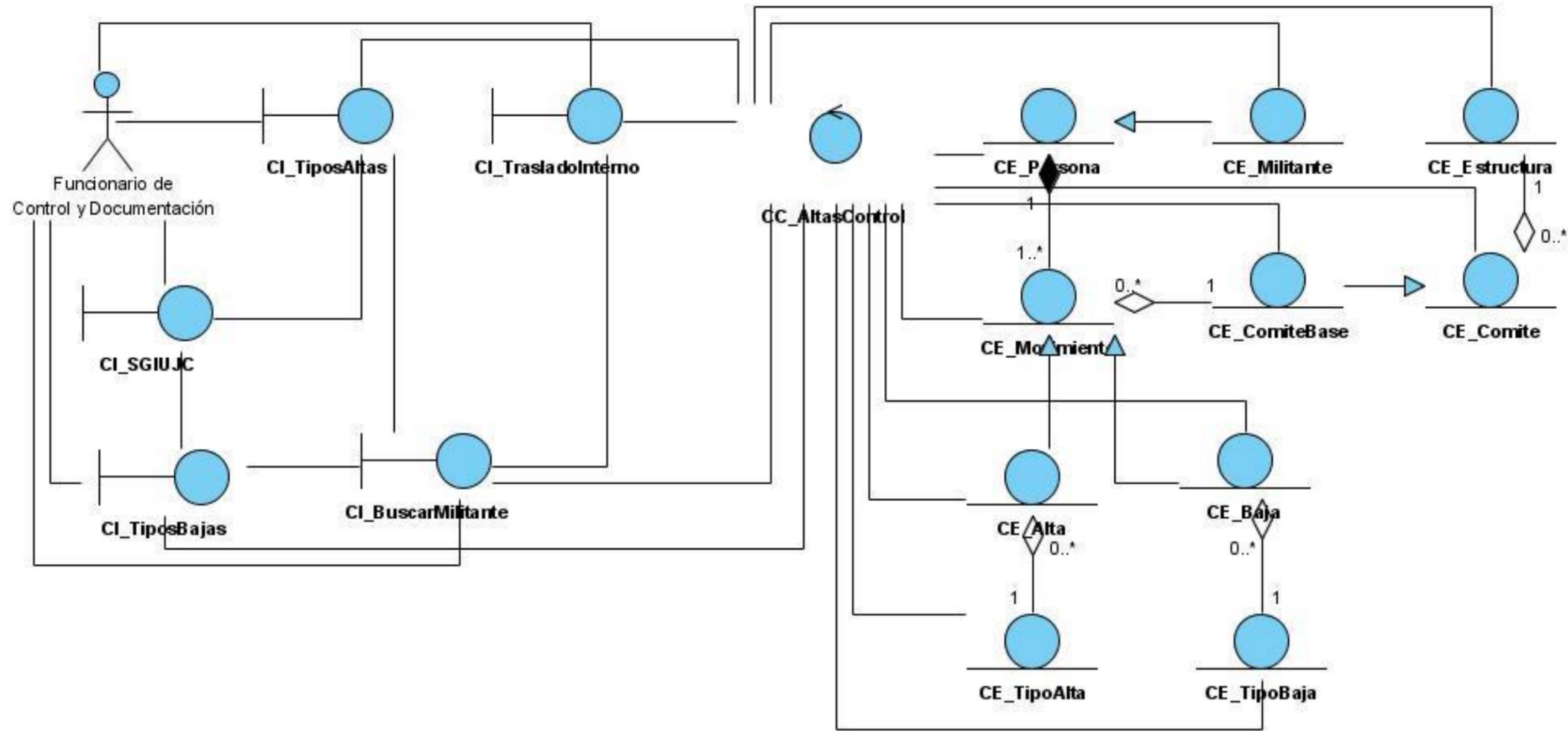
DCA CU_ Gestionar Expediente



DCA CU_ Gestionar Usuario



DCA CU_ Traslado Interno



Anexo 8. Diagramas de clases del diseño

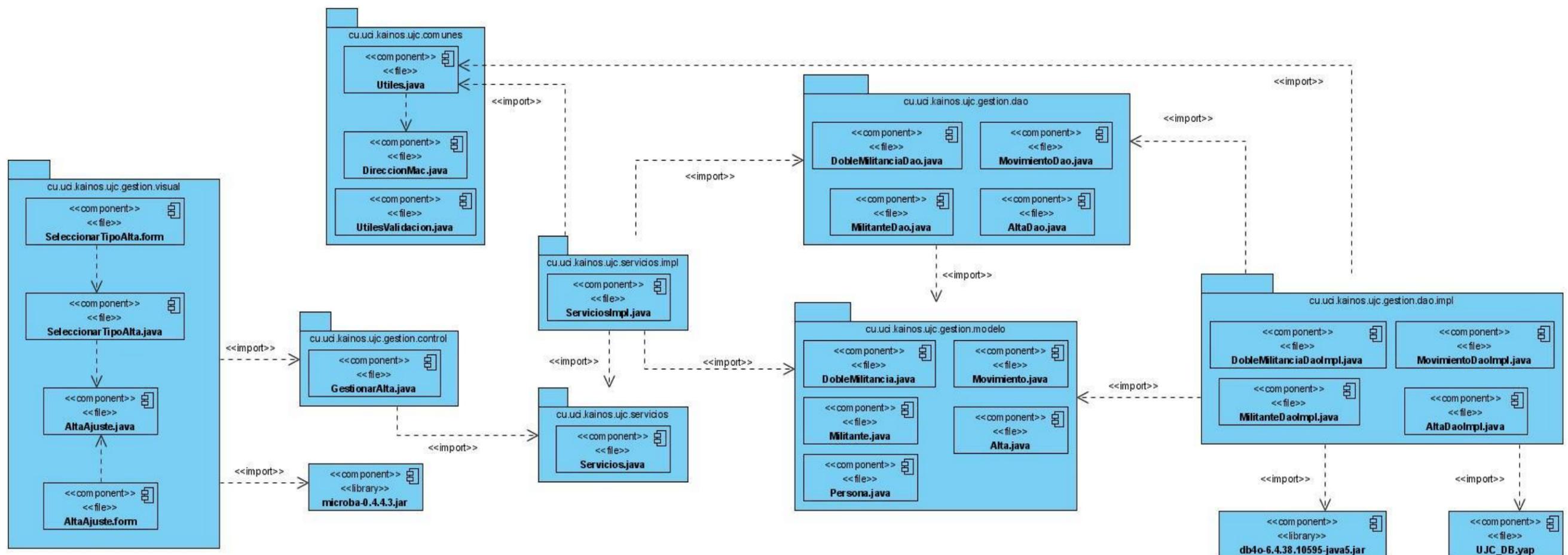
Los diagramas de clases del diseño obtenidos como resultado del análisis y diseño, pueden verse en el fichero del Visual Paradigm (.vpp) adjunto al documento.

Anexo 9. Diagramas de Secuencia

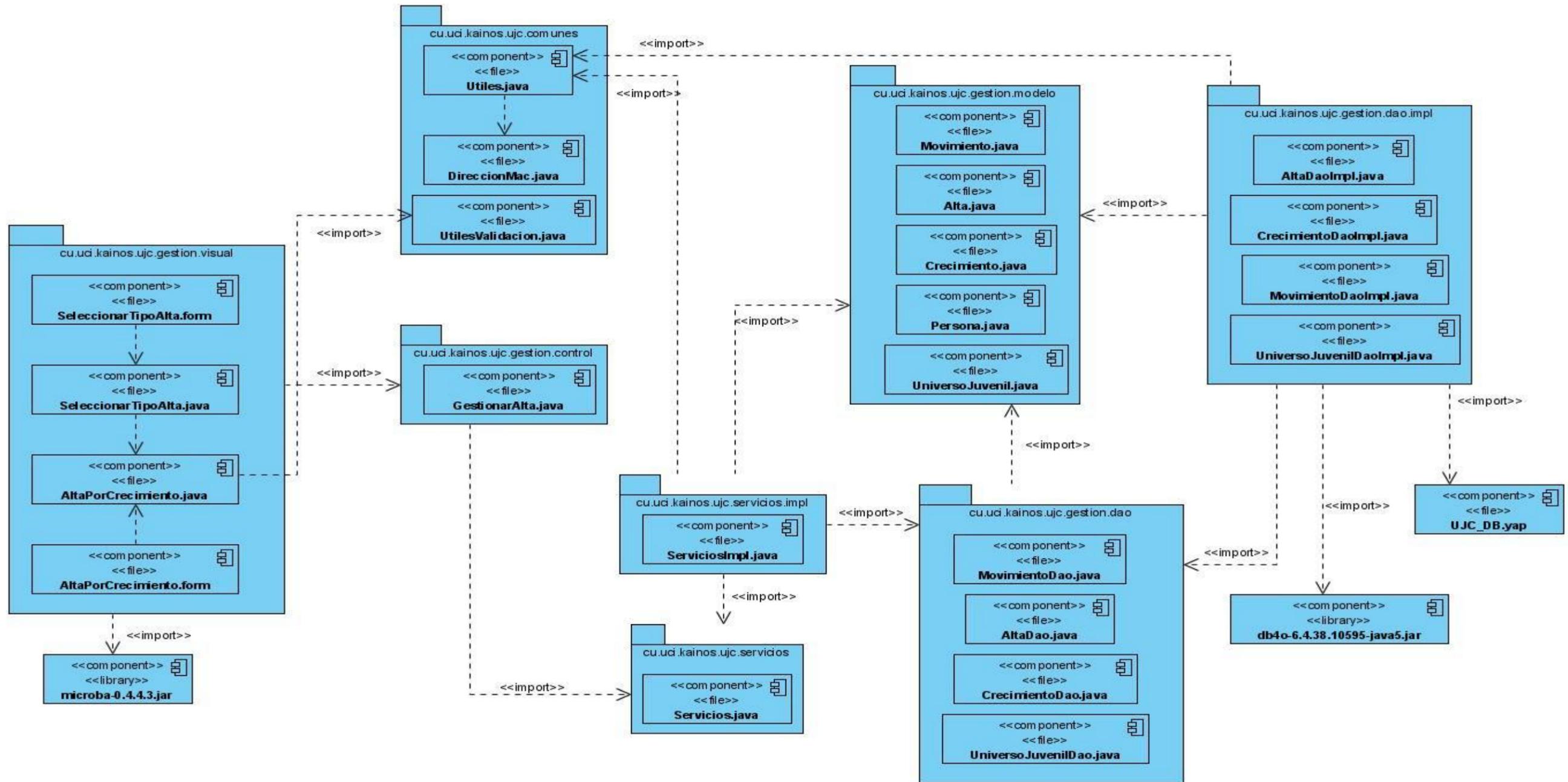
Los diagramas de secuencia obtenidos como resultado del análisis y diseño, pueden verse en el fichero del Visual Paradigm (.vpp) adjunto al documento.

Anexo 10. Diagramas de componente

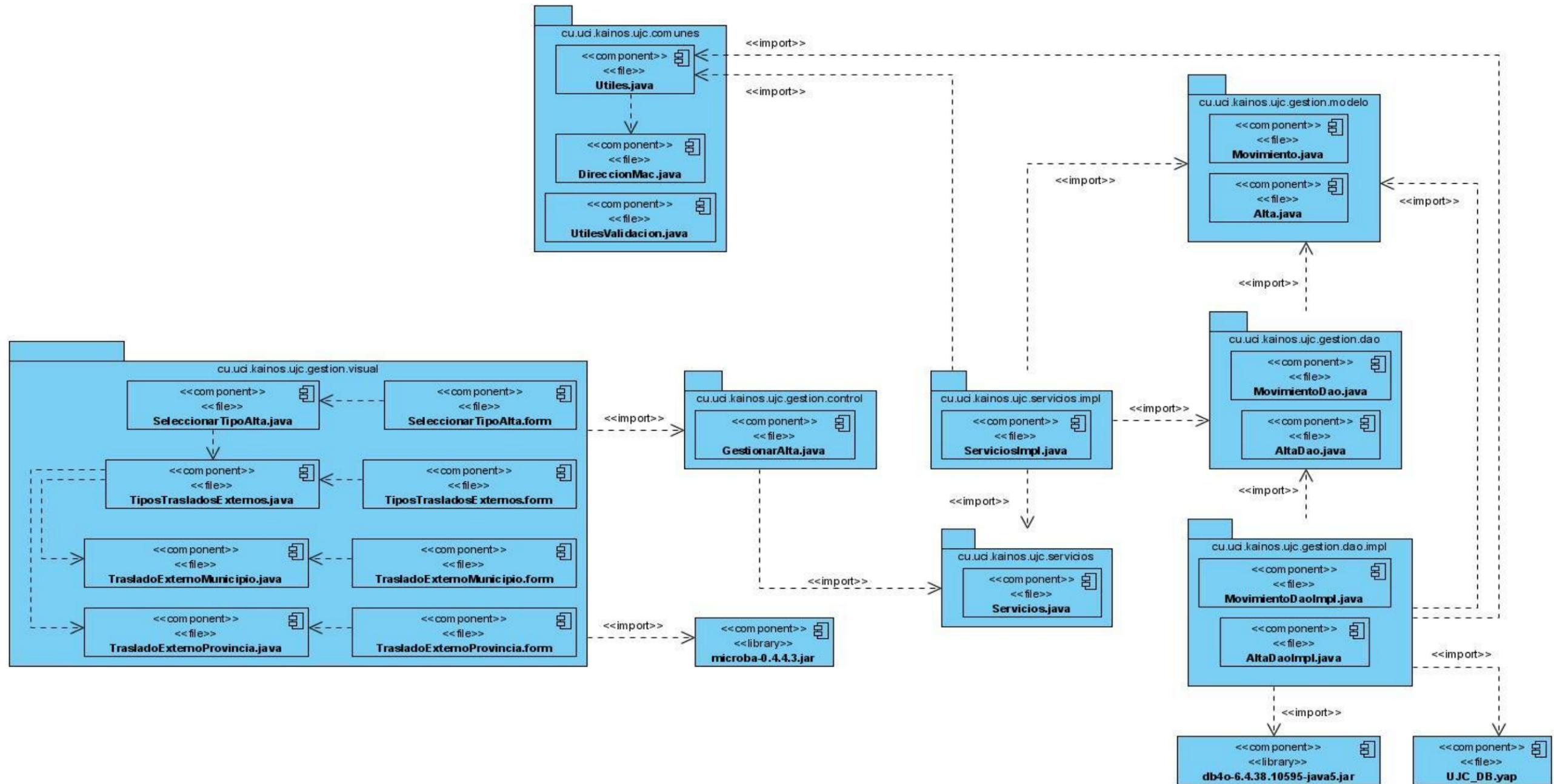
DCD CU_ Realizar Alta por Ajuste



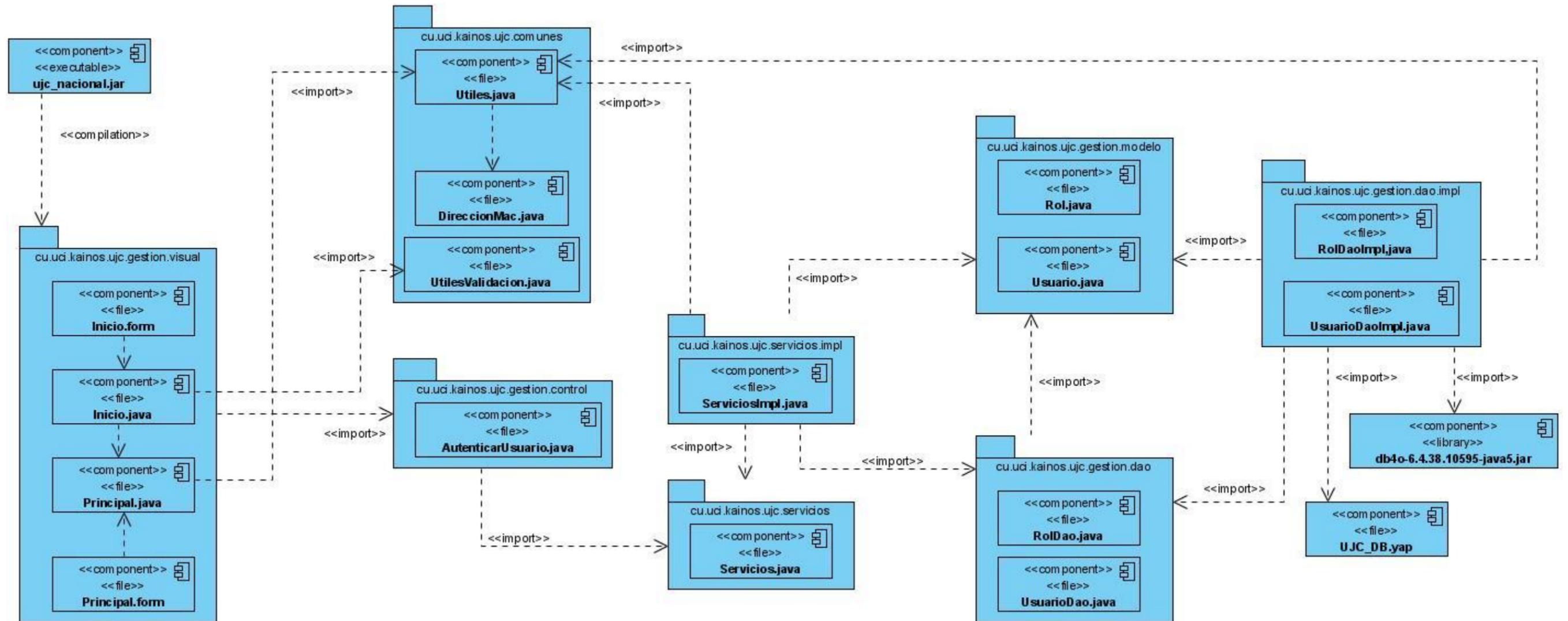
DCD CU_ Realizar Alta por Crecimiento



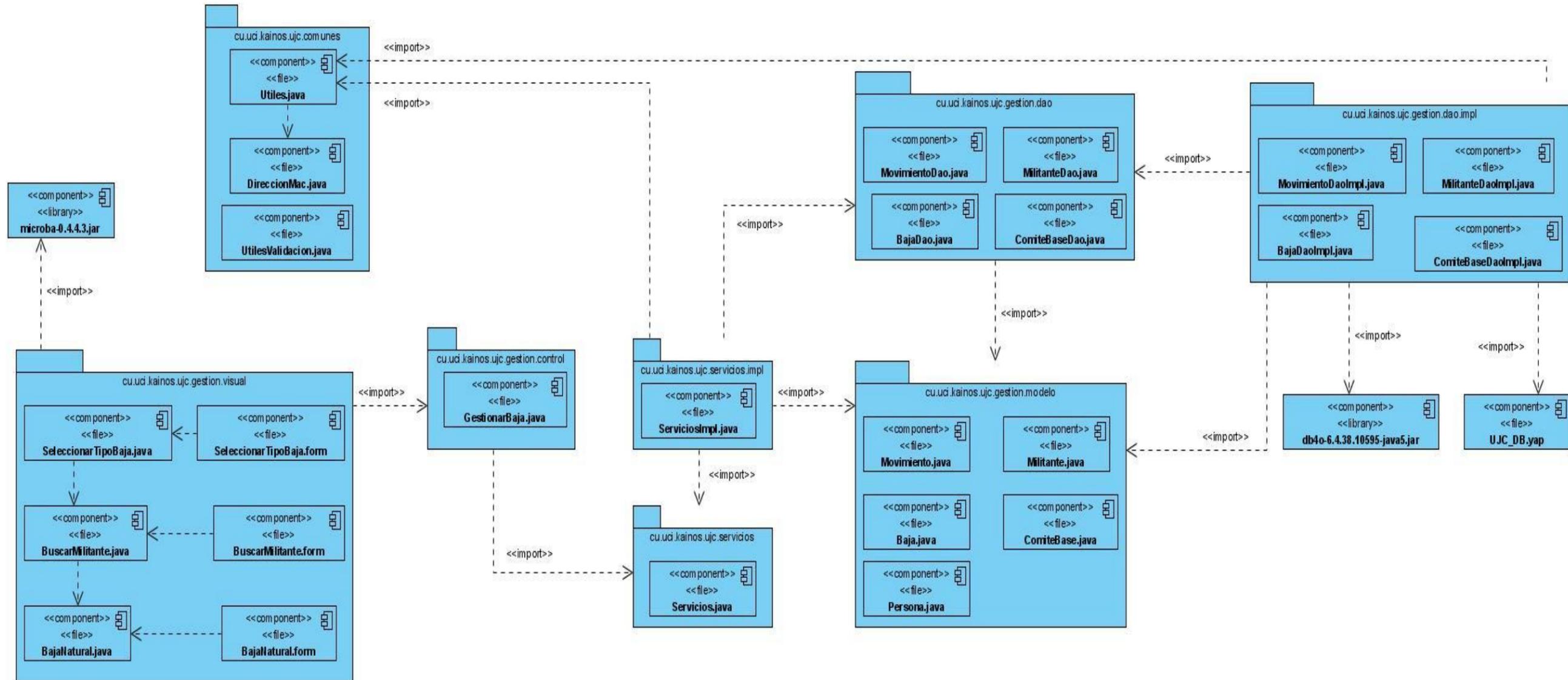
DCD CU_ Realizar Alta por Traslado Externo



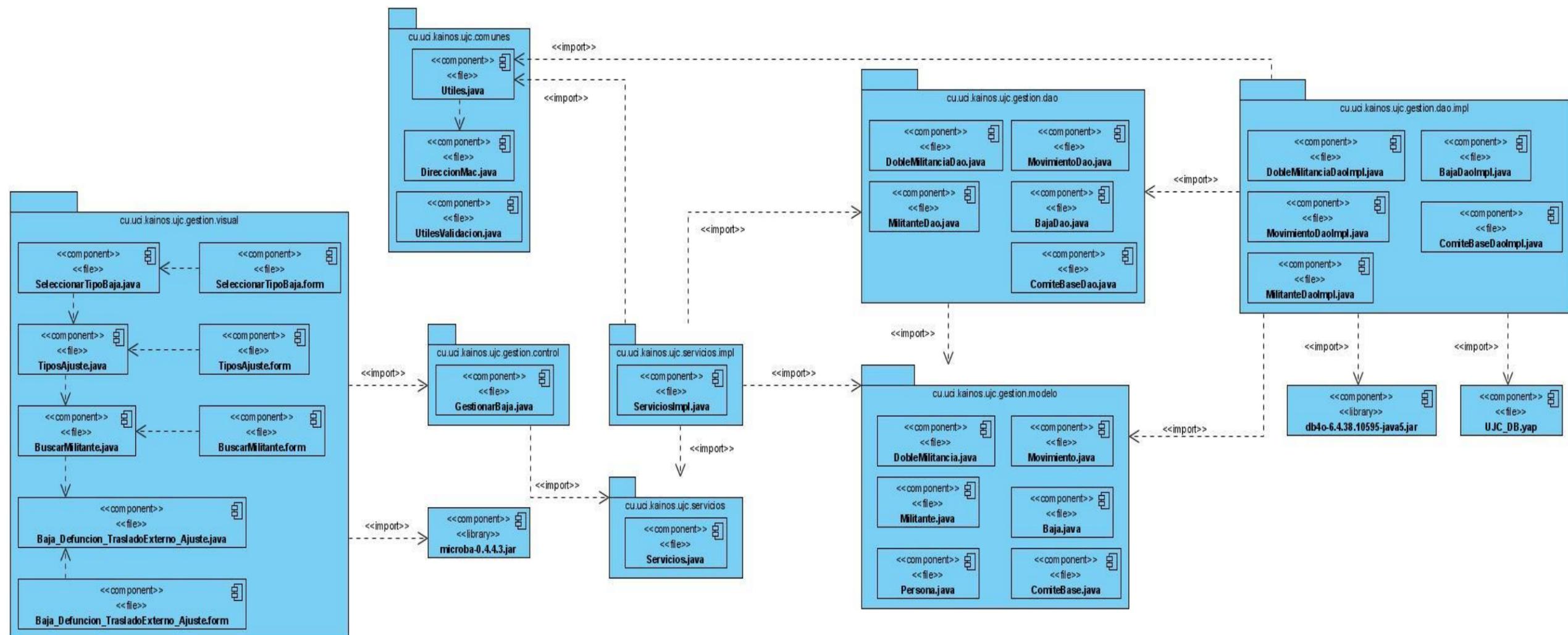
DCD CU_ Autenticar Usuario



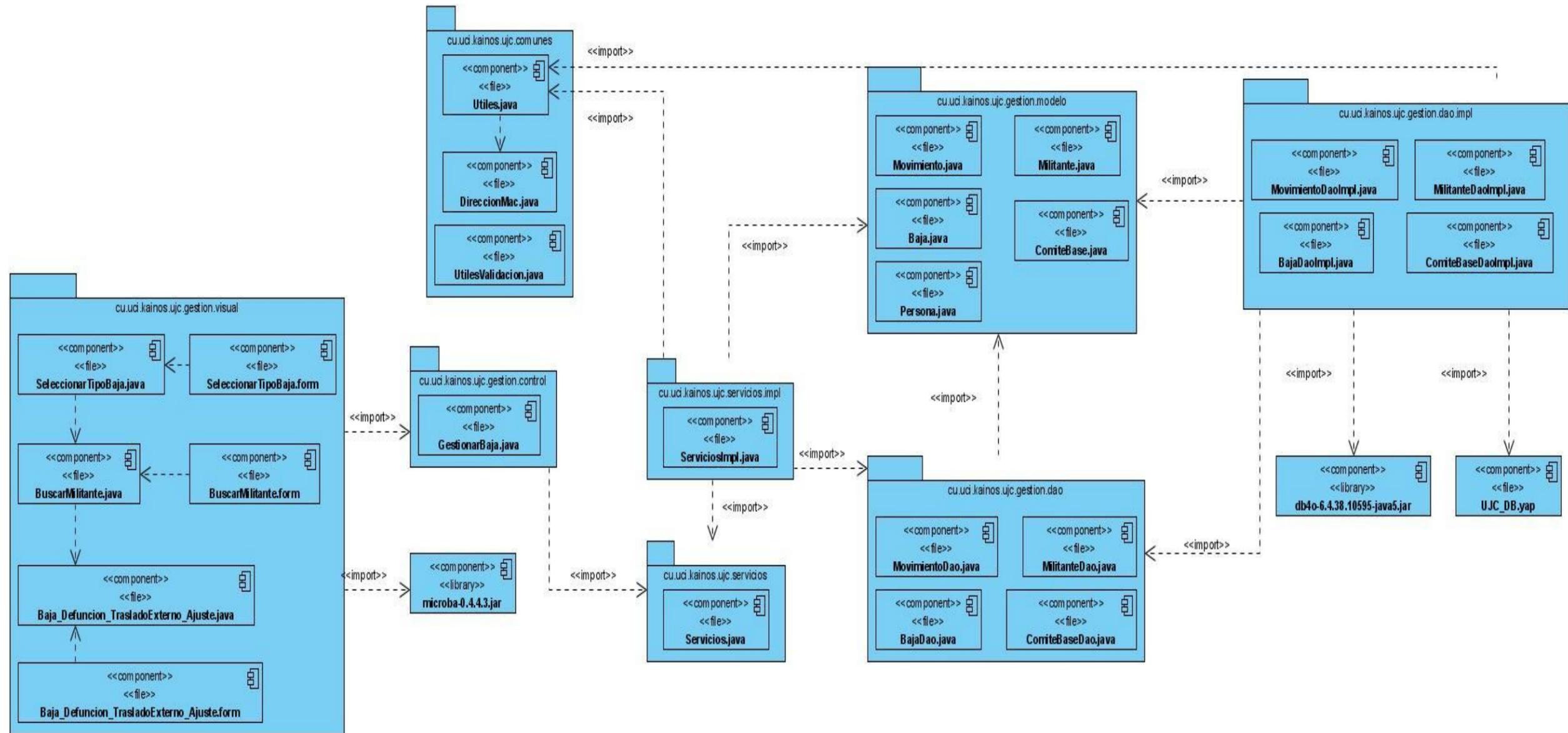
DCD CU_ Realizar Baja Natural



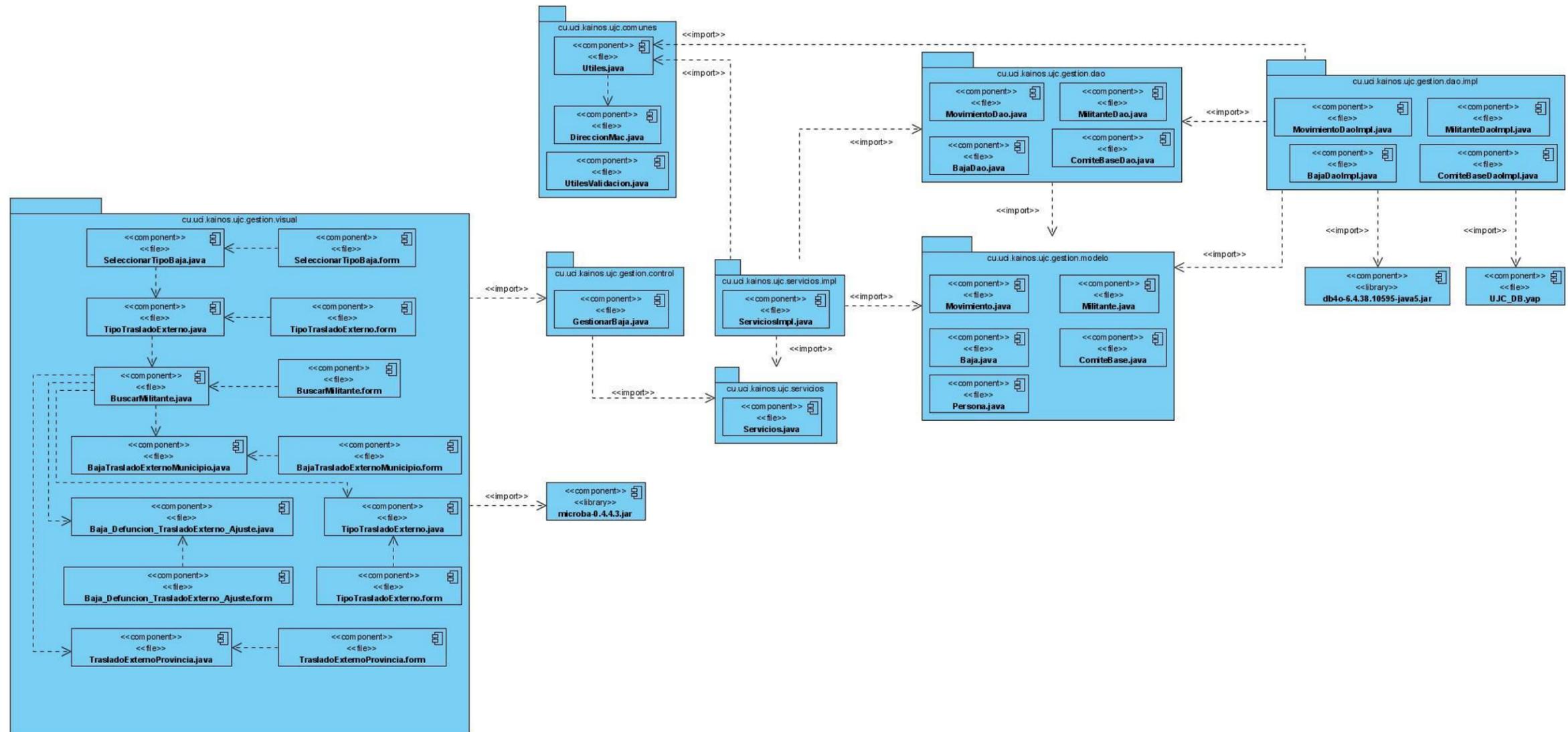
DCD CU_ Realizar Baja Ajuste



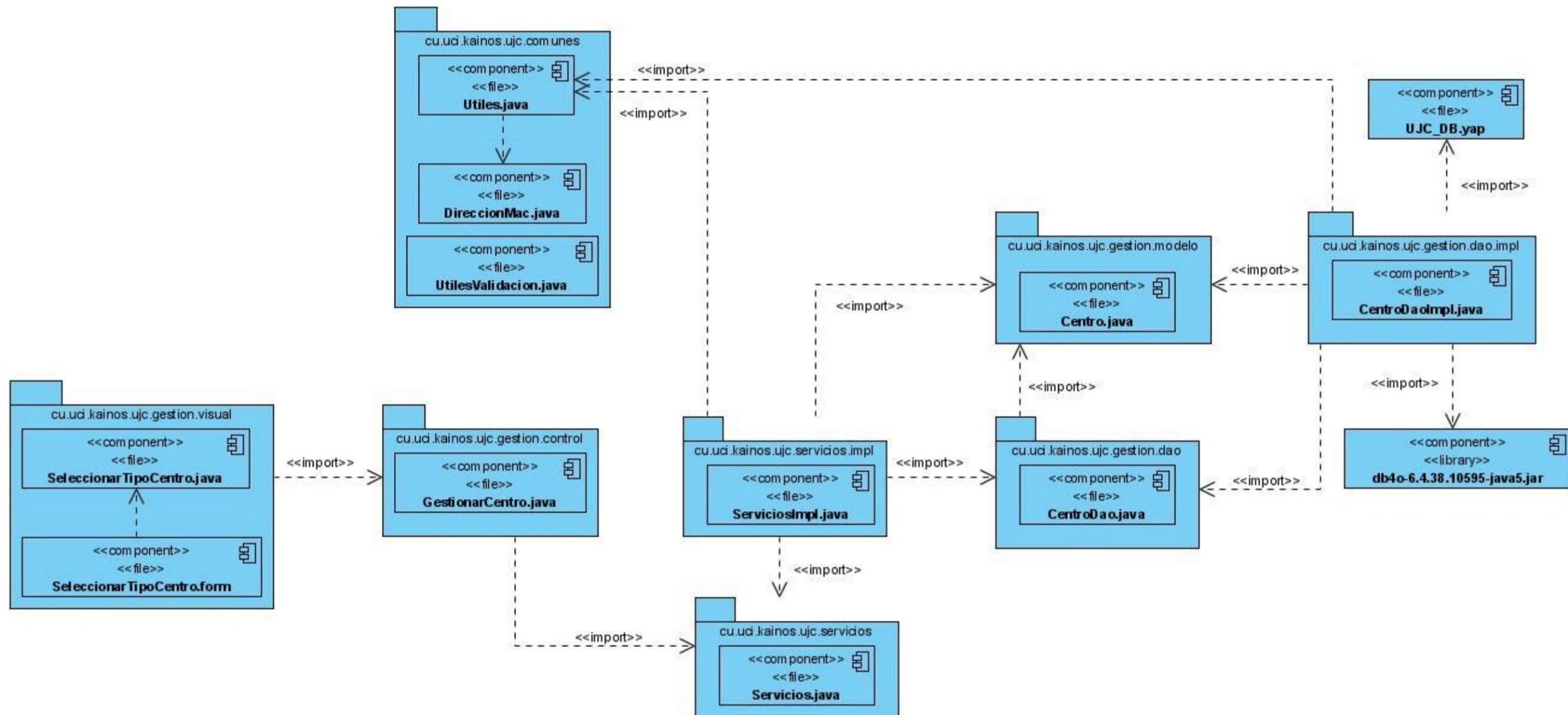
DCD CU_ Realizar Baja Defunción



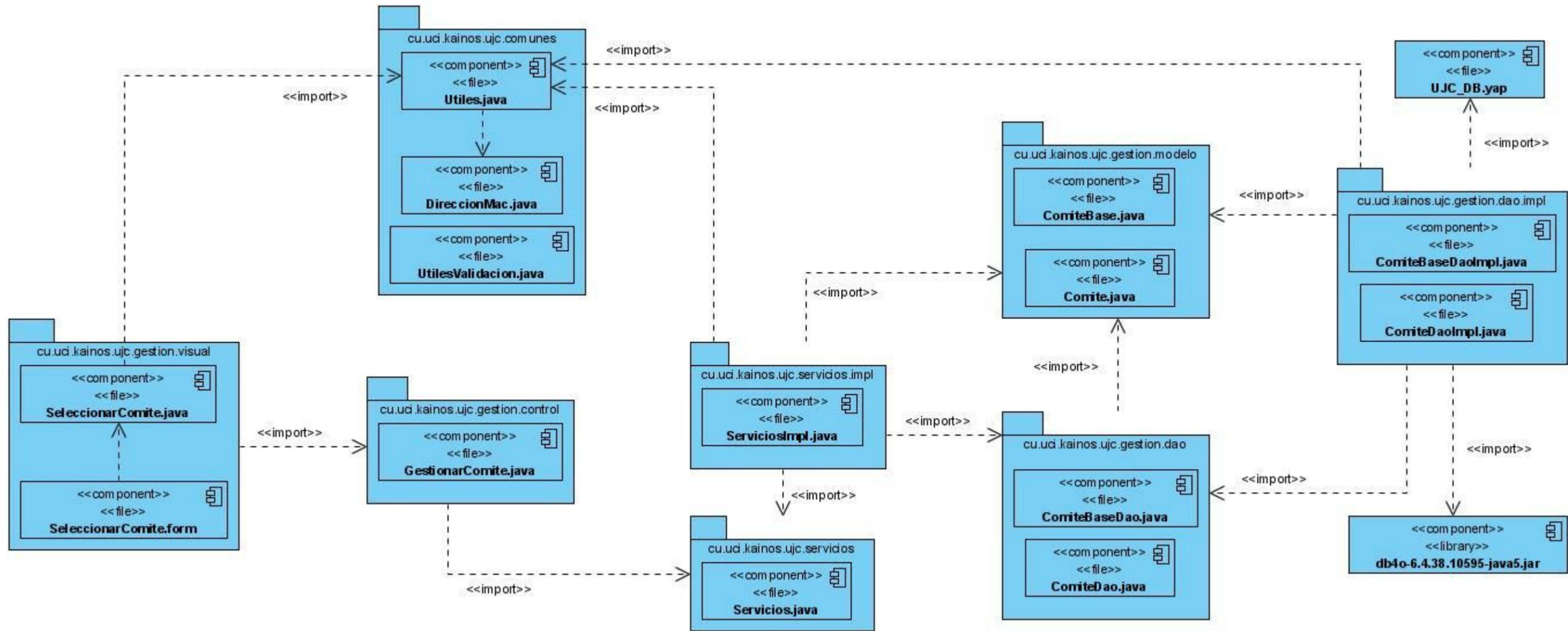
DCD CU_ Realizar Baja por Traslado Externo



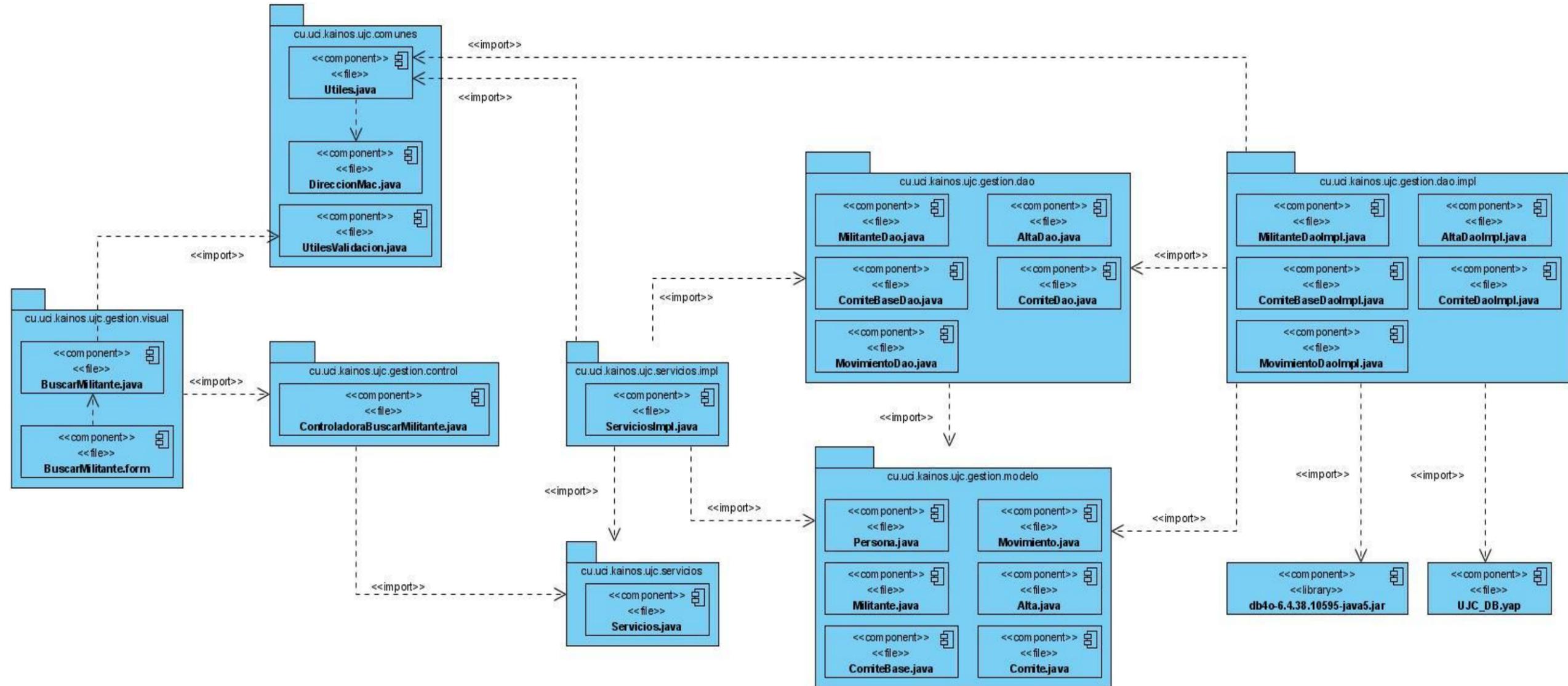
DCD CU_ Buscar Centro



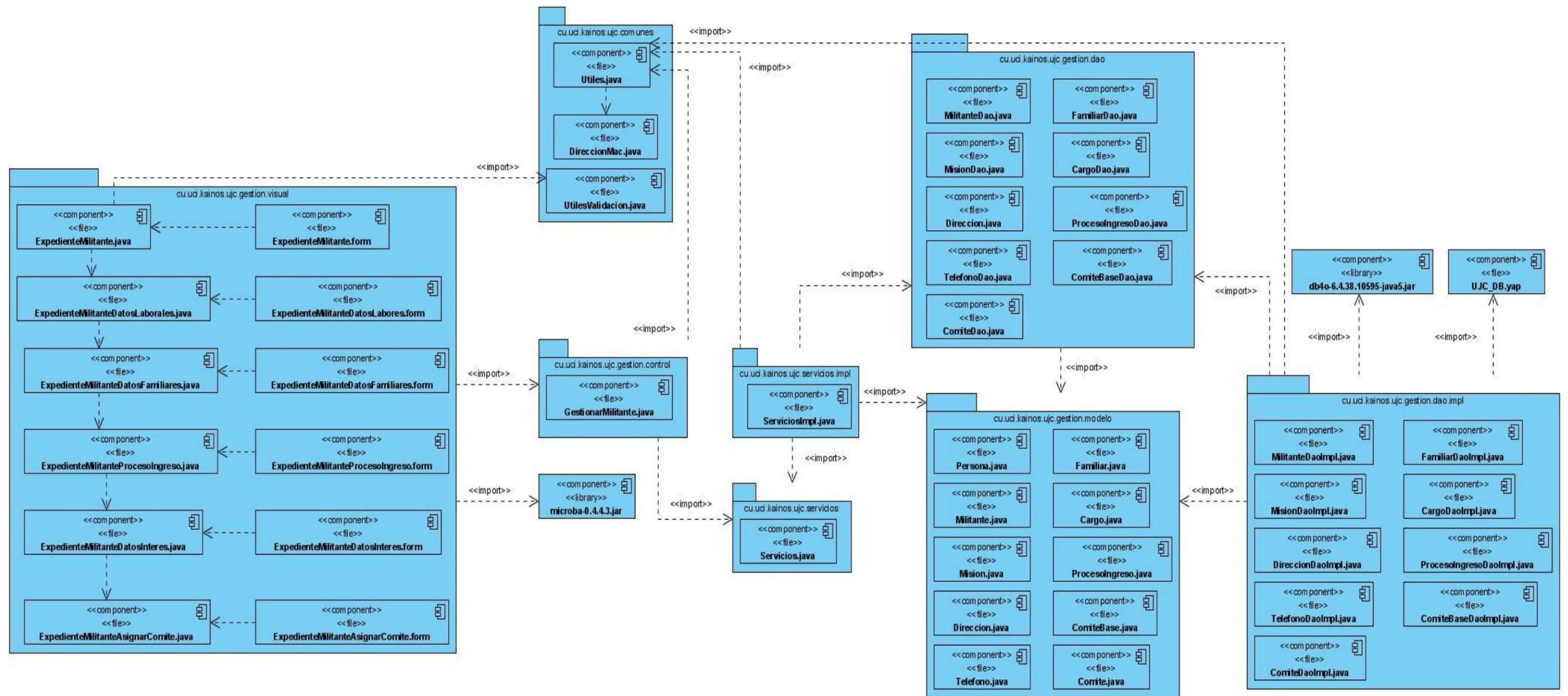
DCD CU_ Buscar Estructura



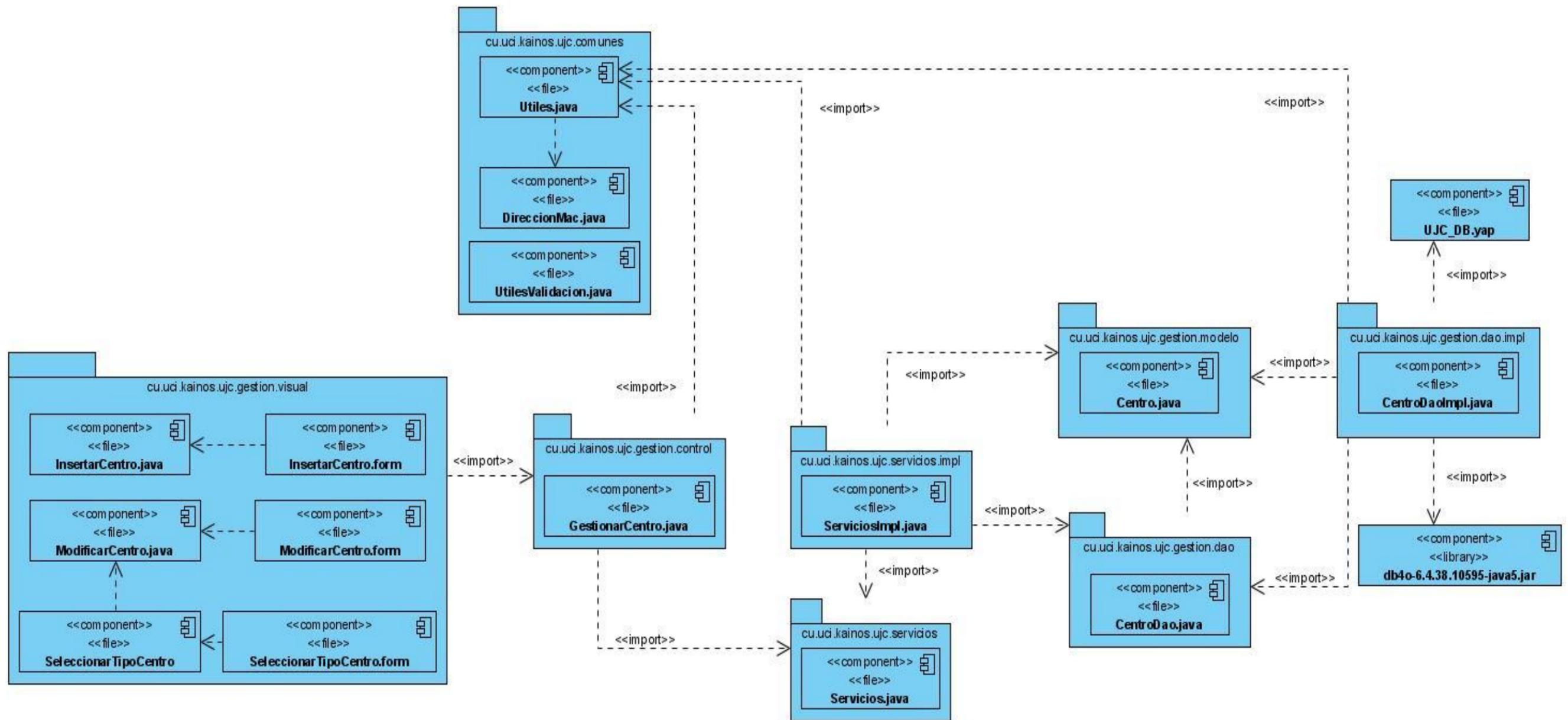
DCD CU_ Buscar Militante



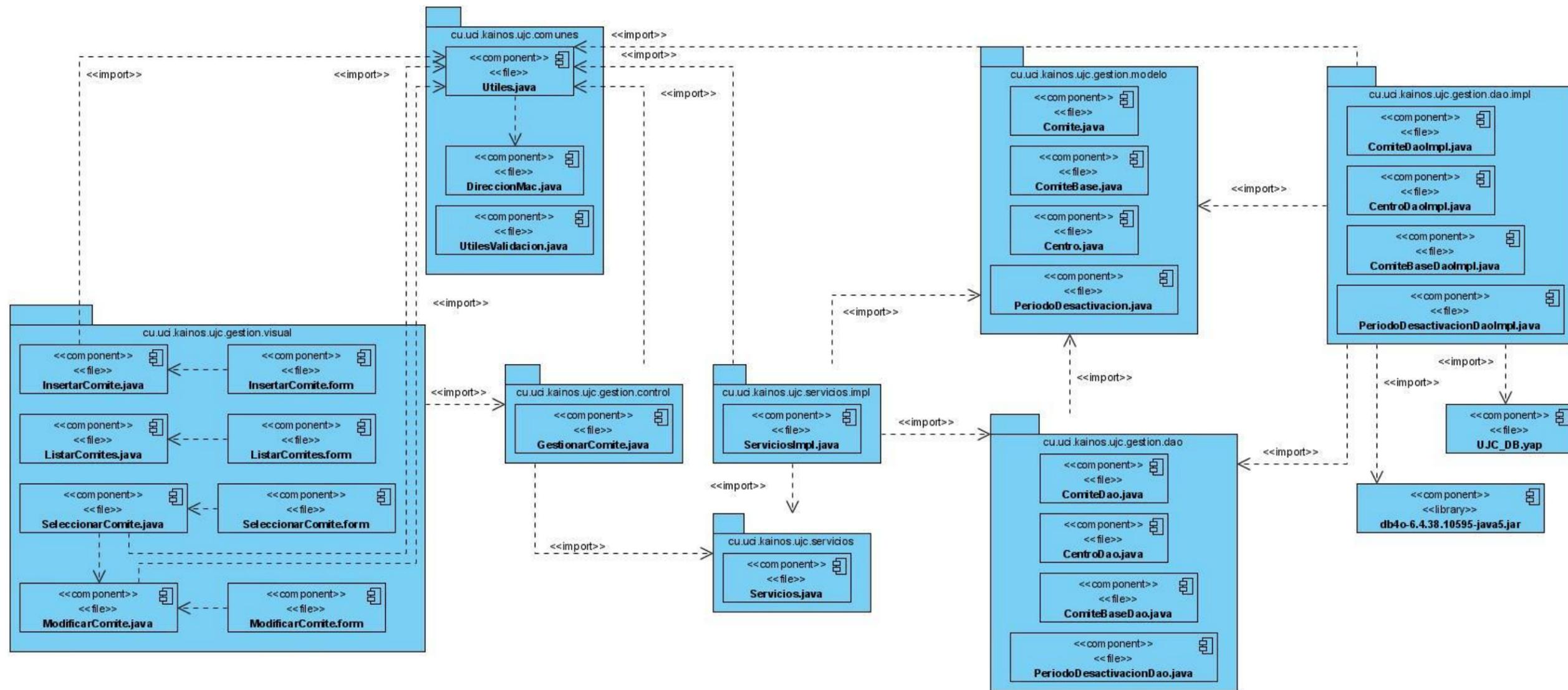
DCD CU_ Crear Expediente



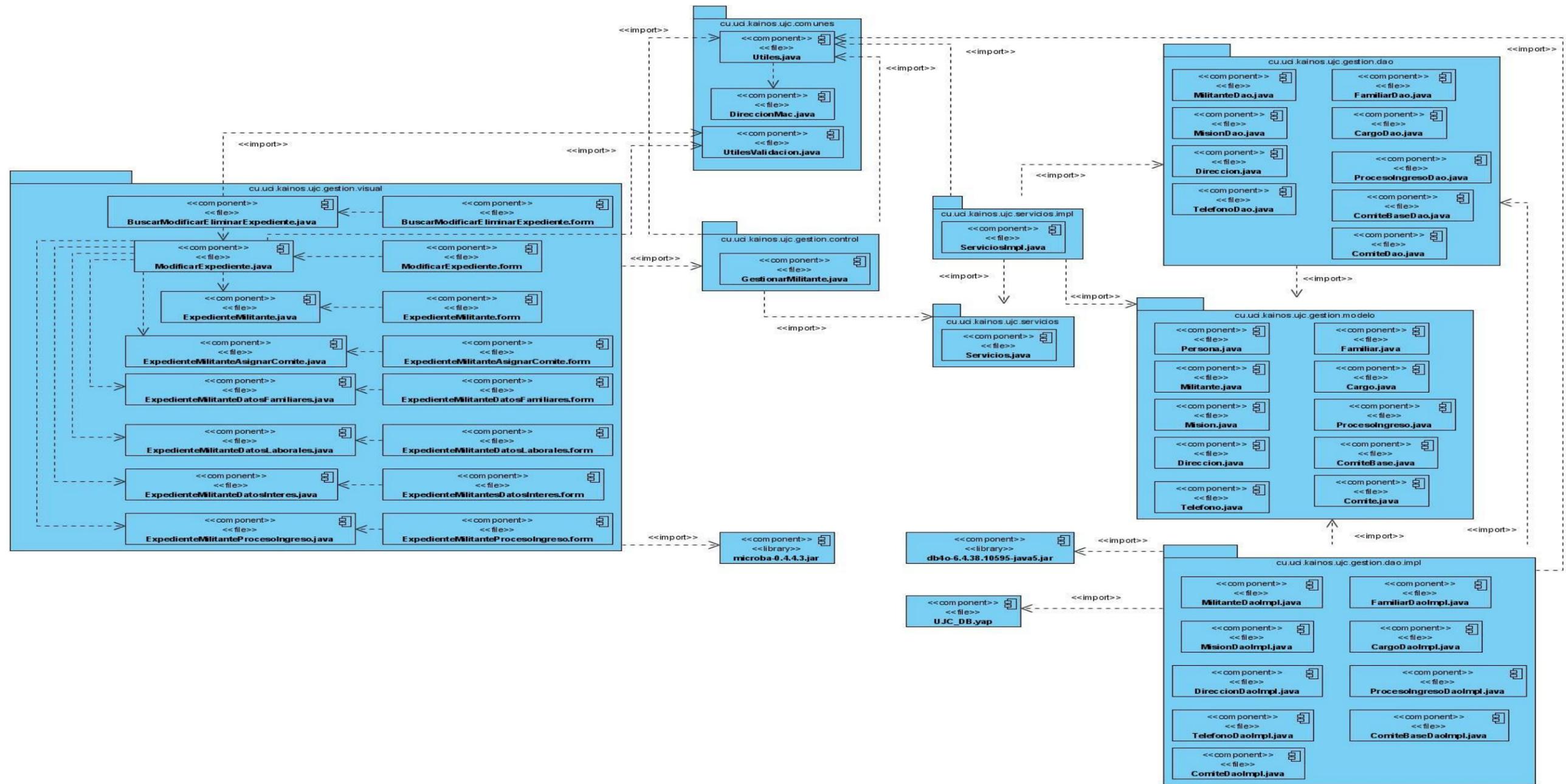
DCD CU_ Gestionar Centro



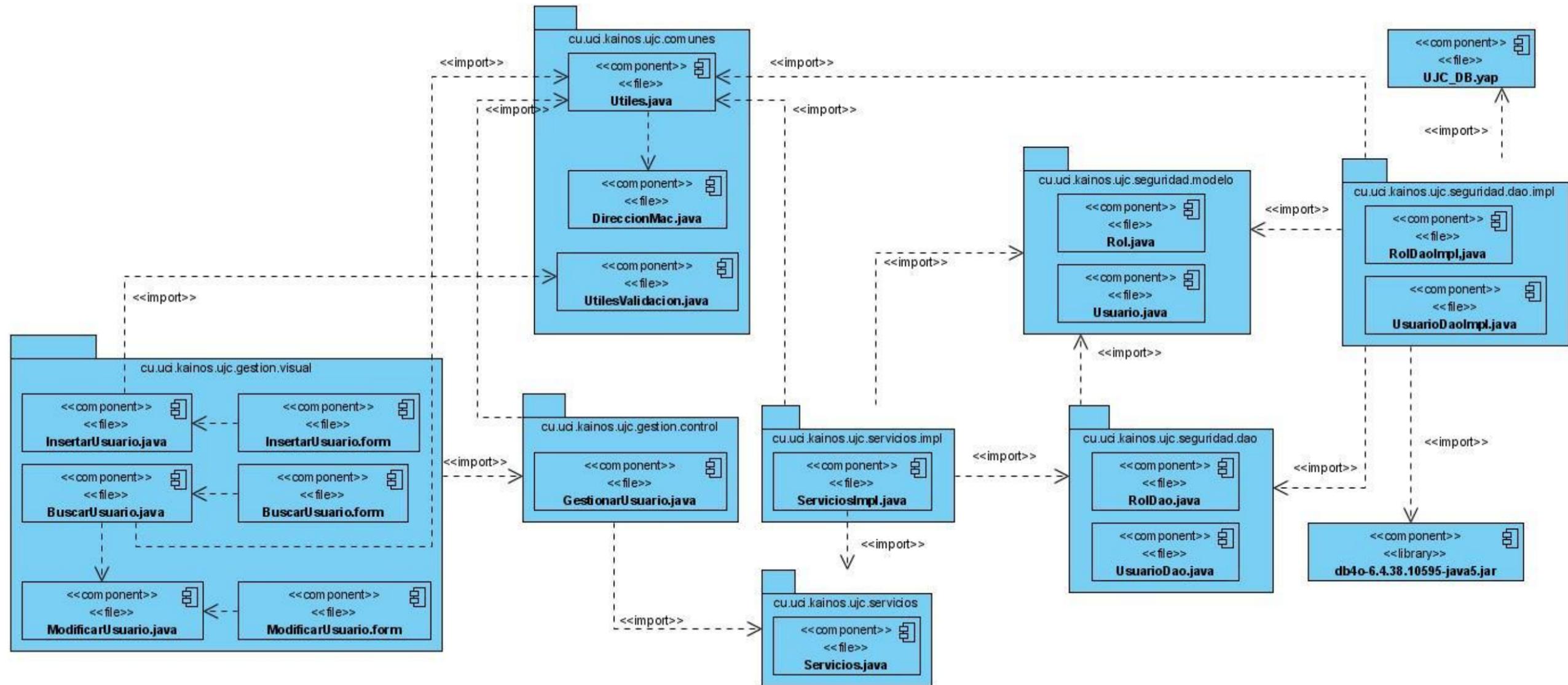
DCD CU_ Gestionar Estructura



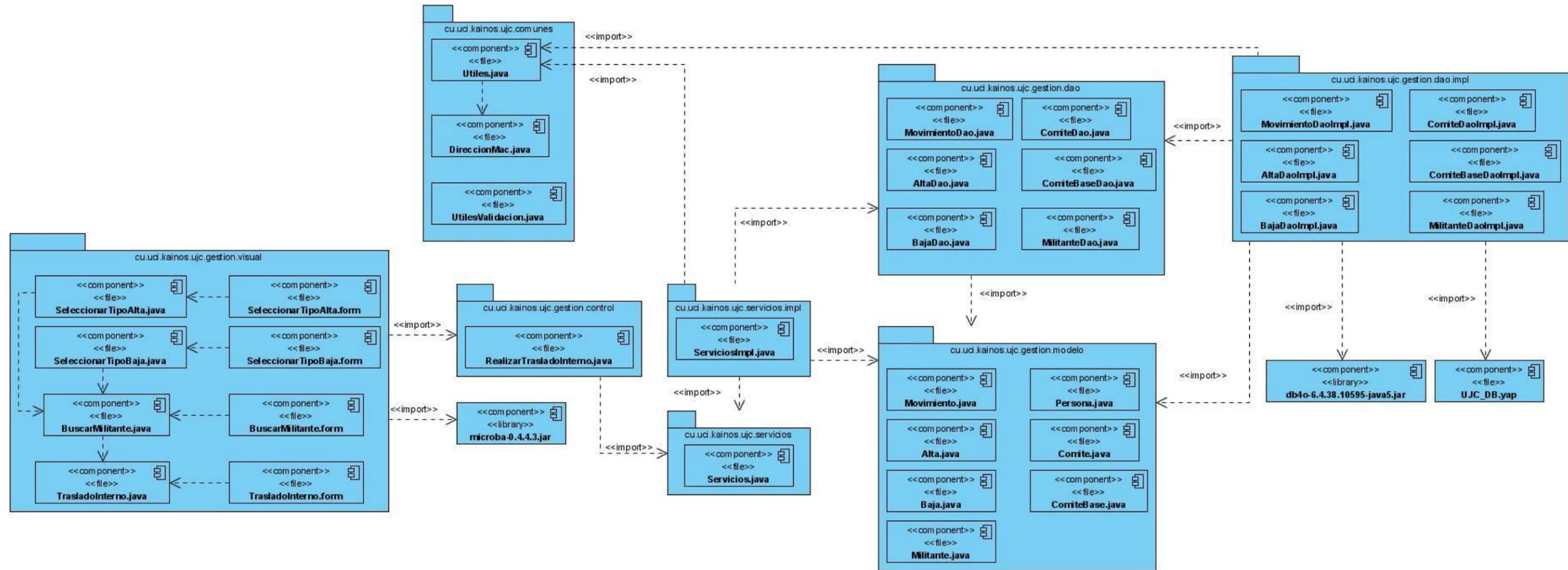
DCD CU_ Gestionar Expediente



DCD CU_ Gestionar Usuario



DCU_ Realizar Traslado Interno



Anexo 11. Casos de Prueba

Caso de Prueba CU: Autenticar Usuario

Condición de Entrada	Casos válidos	Casos no válidos
Usuario	Letras, números y caracteres extraños.	No introducir el usuario o que éste no corresponda con el almacenado en la base de datos.
Contraseña	Letras, números y caracteres extraños.	No introducir la contraseña o que ésta no corresponda con la almacenada en la base de datos.

Caso de uso: Autenticar Usuario
Caso de Prueba: Realizar gestiones de autenticación entrando correctamente usuario y contraseña.
Entrada: Usuario: yaiselita Contraseña: *****
Resultado: Se le da acceso al usuario en el sistema.
Condiciones: Los datos para autenticarse en el sistema deben estar en el rango de valores válidos.

Caso de uso: Autenticar Usuario
Caso de Prueba: Realizar gestiones de autenticación entrando incorrectamente la contraseña.
Entrada: Usuario: yaiselita

Contraseña: ***
Resultado: El sistema muestra un mensaje de error.
Condiciones: El usuario o la contraseña son incorrectos.

Caso de uso: Autenticar Usuario
Caso de Prueba: Realizar gestiones de autenticación dejando el campo Contraseña vacío.
Entrada: Usuario: yaiselita Contraseña:
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos.

Caso de uso: Autenticar Usuario
Caso de Prueba: Realizar gestiones de autenticación dejando el campo Usuario vacío.
Entrada: Usuario: Contraseña: *****
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos.

Caso de uso: Autenticar Usuario
Caso de Prueba: Realizar gestiones de autenticación introduciendo incorrectamente el usuario.
Entrada: Usuario: Yaisel Contraseña: *****
Resultado: El sistema muestra un mensaje de error.
Condiciones: El usuario o la contraseña son incorrectos.

Caso de Prueba CU: Gestionar Usuario

Sección Crear

Condición de Entrada	Casos válidos	Casos no válidos
Usuario	Letras, números y caracteres extraños.	No introducir el usuario o que éste no corresponda con el almacenado en la base de datos.
Contraseña	Letras, números y caracteres extraños.	No introducir la contraseña o que ésta no corresponda con la almacenada en la base de datos.
Repetir contraseña	Letras, números y caracteres extraños.	No repetir la contraseña o que ésta no corresponda con la almacenada en la base de datos.
Nombre completo	Letras y números.	No introducir el nombre completo o introducir caracteres extraños.
Roles	Seleccionable.	No seleccionar un rol.

Caso de uso: Gestionar Usuario
Caso de Prueba: Realizar gestiones de inserción de usuario entrando correctamente todos los datos.
Entrada: Usuario: yaramirez Contraseña: ***** Repetir contraseña: ***** Nombre completo: Yaisel Avilés Ramirez Roles: Administrador
Resultado: El sistema inserta los datos del usuario en la base de datos.
Condiciones: Los datos para insertar un usuario deben estar en el rango de valores válidos.

Caso de uso: Gestionar Usuario
Caso de Prueba: Realizar gestiones de inserción de usuario dejando el campo Usuario vacío.
Entrada: Usuario: Contraseña: ***** Repetir contraseña: ***** Nombre completo: Yaisel Avilés Ramirez Roles: Administrador

Resultado: El sistema muestra un mensaje de error.
Condiciones: Usuario o contraseña incorrectos.

Caso de uso: Gestionar Usuario
Caso de Prueba: Realizar gestiones de inserción de usuario dejando el campo Contraseña vacío.
Entrada: Usuario: yaramirez Contraseña: Repetir contraseña: ***** Nombre completo: Yaisel Avilés Ramirez Roles: Administrador
Resultado: El sistema muestra un mensaje de error.
Condiciones: Usuario o contraseña incorrectos.

Caso de uso: Gestionar Usuario
Caso de Prueba: Realizar gestiones de inserción de usuario entrando contraseñas diferentes.
Entrada: Usuario: yaramirez Contraseña: ***** Repetir contraseña: ***

Nombre completo: Yaisel Avilés Ramirez
Roles: Administrador
Resultado: El sistema muestra un mensaje de error.
Condiciones: Las contraseñas no coinciden.

Caso de uso: Gestionar Usuario
Caso de Prueba: Realizar gestiones de inserción de usuario dejando el campo Repetir contraseña vacío.
Entrada: Usuario: yaramirez Contraseña: ***** Repetir contraseña: Nombre completo: Yaisel Avilés Ramirez Roles: Administrador
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe repetir la contraseña.

Caso de uso: Gestionar Usuario
Caso de Prueba: Realizar gestiones de inserción de usuario entrando incorrectamente el nombre completo.
Entrada: Usuario: yaramirez

Contraseña: *****
Repetir contraseña: *****
Nombre completo: Yaisel_Avilés&Ramirez
Roles: Administrador
Resultado: El sistema muestra un mensaje de error.
Condiciones: El nombre completo no puede contener caracteres extraños.

Caso de uso: Gestionar Usuario
Caso de Prueba: Realizar gestiones de inserción de usuario dejando el campo Nombre completo vacío.
Entrada: Usuario: yaramirez Contraseña: ***** Repetir contraseña: ***** Nombre completo: Roles: Administrador
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Gestionar Usuario
Caso de Prueba: Realizar gestiones de inserción de usuario dejando el campo Roles vacío.
<p>Entrada:</p> <p>Usuario: yaramirez</p> <p>Contraseña: *****</p> <p>Repetir contraseña: *****</p> <p>Nombre completo: Yaisel Avilés Ramirez</p> <p>Roles:</p>
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Sección Modificar o Eliminar

Para realizar algunas de estas operaciones primeramente se realizará la búsqueda del usuario deseado y luego se modificarán sus datos (el nombre de usuario no puede ser modificado) o se eliminarán del sistema, los siguientes casos de prueba están referidos a la búsqueda del usuario.

Condición de Entrada	Casos válidos	Casos no válidos
Rol	Seleccionable.	No seleccionar un rol.

Caso de uso: Gestionar Usuario
Caso de Prueba: Realizar búsqueda del usuario seleccionando un rol.
Entrada: Rol: Administrador
Resultado: El sistema muestra un listado de todos los usuarios que coincidan con el parámetro entrado.
Condiciones: Los datos para buscar un usuario deben estar en el rango de valores válidos.

Caso de uso: Gestionar Usuario
Caso de Prueba: Realizar búsqueda del usuario dejando el campo Rol vacío.
Entrada: Rol:
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe seleccionar un rol para realizar la búsqueda.

Caso de Prueba CU: Gestionar Estructura

Sección Crear

Condición de Entrada	Casos válidos	Casos no válidos
Tipo de Comité	Seleccionable.	No seleccionar un tipo de comité.
Subordinado a	Seleccionable.	No seleccionar una de las opciones.
Provincia	Seleccionable.	No seleccionar una provincia.

Municipio	Seleccionable.	No seleccionar un municipio.
Nombre	Letras y números.	No introducir el nombre o introducir caracteres extraños.
Sector de la Economía	Seleccionable.	No seleccionar un sector de la economía.
Rama	Seleccionable.	No seleccionar una rama.
Tipo de Centro	Seleccionable.	No seleccionar un tipo de centro.
Nombre del Centro	Seleccionable.	No seleccionar un nombre de centro.

Caso de uso: Gestionar Estructura
Caso de Prueba: Realizar gestiones de inserción de estructura introduciendo correctamente todos los datos.
Entrada: Tipo de Comité: Comité de Base Independiente Subordinado a: Comité Especial UJC Provincia: Guantánamo Municipio: Manuel Tames Nombre: CBI Los Industrialistas Sector de la Economía: Industria Rama: Industria Básica Tipo de Centro: Universidad

Nombre del Centro: Universidad de Oriente
Resultado: El sistema inserta una nueva estructura en la base de datos.
Condiciones: Los datos para insertar una estructura deben estar en el rango de valores válidos.

Caso de uso: Gestionar Estructura
Caso de Prueba: Realizar gestiones de inserción de estructura dejando el campo Tipo de Comité vacío.
Entrada: Tipo de Comité: Subordinado a: Comité Especial UJC Provincia: Guantánamo Municipio: Manuel Tames Nombre: CBI Los Industrialistas Sector de la Economía: Industria Rama: Industria Básica Tipo de Centro: Universidad Nombre del Centro: Universidad de Oriente
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Gestionar Estructura
Caso de Prueba: Realizar gestiones de inserción de estructura dejando el campo Subordinado a vacío.
Entrada: Tipo de Comité: Comité de Base Independiente Subordinado a: Provincia: Guantánamo Municipio: Manuel Tames Nombre: CBI Los Industrialistas Sector de la Economía: Industria Rama: Industria Básica Tipo de Centro: Universidad Nombre del Centro: Universidad de Oriente
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Gestionar Estructura
Caso de Prueba: Realizar gestiones de inserción de estructura dejando el campo Provincia vacío.
Entrada: Tipo de Comité: Comité de Base Independiente Subordinado a: Comité Especial UJC

<p>Provincia:</p> <p>Municipio: Manuel Tames</p> <p>Nombre: CBI Los Industrialistas</p> <p>Sector de la Economía: Industria</p> <p>Rama: Industria Básica</p> <p>Tipo de Centro: Universidad</p> <p>Nombre del Centro: Universidad de Oriente</p>
<p>Resultado: El sistema muestra un mensaje de error.</p>
<p>Condiciones: Debe llenar todos los campos obligatorios.</p>

<p>Caso de uso: Gestionar Estructura</p>
<p>Caso de Prueba: Realizar gestiones de inserción de estructura dejando el campo Municipio vacío.</p>
<p>Entrada:</p> <p>Tipo de Comité: Comité de Base Independiente</p> <p>Subordinado a: Comité Especial UJC</p> <p>Provincia: Guantánamo</p> <p>Municipio:</p> <p>Nombre: CBI Los Industrialistas</p> <p>Sector de la Economía: Industria</p> <p>Rama: Industria Básica</p>

Tipo de Centro: Universidad
Nombre del Centro: Universidad de Oriente
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Gestionar Estructura
Caso de Prueba: Realizar gestiones de inserción de estructura dejando el campo Nombre vacío.
Entrada:
Tipo de Comité: Comité de Base Independiente
Subordinado a: Comité Especial UJC
Provincia: Guantánamo
Municipio: Manuel Tames
Nombre:
Sector de la Economía: Industria
Rama: Industria Básica
Tipo de Centro: Universidad
Nombre del Centro: Universidad de Oriente
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Gestionar Estructura
Caso de Prueba: Realizar gestiones de inserción de estructura introduciendo incorrectamente el nombre.
Entrada: Tipo de Comité: Comité de Base Independiente Subordinado a: Comité Especial UJC Provincia: Guantánamo Municipio: Manuel Tames Nombre: CBI Los Industrialist@s Sector de la Economía: Industria Rama: Industria Básica Tipo de Centro: Universidad Nombre del Centro: Universidad de Oriente
Resultado: El sistema muestra un mensaje de error.
Condiciones: El nombre no puede contener caracteres extraños.

Caso de uso: Gestionar Estructura
Caso de Prueba: Realizar gestiones de inserción de estructura dejando el campo Sector de la Economía vacío.
Entrada: Tipo de Comité: Comité de Base Independiente Subordinado a: Comité Especial UJC

Provincia: Guantánamo
Municipio: Manuel Tames
Nombre: CBI Los Industrialistas
Sector de la Economía:
Rama: Industria Básica
Tipo de Centro: Universidad
Nombre del Centro: Universidad de Oriente
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Gestionar Estructura
Caso de Prueba: Realizar gestiones de inserción de estructura dejando el campo Rama vacío.
Entrada:
Tipo de Comité: Comité de Base Independiente
Subordinado a: Comité Especial UJC
Provincia: Guantánamo
Municipio: Manuel Tames
Nombre: CBI Los Industrialistas
Sector de la Economía: Industria
Rama:

Tipo de Centro: Universidad
Nombre del Centro: Universidad de Oriente
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Gestionar Estructura
Caso de Prueba: Realizar gestiones de inserción de estructura dejando el campo Tipo de Centro vacío.
Entrada: Tipo de Comité: Comité de Base Independiente Subordinado a: Comité Especial UJC Provincia: Guantánamo Municipio: Manuel Tames Nombre: CBI Los Industrialistas Sector de la Economía: Industria Rama: Industria Básica Tipo de Centro: Nombre del Centro: Universidad de Oriente
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Gestionar Estructura
Caso de Prueba: Realizar gestiones de inserción de estructura dejando el campo Nombre del Centro vacío.
Entrada: Tipo de Comité: Comité de Base Independiente Subordinado a: Comité Especial UJC Provincia: Guantánamo Municipio: Manuel Tames Nombre: CBI Los Industrialistas Sector de la Economía: Industria Rama: Industria Básica Tipo de Centro: Universidad Nombre del Centro:
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Sección Modificar

Primeramente se realiza una búsqueda (no pueden ser modificados los Comités Provinciales, Municipales, Comité Nacional o Comité Especial UJC) y luego se modifican los datos que puedan ser modificados según los requisitos funcionales. Los siguientes casos de prueba están referidos a la búsqueda del comité.

Condición de Entrada	Casos válidos	Casos no válidos
Estructura	Seleccionable.	No seleccionar una estructura.

Caso de uso: Buscar Estructura
Caso de Prueba: Realizar búsqueda de la estructura seleccionando una de las opciones.
Entrada: Estructura: Comité de Base Independiente
Resultado: El sistema muestra un listado de todas las estructuras que coincidan con el parámetro entrado.
Condiciones: Los datos para buscar una estructura deben estar en el rango de valores válidos.

Caso de uso: Buscar Estructura
Caso de Prueba: Realizar búsqueda de la estructura dejando el campo Estructura vacío.
Entrada: Estructura:
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe seleccionar una estructura para realizar la búsqueda.

Caso de Prueba CU: Gestionar Centro

Sección Crear

Condición de Entrada	Casos válidos	Casos no válidos
Nombre del Centro	Letras y números.	No introducir el nombre del centro o introducir caracteres extraños.
Tipo de Centro	Seleccionable.	No seleccionar un tipo de centro.

Caso de uso: Gestionar Centro
Caso de Prueba: Realizar gestiones de inserción de centro introduciendo correctamente todos los datos.
Entrada: Nombre del Centro: Universidad de Oriente Tipo de Centro: Universidad
Resultado: El sistema inserta una nueva estructura en la base de datos.
Condiciones: Los datos para insertar una estructura deben estar en el rango de valores válidos.

Caso de uso: Gestionar Centro
Caso de Prueba: Realizar gestiones de inserción de centro dejando el campo Nombre del Centro vacío.
Entrada: Nombre del Centro: Tipo de Centro: Universidad

Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Gestionar Centro
Caso de Prueba: Realizar gestiones de inserción de centro introduciendo incorrectamente el nombre del centro.
Entrada: Nombre del Centro: Universida^ de Oriente Tipo de Centro: Universidad
Resultado: El sistema muestra un mensaje de error.
Condiciones: El nombre no puede contener caracteres extraños.

Caso de uso: Gestionar Centro
Caso de Prueba: Realizar gestiones de inserción de centro dejando el campo Tipo de Centro vacío.
Entrada: Nombre del Centro: Universidad de Oriente Tipo de Centro:
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Sección Modificar

Primeramente se realiza una búsqueda del centro deseado y luego se modifican sus datos. Los siguientes casos de prueba están referidos a la búsqueda del centro.

Condición de Entrada	Casos válidos	Casos no válidos
Tipo de Centro	Seleccionable.	No seleccionar un tipo de centro.

Caso de uso: Buscar Centro
Caso de Prueba: Realizar búsqueda del centro seleccionando una de las opciones.
Entrada: Tipo de Centro: Universidad
Resultado: El sistema muestra un listado de todos los centros que coincidan con el parámetro entrado.
Condiciones: Los datos para buscar un centro deben estar en el rango de valores válidos.

Caso de uso: Buscar Centro
Caso de Prueba: Realizar búsqueda del centro dejando el campo Tipo de Centro vacío.
Entrada: Tipo de Centro:
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe escoger un tipo de centro para realizar la búsqueda.

Caso de Prueba CU: Gestionar Crecimiento

Condición de Entrada	Casos válidos	Casos no válidos
Paso1Crecimiento:		
Vía de Ingreso	Seleccionable.	No seleccionar una vía de ingreso.
Paso2Crecimiento:		
Curso escuelas políticas	Seleccionable.	No marcar el campo Curso escuelas políticas.
Paso3Crecimiento:		
Ingreso a la organización	Seleccionable.	No marcar el campo Ingreso a la organización.
Organización que realizó el proceso	Letras, números y caracteres extraños.	No introducir la organización que realizó el proceso.
Provincia donde se realizó el proceso	Seleccionable.	No seleccionar una provincia.
Municipio donde se realizó el proceso	Seleccionable.	No seleccionar municipio.

Caso de uso: Gestionar Crecimiento
Caso de Prueba: Realizar gestiones de crecimiento introduciendo correctamente todos los datos.
Entrada: Vía de Ingreso: Consulta con las masas Curso escuelas políticas: Seleccionado.

Ingreso a la organización: Ingresa
Organización que realizó el proceso: Universidad de las Ciencias Informáticas
Provincia donde se realizó el proceso: La habana
Municipio donde se realizó el proceso: La lisa
Resultado: El sistema inserta un nuevo crecimiento en la base de datos.
Condiciones: Los datos para insertar un crecimiento deben estar en el rango de valores válidos.

Caso de uso: Gestionar Crecimiento
Caso de Prueba: Realizar gestiones de crecimiento introduciendo dejando el campo Vía de Ingreso vacío.
Entrada: Vía de Ingreso: Curso escuelas políticas: Seleccionado. Ingreso a la organización: Ingresa Organización que realizó el proceso: Universidad de las Ciencias Informáticas Provincia donde se realizó el proceso: La habana Municipio donde se realizó el proceso: La lisa
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Gestionar Crecimiento
Caso de Prueba: Realizar gestiones de crecimiento dejando el campo Curso escuelas políticas vacío.
Entrada: Vía de Ingreso: Consulta con las masas Curso escuelas políticas: Ingreso a la organización: Ingresa Organización que realizó el proceso: Universidad de las Ciencias Informáticas Provincia donde se realizó el proceso: La habana Municipio donde se realizó el proceso: La lisa
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Gestionar Crecimiento
Caso de Prueba: Realizar gestiones de crecimiento dejando el campo Ingreso a la organización vacío.
Entrada: Vía de Ingreso: Consulta con las masas Curso escuelas políticas: Seleccionado. Ingreso a la organización: Organización que realizó el proceso: Universidad de las Ciencias Informáticas Provincia donde se realizó el proceso: La habana

Municipio donde se realizó el proceso: La lisa
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Gestionar Crecimiento
Caso de Prueba: Realizar gestiones de crecimiento dejando el campo Organización que realizó el proceso vacío.
Entrada: Vía de Ingreso: Consulta con las masas Curso escuelas políticas: Seleccionado. Ingreso a la organización: Ingresa Organización que realizó el proceso: Provincia donde se realizó el proceso: La habana Municipio donde se realizó el proceso: La lisa
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Gestionar Crecimiento
Caso de Prueba: Realizar gestiones de crecimiento dejando el campo Provincia donde se realizó el proceso vacío.
Entrada: Vía de Ingreso: Consulta con las masas Curso escuelas políticas: Seleccionado. Ingreso a la organización: Ingresa Organización que realizó el proceso: Universidad de las Ciencias Informáticas Provincia donde se realizó el proceso: Municipio donde se realizó el proceso: La lisa
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Gestionar Crecimiento
Caso de Prueba: Realizar gestiones de crecimiento dejando el campo Municipio donde se realizó el proceso vacío.
Entrada: Vía de Ingreso: Consulta con las masas Curso escuelas políticas: Seleccionado. Ingreso a la organización: Ingresa Organización que realizó el proceso: Universidad de las Ciencias Informáticas

Provincia donde se realizó el proceso: La habana
Municipio donde se realizó el proceso:
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de Prueba CU: Realizar Alta por Crecimiento

Para realizar un alta se debe seleccionar inicialmente el tipo.

Condición de Entrada	Casos válidos	Casos no válidos
Tipos de alta	Seleccionable.	No seleccionar un tipo de alta.

Caso de Prueba: Iniciar algún tipo de alta seleccionando un tipo.
Entrada: Tipos de alta: Alta por Crecimiento
Resultado: El sistema muestra la interfaz correspondiente.
Condiciones: Se debe haber seleccionado un tipo de alta.
Caso de Prueba: Iniciar algún tipo de alta dejando el campo Tipos de alta vacío.
Entrada: Tipos de alta:
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe seleccionar una opción.

Para realizar un alta por crecimiento primeramente se buscan los miembros del universo juvenil que tengan el crecimiento completo y luego se procede a crear el expediente de militante del universo juvenil seleccionado. El caso de prueba para este caso de uso es análogo al caso de prueba del caso de uso Gestionar Datos UJ sección Modificar o Eliminar.

Caso de Prueba CU: Realizar Alta por Ajuste

Condición de Entrada	Casos válidos	Casos no válidos
Tipos de Ajustes	Seleccionable.	No seleccionar un tipo de ajuste.

Caso de uso: Realizar Alta por Ajuste
Caso de Prueba: Realizar alta por ajuste seleccionando un tipo de ajuste.
Entrada: Tipos de Ajustes: Error de cómputo
Resultado: El sistema muestra la interfaz correspondiente.
Condiciones: Se debe haber seleccionado un tipo de ajuste.

Caso de uso: Realizar Alta por Ajuste.
Caso de Prueba: Realizar alta por ajuste dejando el campo Tipos de Ajustes vacío.
Entrada: Tipos de Ajuste:
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe seleccionar una opción.

Existen tres tipos de alta por ajuste (Error de cómputo, Doble militancia y Regreso al antiguo CB). Para realizar un alta por ajuste de tipo error de cómputo o doble militancia se crea un expediente (ver casos de prueba para el caso de uso Crear Expediente), si el alta por ajuste es de tipo regreso al antiguo CB primeramente se busca el militante y luego se procede a reincorporarlo (ver caso de prueba del caso de uso Realizar Traslado Interno). Los siguientes casos de prueba están referidos a la búsqueda del militante.

Condición de Entrada	Casos válidos	Casos no válidos
Nombre	Letras y números.	No introducir el nombre o introducir caracteres extraños.

Caso de uso: Buscar Militante
Caso de Prueba: Realizar búsqueda del militante entrando correctamente todos los datos.
Entrada: Nombre: Leidys
Resultado: El sistema muestra un listado de todos los militantes que coincidan con el parámetro entrado.
Condiciones: Los datos para buscar un militante deben estar en el rango de valores válidos.

Caso de uso: Buscar Militante
Caso de Prueba: Realizar búsqueda del militante dejando el campo Nombre vacío.
Entrada: Nombre:
Resultado: El sistema muestra un mensaje de error.

Condiciones: Debe introducir un nombre para realizar la búsqueda.

Caso de uso: Buscar Militante.
Caso de Prueba: Realizar búsqueda del militante introduciendo incorrectamente el nombre.
Entrada: Nombre: *Leidita*
Resultado: El sistema muestra un mensaje de error.
Condiciones: El nombre no puede contener caracteres extraños.

Caso de Prueba CU: Realizar Alta por Traslado Externo

Condición de Entrada	Casos válidos	Casos no válidos
Tipos de Traslados Externos	Seleccionable.	No seleccionar un tipo de traslado externo.

Caso de uso: Realizar Alta por Traslado Externo
Caso de Prueba: Realizar alta por traslado externo seleccionando un tipo de traslado externo.
Entrada: Tipos de Traslados Externos: Provincia
Resultado: El sistema muestra la interfaz correspondiente.
Condiciones: Se debe haber seleccionado un tipo de traslado externo.

Caso de uso: Realizar Alta por Traslado Externo
Caso de Prueba: Realizar alta por traslado externo dejando el campo Tipos de Traslados Externos vacío.
Entrada: Tipos de Traslados Externos:
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe seleccionar una opción.

Existen cinco tipos de traslados externos a continuación se exponen los casos de pruebas para cada uno de ellos.

Para el traslado externo desde el MININT, desde las FAR y desde el Exterior sólo se necesita saber la fecha de incorporación y el sistema pone la fecha por defecto por lo que a este campo no se le hace pruebas.

Caso de Prueba: Realizar Alta por Traslado Externo (Municipio)

Condición de Entrada	Casos válidos	Casos no válidos
Municipio Destino	Seleccionable.	No seleccionar un municipio.

Caso de uso: Realizar Alta por Traslado Externo
Caso de Prueba: Realizar alta por traslado externo entre municipios seleccionando un municipio destino.
Entrada: Municipio: Manuel Tames

Resultado: El sistema muestra la interfaz correspondiente para crear un expediente.
Condiciones: Se debe haber seleccionado un municipio.

Caso de uso: Realizar Alta por Traslado Externo
Caso de Prueba: Realizar alta por traslado externo entre municipios dejando el campo Municipio destino vacío.
Entrada: Municipio destino:
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de Prueba: Realizar Alta por Traslado Externo (Provincia)

Condición de Entrada	Casos válidos	Casos no válidos
Municipio (origen)	Seleccionable.	No seleccionar un municipio.
Provincia Destino	Seleccionable.	No seleccionar una provincia.
Municipio (destino)	Seleccionable.	No seleccionar un municipio.

Caso de uso: Realizar Alta por Traslado Externo
Caso de Prueba: Realizar alta por traslado externo entre provincias seleccionando todos los datos.
Entrada: Municipio (origen): Manuel Tames Provincia Destino: Santiago de Cuba Municipio (destino): Contramaestre
Resultado: El sistema muestra la interfaz correspondiente para crear un expediente.
Condiciones: Se deben haber seleccionado todos los campos.

Caso de uso: Realizar Alta por Traslado Externo
Caso de Prueba: Realizar alta por traslado externo entre provincias dejando el campo Municipio (origen) vacío.
Entrada: Municipio (origen): Provincia Destino: Santiago de Cuba Municipio (destino): Contramaestre
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Realizar Alta por Traslado Externo
Caso de Prueba: Realizar alta por traslado externo entre provincias dejando el campo Provincia Destino vacío.
Entrada: Municipio (origen): Manuel Tames Provincia Destino: Municipio (destino): Contramaestre
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Realizar Alta por Traslado Externo
Caso de Prueba: Realizar alta por traslado externo entre provincias dejando el campo Municipio (destino) vacío.
Entrada: Municipio (origen): Manuel Tames Provincia Destino: Santiago de Cuba Municipio (destino):
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de Prueba CU: Realizar Traslado Interno

Para realizar un traslado interno primeramente se busca el militante (los casos de prueba referente a la búsqueda de un militante fueron expuestos en el caso de prueba del caso de uso Realizar Alta por Ajuste), y luego se incorpora el militante en otro CB.

Condición de Entrada	Casos válidos	Casos no válidos
Ubicación (árbol jerárquico)	Seleccionable.	No seleccionar un comité.

Caso de uso: Realizar Traslado Interno
Caso de Prueba: Realizar traslado interno seleccionando un comité.
Entrada: Ubicación: CBI Mantua
Resultado: El sistema da baja al militante de su antiguo CB y lo incorpora al nuevo.
Condiciones: Se debe haber seleccionado un comité.

Caso de uso: Realizar Traslado Interno
Caso de Prueba: Realizar traslado interno sin seleccionar un comité.
Entrada: Ubicación:
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe seleccionar un comité.

Caso de Prueba CU: Realizar Baja Natural

Para realizar una baja se debe seleccionar inicialmente el tipo.

Condición de Entrada	Casos válidos	Casos no válidos
Tipos de baja	Seleccionable.	No seleccionar un tipo de baja.

Caso de Prueba: Iniciar algún tipo de baja seleccionando un tipo.
Entrada: Tipos de baja: Baja Natural
Resultado: El sistema muestra la interfaz correspondiente.
Condiciones: Se debe haber seleccionado un tipo de baja.

Caso de Prueba: Iniciar algún tipo de baja dejando el campo Tipos de baja vacío.
Entrada: Tipos de baja:
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe seleccionar una opción.

Para realizar cualquier tipo de baja inicialmente se busca el militante (los casos de prueba referente a la búsqueda de un militante fueron expuestos en el caso de prueba del caso de uso Realizar Alta por Ajuste) y luego se le da baja en el sistema.

Condición de Entrada	Casos válidos	Casos no válidos
Motivo	Seleccionable.	No seleccionar un motivo.

Caso de uso: Realizar Baja Natural
Caso de Prueba: Realizar baja natural seleccionando un motivo.
Entrada: Motivo: Edad límite
Resultado: El sistema da baja al militante.
Condiciones: Se debe haber seleccionado un motivo.

Caso de uso: Realizar Baja Natural
Caso de Prueba: Realizar baja natural dejando el campo Motivo vacío.
Entrada: Motivo:
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe escoger un motivo.

Caso de Prueba CU: Realizar Baja por Ajuste

Para realizar la baja por ajuste, después de seleccionado el militante se debe escoger el tipo de ajuste, los siguientes casos de pruebas están referidos a la selección del tipo de ajuste.

Condición de Entrada	Casos válidos	Casos no válidos
Motivo de la baja	Seleccionable.	No seleccionar el motivo.

Caso de uso: Realizar Baja por ajuste
Caso de Prueba: Realizar baja por ajuste seleccionando el motivo.
Entrada: Motivo de la baja: Doble Militancia
Resultado: El sistema da baja al militante.
Condiciones: Se debe haber seleccionado el motivo.

Caso de uso: Realizar Baja por ajuste
Caso de Prueba: Realizar baja por ajuste dejando el campo Motivo de la baja vacío.
Entrada: Motivo de la baja:
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe seleccionar el motivo.

Caso de Prueba CU: Realizar Baja por Traslado Externo

Condición de Entrada	Casos válidos	Casos no válidos
Tipos de Traslados Externos	Seleccionable.	No seleccionar un tipo de traslado externo.

Caso de uso: Realizar Baja por Traslado Externo
Caso de Prueba: Realizar baja por traslado externo seleccionando un tipo de traslado externo.
Entrada: Tipos de Traslados Externos: Provincia
Resultado: El sistema muestra la interfaz correspondiente.
Condiciones: Se debe haber seleccionado un tipo de traslado externo.

Caso de uso: Realizar Baja por Traslado Externo
Caso de Prueba: Realizar baja por traslado externo dejando el campo Tipos de Traslados Externos vacío.
Entrada: Tipos de Traslados Externos:
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe seleccionar una opción.

Existen cinco tipos de traslados externos a continuación se exponen los casos de pruebas para cada uno de ellos.

Para el traslado externo desde el MININT, desde las FAR, desde una Provincia y desde el Exterior sólo se necesita saber la fecha de incorporación y el sistema pone la fecha por defecto, por lo que a este campo no se le hace pruebas.

Caso de Prueba: Realizar Alta por Traslado Externo (Municipio)

Condición de Entrada	Casos válidos	Casos no válidos
Municipio (origen)	Seleccionable.	No seleccionar un municipio.

Caso de uso: Realizar Baja por Traslado Externo
Caso de Prueba: Realizar baja por traslado externo entre municipios seleccionando un municipio origen.
Entrada: Municipio (origen): Manuel Tames
Resultado: El sistema da baja al militante.
Condiciones: Se debe haber seleccionado un municipio.

Caso de uso: Realizar Baja por Traslado Externo
Caso de Prueba: Realizar baja por traslado externo entre municipios dejando el campo Municipio (origen) vacío.
Entrada: Municipio (Origen):
Resultado: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de Prueba: Crear Expediente

Condición de Entrada	Casos válidos	Casos no válidos
Datos Personales		
Nombre	Letras y números.	No introducir el nombre o introducir caracteres extraños.
Segundo nombre	Letras y números.	Introducir caracteres extraños.
Apellido	Letras y números.	No introducir el primer apellido o introducir caracteres extraños.
Segundo apellido	Letras y números.	No introducir el segundo apellido o introducir caracteres extraños.
Carné de identidad	Números.	No introducir el número de carné de identidad. Introducir letras o caracteres extraños. Introducir un total de dígitos menor o mayor que 11.
Color de la piel	Seleccionable.	No seleccionar un color de la piel.
Sexo	Seleccionable.	No seleccionar un sexo.
Número de carné del militante	Letras, números y caracteres extraños.	No introducir el número de carné del militante.
Dirección Particular	Letras, números y caracteres extraños.	No introducir la dirección particular.
Provincia	Seleccionable.	No seleccionar una provincia.
Municipio	Seleccionable.	No seleccionar un municipio.

Nivel cultural vencido	Seleccionable.	No seleccionar un nivel cultural.
Teléfono	Números.	Introducir letras o caracteres extraños.
Lugar de nacimiento	Letras, números y caracteres extraños.	
Asignar Comité		
Ubicación	Seleccionable.	No seleccionar un CB.
Datos Familiares		
Nombre	Letras y números.	Introducir caracteres extraños.
Segundo nombre	Letras y números.	Introducir caracteres extraños.
Apellido	Letras y números.	Introducir caracteres extraños.
Segundo apellido	Letras y números.	Introducir caracteres extraños.
Datos de Interés		
Duración	Letras y números.	Introducir caracteres extraños.
País	Letras.	Introducir números o caracteres extraños.
Motivo	Letras, números y caracteres extraños.	
Año (cargo)	Números.	Introducir letras o caracteres extraños.
Datos Laborales		
Clasificador General	Seleccionable.	No seleccionar un clasificador general.
Clasificador Ocupacional	Seleccionable.	No seleccionar un clasificador ocupacional.

Labor	Letras.	Introducir números o caracteres extraños.
Proceso de Ingreso		
Vía de ingreso utilizada	Seleccionable.	No seleccionar una vía de ingreso.
Organización que desarrolló el proceso	Letras, números y caracteres extraños.	No introducir la organización que desarrolló el proceso.

Caso de uso: Crear Expediente
Caso de prueba: Crear un expediente entrando correctamente todos los datos.
<p>Entrada:</p> <p>Datos Personales</p> <p>Nombre: Marlen</p> <p>Segundo nombre:</p> <p>Apellido: Fajardo</p> <p>Segundo apellido: Díaz</p> <p>Carné de identidad: 86100212258</p> <p>Color de la piel: Blanca</p> <p>Sexo: Femenino</p> <p>Número carné militante: 865M</p> <p>Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín.</p> <p>Provincia: Holguín.</p> <p>Municipio: Holguín.</p> <p>Nivel cultural vencido: Secundaria Básica.</p> <p>Asignar Comité</p> <p>CBI Mantua.</p> <p>Datos Familiares</p> <p>Primer nombre: Justo</p> <p>Segundo nombre: Enrique</p> <p>Primer apellido: Fajardo</p> <p>Segundo apellido: Cabrera</p>

<p>Datos Laborales</p> <p>Clasificador General: MILITANTES OBREROS.</p> <p>Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas.</p> <p>Labor: Limpia la UBPC.</p> <p>Proceso de Ingreso</p> <p>Vía de ingreso utilizada: Asamblea de ejemplares.</p> <p>Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.</p>
<p>Resultados: El sistema crea un expediente.</p>
<p>Condiciones: Los datos para crear un expediente deben de estar en el rango de valores válidos.</p>

<p>Caso de uso: Crear Expediente</p>
<p>Caso de prueba: Crear un expediente entrando incorrectamente el nombre del militante (Datos Personales).</p>
<p>Entrada:</p> <p>Datos Personales</p> <p>Nombre: M@rlen</p> <p>Segundo nombre:</p> <p>Apellido: Fajardo</p> <p>Segundo apellido: Díaz</p> <p>Carné de identidad: 86100212258</p> <p>Color de la piel: Blanca</p> <p>Sexo: Femenino</p> <p>Número carné militante: 865M</p> <p>Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín.</p> <p>Provincia: Holguín.</p> <p>Municipio: Holguín.</p> <p>Nivel cultural vencido: Secundaria Básica.</p> <p>Asignar Comité</p> <p>CBI Mantua.</p> <p>Datos Familiares</p> <p>Primer nombre: Justo</p> <p>Segundo nombre: Enrique</p>

Primer apellido: Fajardo
Segundo apellido: Cabrera
Datos Laborales
Clasificador General: MILITANTES OBREROS.
Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas.
Labor: Limpia la UBPC.
Proceso de Ingreso
Vía de ingreso utilizada: Asamblea de ejemplares.
Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.
Resultados: El sistema muestra un mensaje de error.
Condiciones: El nombre no puede contener caracteres extraños.

Caso de uso: Crear Expediente
Caso de prueba: Crear un expediente dejando el campo Nombre (Datos Personales) vacío.
Entrada:
Datos Personales
Nombre:
Segundo nombre:
Apellido: Fajardo
Segundo apellido: Díaz
Carné de identidad: 86100212258
Color de la piel: Blanca
Sexo: Femenino
Número carné militante: 865M
Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín.
Provincia: Holguín.
Municipio: Holguín.
Nivel cultural vencido: Secundaria Básica.
Asignar Comité
CBI Mantua.
Datos Familiares
Primer nombre: Justo

Segundo nombre: Enrique Primer apellido: Fajardo Segundo apellido: Cabrera Datos Laborales Clasificador General: MILITANTES OBREROS. Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas. Labor: Limpia la UBPC. Proceso de Ingreso Vía de ingreso utilizada: Asamblea de ejemplares. Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.
Resultados: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Crear Expediente
Caso de prueba: Crear un expediente introduciendo incorrectamente el segundo nombre (Datos Personales).
Entrada: Datos Personales Nombre: Marlen Segundo nombre: Mar&a Apellido: Fajardo Segundo apellido: Díaz Carné de identidad: 86100212258 Color de la piel: Blanca Sexo: Femenino Número carné militante: 865M Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín. Provincia: Holguín. Municipio: Holguín. Nivel cultural vencido: Secundaria Básica. Asignar Comité CBI Mantua.

<p>Datos Familiares</p> <p>Primer nombre: Justo Segundo nombre: Enrique Primer apellido: Fajardo Segundo apellido: Cabrera</p> <p>Datos Laborales</p> <p>Clasificador General: MILITANTES OBREROS. Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas. Labor: Limpia la UBPC.</p> <p>Proceso de Ingreso</p> <p>Vía de ingreso utilizada: Asamblea de ejemplares. Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.</p> <p>Resultados: El sistema muestra un mensaje de error.</p> <p>Condiciones: El nombre no puede contener caracteres extraños.</p>

<p>Caso de uso: Crear Expediente</p>
<p>Caso de prueba: Crear un expediente dejando el campo Apellido (Datos Personales) vacío.</p>
<p>Entrada:</p> <p>Datos Personales</p> <p>Nombre: Marlen Segundo nombre: Apellido: Segundo apellido: Díaz Carné de identidad: 86100212258 Color de la piel: Blanca Sexo: Femenino Número carné militante: 865M Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín. Provincia: Holguín. Municipio: Holguín. Nivel cultural vencido: Secundaria Básica.</p> <p>Asignar Comité</p>

<p>CBI Mantua.</p> <p>Datos Familiares</p> <p>Primer nombre: Justo</p> <p>Segundo nombre: Enrique</p> <p>Primer apellido: Fajardo</p> <p>Segundo apellido: Cabrera</p> <p>Datos Laborales</p> <p>Clasificador General: MILITANTES OBREROS.</p> <p>Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas.</p> <p>Labor: Limpia la UBPC.</p> <p>Proceso de Ingreso</p> <p>Vía de ingreso utilizada: Asamblea de ejemplares.</p> <p>Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.</p>
<p>Resultados: El sistema muestra un mensaje de error.</p>
<p>Condiciones: Debe llenar todos los campos obligatorios.</p>

<p>Caso de uso: Crear Expediente</p>
<p>Caso de prueba: Crear un expediente introduciendo incorrectamente el primer apellido (Datos Personales).</p>
<p>Entrada:</p> <p>Datos Personales</p> <p>Nombre: Marlen</p> <p>Segundo nombre:</p> <p>Apellido: Faj<>rdo</p> <p>Segundo apellido: Díaz</p> <p>Carné de identidad: 86100212258</p> <p>Color de la piel: Blanca</p> <p>Sexo: Femenino</p> <p>Número carné militante: 865M</p> <p>Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín.</p> <p>Provincia: Holguín.</p> <p>Municipio: Holguín.</p>

Nivel cultural vencido: Secundaria Básica.
Asignar Comité
CBI Mantua.
Datos Familiares
Primer nombre: Justo
Segundo nombre: Enrique
Primer apellido: Fajardo
Segundo apellido: Cabrera
Datos Laborales
Clasificador General: MILITANTES OBREROS.
Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas.
Labor: Limpia la UBPC.
Proceso de Ingreso
Vía de ingreso utilizada: Asamblea de ejemplares.
Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.
Resultados: El sistema muestra un mensaje de error.
Condiciones: Los apellidos no pueden contener caracteres extraños.

Caso de uso: Crear Expediente
Caso de prueba: Crear un expediente dejando el campo Segundo apellido (Datos Personales) vacío.
Entrada:
Datos Personales
Nombre: Marlen
Segundo nombre:
Apellido: Fajardo
Segundo apellido:
Carné de identidad: 86100212258
Color de la piel: Blanca
Sexo: Femenino
Número carné militante: 865M
Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín.

<p>Provincia: Holguín. Municipio: Holguín. Nivel cultural vencido: Secundaria Básica.</p> <p>Asignar Comité CBI Mantua.</p> <p>Datos Familiares Primer nombre: Justo Segundo nombre: Enrique Primer apellido: Fajardo Segundo apellido: Cabrera</p> <p>Datos Laborales Clasificador General: MILITANTES OBREROS. Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas. Labor: Limpia la UBPC.</p> <p>Proceso de Ingreso Vía de ingreso utilizada: Asamblea de ejemplares. Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.</p>
<p>Resultados: El sistema muestra un mensaje de error.</p>
<p>Condiciones: Debe llenar todos los campos obligatorios.</p>

<p>Caso de uso: Crear Expediente</p>
<p>Caso de prueba: Crear un expediente introduciendo incorrectamente el segundo apellido (Datos Personales).</p>
<p>Entrada:</p> <p>Datos Personales Nombre: Marlen Segundo nombre: Apellido: Fajardo Segundo apellido: D*az Carné de identidad: 86100212258 Color de la piel: Blanca Sexo: Femenino</p>

Número carné militante: 865M
Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín.
Provincia: Holguín.
Municipio: Holguín.
Nivel cultural vencido: Secundaria Básica.
Asignar Comité
CBI Mantua.
Datos Familiares
Primer nombre: Justo
Segundo nombre: Enrique
Primer apellido: Fajardo
Segundo apellido: Cabrera
Datos Laborales
Clasificador General: MILITANTES OBREROS.
Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas.
Labor: Limpia la UBPC.
Proceso de Ingreso
Vía de ingreso utilizada: Asamblea de ejemplares.
Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.
Resultados: El sistema muestra un mensaje de error.
Condiciones: Los apellidos no pueden contener caracteres extraños.

Caso de uso: Crear Expediente
Caso de prueba: Crear un expediente dejando el campo Carné de identidad (Datos Personales) vacío.
Entrada:
Datos Personales
Nombre: Marlen
Segundo nombre:
Apellido: Fajardo
Segundo apellido: Díaz
Carné de identidad:
Color de la piel: Blanca

<p>Sexo: Femenino</p> <p>Número carné militante: 865M</p> <p>Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín.</p> <p>Provincia: Holguín.</p> <p>Municipio: Holguín.</p> <p>Nivel cultural vencido: Secundaria Básica.</p> <p>Asignar Comité</p> <p>CBI Mantua.</p> <p>Datos Familiares</p> <p>Primer nombre: Justo</p> <p>Segundo nombre: Enrique</p> <p>Primer apellido: Fajardo</p> <p>Segundo apellido: Cabrera</p> <p>Datos Laborales</p> <p>Clasificador General: MILITANTES OBREROS.</p> <p>Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas.</p> <p>Labor: Limpia la UBPC.</p> <p>Proceso de Ingreso</p> <p>Vía de ingreso utilizada: Asamblea de ejemplares.</p> <p>Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.</p>
<p>Resultados: El sistema muestra un mensaje de error.</p>
<p>Condiciones: Debe llenar todos los campos obligatorios.</p>

<p>Caso de uso: Crear Expediente</p>
<p>Caso de prueba: Crear un expediente introduciendo letras en el carné de identidad (Datos Personales).</p>
<p>Entrada:</p> <p>Datos Personales</p> <p>Nombre: Marlen</p> <p>Segundo nombre:</p> <p>Apellido: Fajardo</p> <p>Segundo apellido: Díaz</p> <p>Carné de identidad: 8605o28DG19</p>

Color de la piel: Blanca
Sexo: Femenino
Número carné militante: 865M
Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín.
Provincia: Holguín.
Municipio: Holguín.
Nivel cultural vencido: Secundaria Básica.
Asignar Comité
CBI Mantua.
Datos Familiares
Primer nombre: Justo
Segundo nombre: Enrique
Primer apellido: Fajardo
Segundo apellido: Cabrera
Datos Laborales
Clasificador General: MILITANTES OBREROS.
Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas.
Labor: Limpia la UBPC.
Proceso de Ingreso
Vía de ingreso utilizada: Asamblea de ejemplares.
Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.
Resultados: El sistema muestra un mensaje de error.
Condiciones: El carné de identidad debe contener sólo números.

Caso de uso: Crear Expediente
Caso de prueba: Crear un expediente introduciendo caracteres extraños en el carné de identidad (Datos Personales).
Entrada:
Datos Personales
Nombre: Marlen
Segundo nombre:
Apellido: Fajardo

Segundo apellido: Díaz
Carné de identidad: 86056285^19
Color de la piel: Blanca
Sexo: Femenino
Número carné militante: 865M
Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín.
Provincia: Holguín.
Municipio: Holguín.
Nivel cultural vencido: Secundaria Básica.
Asignar Comité
CBI Mantua.
Datos Familiares
Primer nombre: Justo
Segundo nombre: Enrique
Primer apellido: Fajardo
Segundo apellido: Cabrera
Datos Laborales
Clasificador General: MILITANTES OBREROS.
Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas.
Labor: Limpia la UBPC.
Proceso de Ingreso
Vía de ingreso utilizada: Asamblea de ejemplares.
Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.
Resultados: El sistema muestra un mensaje de error.
Condiciones: El carné de identidad debe contener sólo números.

Caso de uso: Crear Expediente
Caso de prueba: Crear un expediente introduciendo un numero de dígitos distinto de 11 en el carné de identidad (Datos Personales).
Entrada: Datos Personales Nombre: Marlen Segundo nombre: Apellido: Fajardo Segundo apellido: Díaz Carné de identidad: 86054567892345 Color de la piel: Blanca Sexo: Femenino Número carné militante: 865M Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín. Provincia: Holguín. Municipio: Holguín. Nivel cultural vencido: Secundaria Básica. Asignar Comité CBI Mantua. Datos Familiares Primer nombre: Justo Segundo nombre: Enrique Primer apellido: Fajardo Segundo apellido: Cabrera Datos Laborales Clasificador General: MILITANTES OBREROS. Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas. Labor: Limpia la UBPC. Proceso de Ingreso Vía de ingreso utilizada: Asamblea de ejemplares. Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.
Resultados: El sistema muestra un mensaje de error.
Condiciones: El carné de identidad debe contener 11 dígitos.

Caso de uso: Crear Expediente
Caso de prueba: Crear un expediente dejando el campo Color de la piel (Datos Personales) vacío.
Entrada: Datos Personales Nombre: Marlen Segundo nombre: Apellido: Fajardo Segundo apellido: Díaz Carné de identidad: 86081528819 Color de la piel: Sexo: Femenino Número carné militante: 865M Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín. Provincia: Holguín. Municipio: Holguín. Nivel cultural vencido: Secundaria Básica. Asignar Comité CBI Mantua. Datos Familiares Primer nombre: Justo Segundo nombre: Enrique Primer apellido: Fajardo Segundo apellido: Cabrera Datos Laborales Clasificador General: MILITANTES OBREROS. Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas. Labor: Limpia la UBPC. Proceso de Ingreso Vía de ingreso utilizada: Asamblea de ejemplares. Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.
Resultados: El sistema muestra un mensaje de error.

Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Crear Expediente

Caso de prueba: Crear un expediente dejando el campo Sexo (Datos Personales) vacío.

Entrada:

Datos Personales

Nombre: Marlen

Segundo nombre:

Apellido: Fajardo

Segundo apellido: Díaz

Carné de identidad: 86081528819

Color de la piel: Blanca

Sexo:

Número carné militante: 865M

Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín.

Provincia: Holguín.

Municipio: Holguín.

Nivel cultural vencido: Secundaria Básica.

Asignar Comité

CBI Mantua.

Datos Familiares

Primer nombre: Justo

Segundo nombre: Enrique

Primer apellido: Fajardo

Segundo apellido: Cabrera

Datos Laborales

Clasificador General: MILITANTES OBREROS.

Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas.

Labor: Limpia la UBPC.

Proceso de Ingreso

Vía de ingreso utilizada: Asamblea de ejemplares.

Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.

Resultados: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Crear Expediente
Caso de prueba: Crear un expediente dejando el campo Número de carné del militante (Datos Personales) vacío.
Entrada: Datos Personales Nombre: Marlen Segundo nombre: Apellido: Fajardo Segundo apellido: Díaz Carné de identidad: 86081528819 Color de la piel: Blanca Sexo: Femenino Número carné militante: Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín. Provincia: Holguín. Municipio: Holguín. Nivel cultural vencido: Secundaria Básica. Asignar Comité CBI Mantua. Datos Familiares Primer nombre: Justo Segundo nombre: Enrique Primer apellido: Fajardo Segundo apellido: Cabrera Datos Laborales Clasificador General: MILITANTES OBREROS. Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas. Labor: Limpia la UBPC. Proceso de Ingreso

Vía de ingreso utilizada: Asamblea de ejemplares.
Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.
Resultados: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Crear Expediente
Caso de prueba: Crear un expediente dejando el campo Dirección particular (Datos Personales) vacío.
Entrada: Datos Personales Nombre: Marlen Segundo nombre: Apellido: Fajardo Segundo apellido: Díaz Carné de identidad: 86081528819 Color de la piel: Blanca Sexo: Femenino Número carné militante: 865M Dirección particular: Provincia: Holguín. Municipio: Holguín. Nivel cultural vencido: Secundaria Básica. Asignar Comité CBI Mantua. Datos Familiares Primer nombre: Justo Segundo nombre: Enrique Primer apellido: Fajardo Segundo apellido: Cabrera Datos Laborales Clasificador General: MILITANTES OBREROS. Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas. Labor: Limpia la UBPC.

<p>Proceso de Ingreso</p> <p>Vía de ingreso utilizada: Asamblea de ejemplares.</p> <p>Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.</p>
<p>Resultados: El sistema muestra un mensaje de error.</p>
<p>Condiciones: Debe llenar todos los campos obligatorios.</p>

<p>Caso de uso: Crear Expediente</p>
<p>Caso de prueba: Crear un expediente dejando el campo Provincia (Datos Personales) vacío.</p>
<p>Entrada:</p> <p>Datos Personales</p> <p>Nombre: Marlen</p> <p>Segundo nombre:</p> <p>Apellido: Fajardo</p> <p>Segundo apellido: Díaz</p> <p>Carné de identidad: 86081528819</p> <p>Color de la piel: Blanca</p> <p>Sexo: Femenino</p> <p>Número carné militante: 865M</p> <p>Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín.</p> <p>Provincia:</p> <p>Municipio: Holguín.</p> <p>Nivel cultural vencido: Secundaria Básica.</p> <p>Asignar Comité</p> <p>CBI Mantua.</p> <p>Datos Familiares</p> <p>Primer nombre: Justo</p> <p>Segundo nombre: Enrique</p> <p>Primer apellido: Fajardo</p> <p>Segundo apellido: Cabrera</p> <p>Datos Laborales</p> <p>Clasificador General: MILITANTES OBREROS.</p> <p>Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas.</p>

<p>Labor: Limpia la UBPC.</p> <p>Proceso de Ingreso</p> <p>Vía de ingreso utilizada: Asamblea de ejemplares.</p> <p>Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.</p>
<p>Resultados: El sistema muestra un mensaje de error.</p>
<p>Condiciones: Debe llenar todos los campos obligatorios.</p>

<p>Caso de uso: Crear Expediente</p>
<p>Caso de prueba: Crear un expediente dejando el campo Municipio (Datos Personales) vacío.</p>
<p>Entrada:</p> <p>Datos Personales</p> <p>Nombre: Marlen</p> <p>Segundo nombre:</p> <p>Apellido: Fajardo</p> <p>Segundo apellido: Díaz</p> <p>Carné de identidad: 86081528819</p> <p>Color de la piel: Blanca</p> <p>Sexo: Femenino</p> <p>Número carné militante: 865M</p> <p>Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín.</p> <p>Provincia: Holguín</p> <p>Municipio:</p> <p>Nivel cultural vencido: Secundaria Básica.</p> <p>Asignar Comité</p> <p>CBI Mantua.</p> <p>Datos Familiares</p> <p>Primer nombre: Justo</p> <p>Segundo nombre: Enrique</p> <p>Primer apellido: Fajardo</p> <p>Segundo apellido: Cabrera</p> <p>Datos Laborales</p> <p>Clasificador General: MILITANTES OBREROS.</p>

Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas. Labor: Limpia la UBPC. Proceso de Ingreso Vía de ingreso utilizada: Asamblea de ejemplares. Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.
Resultados: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Crear Expediente
Caso de prueba: Crear un expediente dejando el campo Nivel cultural vencido (Datos Personales) vacío.
Entrada: Datos Personales Nombre: Marlen Segundo nombre: Apellido: Fajardo Segundo apellido: Díaz Carné de identidad: 86081528819 Color de la piel: Blanca Sexo: Femenino Número carné militante: 865M Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín. Provincia: Holguín Municipio: Holguín. Nivel cultural vencido: Asignar Comité CBI Mantua. Datos Familiares Primer nombre: Justo Segundo nombre: Enrique Primer apellido: Fajardo Segundo apellido: Cabrera Datos Laborales

Clasificador General: MILITANTES OBREROS. Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas. Labor: Limpia la UBPC. Proceso de Ingreso Vía de ingreso utilizada: Asamblea de ejemplares. Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.
Resultados: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Crear Expediente
Caso de prueba: Crear un expediente sin seleccionar la ubicación del militante (Asignar Comité) vacío.
Entrada: Datos Personales Nombre: Marlen Segundo nombre: Apellido: Fajardo Segundo apellido: Díaz Carné de identidad: 86081528819 Color de la piel: Blanca Sexo: Femenino Número carné militante: 865M Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín. Provincia: Holguín Municipio: Holguín. Nivel cultural vencido: Secundaria Básica. Asignar Comité Datos Familiares Primer nombre: Justo Segundo nombre: Enrique Primer apellido: Fajardo Segundo apellido: Cabrera

<p>Datos Laborales</p> <p>Clasificador General: MILITANTES OBREROS.</p> <p>Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas.</p> <p>Labor: Limpia la UBPC.</p> <p>Proceso de Ingreso</p> <p>Vía de ingreso utilizada: Asamblea de ejemplares.</p> <p>Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.</p>
<p>Resultados: El sistema muestra un mensaje de error.</p>
<p>Condiciones: Debe llenar todos los campos obligatorios.</p>

<p>Caso de uso: Crear Expediente</p>
<p>Caso de prueba: Crear un expediente dejando el campo Clasificador General (Datos Laborales) vacío.</p>
<p>Entrada:</p> <p>Datos Personales</p> <p>Nombre: Marlen</p> <p>Segundo nombre:</p> <p>Apellido: Fajardo</p> <p>Segundo apellido: Díaz</p> <p>Carné de identidad: 86081528819</p> <p>Color de la piel: Blanca</p> <p>Sexo: Femenino</p> <p>Número carné militante: 865M</p> <p>Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín.</p> <p>Provincia: Holguín</p> <p>Municipio: Holguín.</p> <p>Nivel cultural vencido: Secundaria Básica.</p> <p>Asignar Comité</p> <p>CBI Mantua.</p> <p>Datos Familiares</p> <p>Primer nombre: Justo</p> <p>Segundo nombre: Enrique</p> <p>Primer apellido: Fajardo</p>

<p>Segundo apellido: Cabrera</p> <p>Datos Laborales</p> <p>Clasificador General:</p> <p>Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas. Labor: Limpia la UBPC.</p> <p>Proceso de Ingreso</p> <p>Vía de ingreso utilizada: Asamblea de ejemplares. Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.</p>
<p>Resultados: El sistema muestra un mensaje de error.</p>
<p>Condiciones: Debe llenar todos los campos obligatorios.</p>

<p>Caso de uso: Crear Expediente</p>
<p>Caso de prueba: Crear un expediente dejando el campo Clasificador Ocupacional (Datos Laborales) vacío.</p>
<p>Entrada:</p> <p>Datos Personales</p> <p>Nombre: Marlen Segundo nombre: Apellido: Fajardo Segundo apellido: Díaz Carné de identidad: 86081528819 Color de la piel: Blanca Sexo: Femenino Número carné militante: 865M Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín. Provincia: Holguín Municipio: Holguín. Nivel cultural vencido: Secundaria Básica.</p> <p>Asignar Comité</p> <p>CBI Mantua.</p> <p>Datos Familiares</p> <p>Primer nombre: Justo</p>

Segundo nombre: Enrique
Primer apellido: Fajardo
Segundo apellido: Cabrera
Datos Laborales
Clasificador General: MILITANTES OBREROS.
Clasificador Ocupacional:
Labor: Limpia la UBPC.
Proceso de Ingreso
Vía de ingreso utilizada: Asamblea de ejemplares.
Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.
Resultados: El sistema muestra un mensaje de error.
Condiciones: Debe llenar todos los campos obligatorios.

Caso de uso: Crear Expediente
Caso de prueba: Crear un expediente introduciendo números en la labor (Datos Laborales).
Entrada:
Datos Personales
Nombre: Marlen
Segundo nombre:
Apellido: Fajardo
Segundo apellido: Díaz
Carné de identidad: 86081528819
Color de la piel: Blanca
Sexo: Femenino
Número carné militante: 865M
Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín.
Provincia: Holguín
Municipio: Holguín.
Nivel cultural vencido: Secundaria Básica.
Asignar Comité
CBI Mantua.
Datos Familiares

<p>Primer nombre: Justo</p> <p>Segundo nombre: Enrique</p> <p>Primer apellido: Fajardo</p> <p>Segundo apellido: Cabrera</p> <p>Datos Laborales</p> <p>Clasificador General: MILITANTES OBREROS.</p> <p>Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas.</p> <p>Labor: Limpia la UBPC.</p> <p>Proceso de Ingreso</p> <p>Vía de ingreso utilizada: Asamblea de ejemplares.</p> <p>Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.</p>
<p>Resultados: El sistema muestra un mensaje de error.</p>
<p>Condiciones: La labor solo puede contener letras.</p>

<p>Caso de uso: Crear Expediente</p>
<p>Caso de prueba: Crear un expediente introduciendo caracteres extraños en la labor (Datos Laborales).</p>
<p>Entrada:</p> <p>Datos Personales</p> <p>Nombre: Marlen</p> <p>Segundo nombre:</p> <p>Apellido: Fajardo</p> <p>Segundo apellido: Díaz</p> <p>Carné de identidad: 86081528819</p> <p>Color de la piel: Blanca</p> <p>Sexo: Femenino</p> <p>Número carné militante: 865M</p> <p>Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín.</p> <p>Provincia: Holguín</p> <p>Municipio: Holguín.</p> <p>Nivel cultural vencido: Secundaria Básica.</p> <p>Asignar Comité</p> <p>CBI Mantua.</p>

<p>Datos Familiares</p> <p>Primer nombre: Justo Segundo nombre: Enrique Primer apellido: Fajardo Segundo apellido: Cabrera</p> <p>Datos Laborales</p> <p>Clasificador General: MILITANTES OBREROS. Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas. Labor: Limpia la UBPC.</p> <p>Proceso de Ingreso</p> <p>Vía de ingreso utilizada: Asamblea de ejemplares. Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.</p> <p>Resultados: El sistema muestra un mensaje de error.</p> <p>Condiciones: La labor solo puede contener letras.</p>
--

<p>Caso de uso: Crear Expediente</p>
<p>Caso de prueba: Crear un expediente dejando el campo Vía de ingreso utilizada (Proceso de Ingreso) vacío.</p>
<p>Entrada:</p> <p>Datos Personales</p> <p>Nombre: Marlen Segundo nombre: Apellido: Fajardo Segundo apellido: Díaz Carné de identidad: 86081528819 Color de la piel: Blanca Sexo: Femenino Número carné militante: 865M Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín. Provincia: Holguín Municipio: Holguín. Nivel cultural vencido: Secundaria Básica.</p>

<p>Asignar Comité CBI Mantua.</p> <p>Datos Familiares Primer nombre: Justo Segundo nombre: Enrique Primer apellido: Fajardo Segundo apellido: Cabrera</p> <p>Datos Laborales Clasificador General: MILITANTES OBREROS. Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas. Labor: Limpia la UBPC.</p> <p>Proceso de Ingreso Vía de ingreso utilizada: Organización que desarrolló el proceso: Universidad de las Ciencias Informáticas.</p>
<p>Resultados: El sistema muestra un mensaje de error.</p>
<p>Condiciones: Debe llenar todos los campos obligatorios.</p>

<p>Caso de uso: Crear Expediente</p>
<p>Caso de prueba: Crear un expediente dejando el campo Organización que desarrolló el proceso (Proceso de Ingreso) vacío.</p>
<p>Entrada:</p> <p>Datos Personales Nombre: Marlen Segundo nombre: Apellido: Fajardo Segundo apellido: Díaz Carné de identidad: 86081528819 Color de la piel: Blanca Sexo: Femenino Número carné militante: 865M Dirección particular: Calle 13 % 26 y 30. Nuevo Llano. Holguín. Provincia: Holguín</p>

Municipio: Holguín.

Nivel cultural vencido: Secundaria Básica.

Asignar Comité

CBI Mantua.

Datos Familiares

Primer nombre: Justo

Segundo nombre: Enrique

Primer apellido: Fajardo

Segundo apellido: Cabrera

Datos Laborales

Clasificador General: MILITANTES OBREROS.

Clasificador Ocupacional: Militantes Obreros Agrícolas de las UBPC o Granjas.

Labor: Limpia la UBPC.

Proceso de Ingreso

Vía de ingreso utilizada: Asamblea de ejemplares.

Organización que desarrolló el proceso:

Resultados: El sistema muestra un mensaje de error.

Condiciones: Debe llenar todos los campos obligatorios.

GLOSARIO DE TÉRMINOS

.NET: Es un proyecto de *Microsoft* para crear una nueva plataforma de desarrollo de *software* con énfasis en transparencia de redes, con independencia de plataforma de *hardware* y que permita un rápido desarrollo de aplicaciones.

C/B: Comité de Base.

C++: Es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup.

Case: *Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadoras.

CI: Carné de Identidad.

Classpath: Variable que permite a la máquina virtual de Java, ubicar todas las clases que la constituyen así como los directorios donde se encuentran las librerías externas. Indica dónde debe buscar los archivos a compilar o ejecutar, sin tener que escribir en cada ejecución la ruta completa.

CU: Caso de uso.

CUN: Casos de Uso del Negocio.

CUS: Caso de Uso del Sistema.

DB4O: Novedoso motor de base de datos orientada a objetos. Sus siglas se corresponden con la expresión "*DataBase (for) Objects*", que a su vez es el nombre de la compañía que lo desarrolla. Es una base de datos orientada a objetos completamente nativa de alto rendimiento. Compacta y válida como Base de datos embebida en la aplicación.

DCA: Diagrama de Clases del Análisis.

DCD: Diagrama de clases del diseño.

Despliegue: Ocurre cuando varios trabajos más o menos independientes se distribuyen entre diferentes dispositivos hardware (procesadores).

Diagrama: Representación gráfica para facilitar el entendimiento de un algoritmo o proceso que utiliza una serie de símbolos con significados especiales.

Eclipse: Es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.

Eficacia: Capacidad de lograr un fin deseado o esperado.

Eficiencia: Capacidad de alcanzar los objetivos y metas programadas con el mínimo

FEU: Federación Estudiantil Universitaria.

Gestión de información: Es el proceso que se encarga de suministrar los recursos necesarios para la toma de decisiones, así como para mejorar los procesos, productos y servicios de la organización.

GOF: *Gang of Four* (Banda de los Cuatro) es el nombre con el que se conoce comúnmente a los autores del libro Patrones de Diseño, el cual hace referencia específicamente al diseño orientado a objetos.

GUI: La interfaz gráfica de usuario (en Idioma inglés *Graphical User Interface*, GUI) es un tipo de interfaz de usuario que utiliza un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

Hibernate: Es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos XML que permiten establecer estas relaciones.

IBM: International Business Machines es una empresa que fabrica y comercializa herramientas, programas y servicios relacionados con la informática.

IDEs: Un entorno de desarrollo integrado o, en inglés, *Integrated Development Environment*, es un programa compuesto por un conjunto de herramientas para un programador. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI.

IEEE: Corresponde a las siglas de *The Institute of Electrical and Electronics Engineers*, el Instituto de Ingenieros Eléctricos y Electrónicos, una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas.

Java: Lenguaje de programación orientado a objetos desarrollado por *Sun Microsystems*.

Módulo: Es una parte de un programa informático, el cual realizará una o varias de las tareas de dicho programa para cumplir con su función u objetivos.

NetBeans: Una plataforma para el desarrollo de aplicaciones de escritorio usando *Java* y a un entorno de desarrollo integrado (IDE) desarrollado usando la Plataforma NetBeans.

Optimización: Buscar la mejor manera de realizar una actividad. Consiste en atribuir

PCC: Partido Comunista de Cuba.

POO: Programación Orientada a Objetos, es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora. Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo y encapsulamiento.

RAM: *Random Acces Memory*, Memoria de Acceso Aleatorio, es la memoria principal, donde la computadora guarda los datos que está utilizando en el momento presente.

RUP: *Rational Unified Process* (Proceso Unificado de Rational), es un proceso de desarrollo de software.

Sistema integral: Un sistema formado por varios subsistemas los cuales pueden realizar una actividad determinada. Dichas actividades pueden ser medidas en cuanto a su resultado.

Sun Microsystems: Es una empresa informática fabricante de semiconductores y software. Las siglas SUN se derivan de: *Stanford University Network*, proyecto que se había creado para interconectar en red las bibliotecas de la Universidad de *Stanford*.

UCI: Universidad de las Ciencias Informáticas.

UJC: Unión de Jóvenes Comunista.

Universo juvenil: Todos los jóvenes entre los 16 y 30 años de edad que no forma parte de las filas de UJC.

XML: siglas en inglés de *Extensible Markup Language*, lenguaje de marcas extensible.