



Facultad 2

“Sistema informático para la gestión de la información de los procesos de la UJC en la Facultad 2”

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

Autores: Lianne Guanche Olivera

Sergio Luis Reyna Pérez

Tutor: P. As. Ing. Darling Darías Pérez Ms. C.

Co-tutor: Ing. Claudia Amanda García Gómez

La Habana, 9 de febrero de 2024

Año 65 de la Revolución

DECLARACIÓN DE AUTORÍA

Los autores del trabajo de diploma con título “**Sistema informático para la gestión de la información de los procesos de la UJC en la Facultad 2**” conceden a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declaran como únicos autores de su contenido. Para que así conste firman la presente a los 27 días del mes de noviembre del año 2023.

Lianne Guanche Olivera

Firma del Autor

Sergio Luis Reyna Pérez

Firma del Autor

P. As. Ing. Darling Darías Pérez Ms. C.

Firma del Tutor

Ing. Claudia Amanda García Gómez

Firma del Tutor

DATOS DE CONTACTO

¹Nombre y Apellido: Darling Darías Pérez

Graduada de Ingeniera en Ciencias Informáticas. Profesora Principal de 5to año en la carrera de ICI de la Facultad 2. Posee 15 años de experiencia laboral impartiendo clases de las asignaturas de Ingeniería de Software, Gestión de Software, Base de datos, Proyecto de Investigación y Desarrollo e Introducción a las Ciencias Informáticas. De ellos, 6 años como directiva en un Centro y luego en la facultad. (Subdirectora del Centro de Informática Médica: CECIM y Vicedecana de la Facultad 7) Además de tener experiencia en el rol de Analista y probadora de la calidad en dicho centro productivo. Realizó y defendió la maestría externa de Informática en Salud, en el Centro de Cibernética Aplicada a las Matemáticas. Ha participado en eventos científicos y grupos investigativos. Ha tutorado ejercicios de culminación de estudios en ocasiones anteriores.

Correo electrónico: ddarias@uci.cu

Teléfono: 53103636

²Nombre y Apellidos: Claudia Amanda García Gómez

Graduada en el 2022 de Ingeniería en Ciencias Informáticas. Se desempeña como profesora del Departamento de Informática en la asignatura de Sistema de Base de Datos, en las carreras de ICI, ICS y el ciclo corto de ARSI. Primera vez que ejerce el papel de tutora para ejercicios de culminación de estudios.

Correo electrónico: claudiaagg@uci.cu

Teléfono: 55238060

AGRADECIMIENTOS

Sergio

Esta tesis no hubiera sido posible sin el apoyo de esta institución y de todos los que de una forma u otra confiaron en mí en todos los años de estudios me inspiraron a que llegara donde estoy.

A la universidad de ciencias informáticas por abrirme sus puertas y brindarme la oportunidad de convertirme en un profesional.

A mis tutoras Claudia y Darling por estar presentes en cada paso de este hermoso trabajo, su experiencia, paciencia y comprensión en el momento de esta investigación.

Gracias infinitas a mis padres, por premiarme con el regalo de la vida, su amor infinito y su apoyo incondicional. Mi gratitud a mis amigos Henry, Enrique, Kimandy, Oswald, Oscar y Joel, mis hermanos Carlos, Alberto, Arturo y mi tío Reynaldo que ha sido como un padre, por brindarme distintos puntos de vista y por sus experiencias compartidas que me permitieron desarrollarme como persona a lo largo de mi carrera.

Por último pero no menos importante, agradecerme a mí por creer en que podía lograrlo y por no rendirme cuando muchas veces tenía ganas de hacerlo. Así como a tantos que no mencioné porque sería muy extensa mi lista.

Lianne

El presente trabajo es dedicado en especial a mis padres Leisy y Héctor traerme a este mundo y enseñarme el amor verdadero, por estar atrás de mí siempre y por apoyarme en cada una de mis decisiones. A pesar de que ya no estén juntos, siempre se han comprometido en asegurar mi educación y bienestar. Los amo demasiado.

También quiero agradecerle a mi padrastro Raynel por ser mi segundo padre y ayudarme en todo, desde la escuela hasta aprender a cocinar. A mi madrastra Adelita por escuchar y apoyarme en mis decisiones.

A mi abuela Yeyita que siempre fue mi ángel de la guarda. Siempre la llevo en mi corazón y sé que ella me cuida desde arriba.

Les quiero agradecer a mis profesores como a Claudia, mi tutora a la cual volví loca con mis visitas, a Madelín; mi querida, estricta y graciosa profesora de GPI por los consejos que me ha dado, a Darling por su apoyo, a Lianne Guillén, quien me dio base de datos, a mi profe de Economía Política, y otros más que me ayudaron a formarme como ingeniera en ciencias informáticas.

A mis amigos y a los conocidos de mis amigos como Oswald y los compas de Sergio, con quien he compartido momentos únicos. A mi querida Yadelis por siempre estar dispuesta a darme su mano. A todos mis compañeros de aula que me dieron ratos bastante graciosos y divertidos.

A mi Señor todopoderoso que siempre está conmigo y me guía en cada paso que doy.

Y por último, a todos los que hicieron que mi sueño se cumpliera.

DEDICATORIA

A mis padres Concepcion Reyna Matos y Mileny Pérez Ruiz.

A la Universidad de Ciencias Informáticas por hacer de mi este profesional.

Sergio

A mi mamá y mi papá por ser mi ejemplo y guía.

A Raynel por ayudarme en todo lo que podía.

A mi abuela que siempre estuvo conmigo y ahora me cuida desde el cielo.

A mis profesores y compañeros que me apoyaron durante mi carrera.

A mi amiga del alma Yardelis por ayudarme y estar siempre para mí.

A mi Señor que me sirvió de guía en todo momento y es mi luz en la
oscuridad.

Lianne

RESUMEN

En la Facultad 2 de la Universidad de las Ciencias Informáticas la información relacionada con los procesos de La Unión de Jóvenes Comunistas se gestionan manualmente, ya que no se cuenta con un sistema informático para la organización y control de estos procesos, lo que puede ocasionar retrasos en las entregas de las actas de los Comités de Base, así como pérdidas o duplicidad de datos de los militantes.

En el presente trabajo de diploma se realiza un sistema que permite dar solución a estos problemas, referentes al funcionamiento de la organización. Además, siguiendo AUP-UCI como metodología de desarrollo de software en conjunto con el Lenguaje Unificado de Modelado (UML por sus siglas en inglés), se llevan a cabo la modelación del negocio, el levantamiento de requisitos, el diseño, la implementación y la ejecución de las pruebas.

PALABRAS CLAVE

Comités de Base, gestión, sistema informático, Unión de Jóvenes Comunistas.

ABSTRACT

In Faculty 2 of the University of Computer Sciences, the information related to the processes of the Young Communist League is managed manually, since there is no computer system for the organization and control of these processes, which can cause delays in the deliveries of the minutes of the Base Committees, as well as loss or duplication of data from the militants.

In this diploma work, a computer system is created that allows solutions to these problems, related to the functioning of the organization. In addition, following AUP-UCI as a software development methodology in conjunction with the Unified Modeling Language (UML), business modeling, requirements gathering, design, implementation and implementation are carried out execution of the tests.

KEYWORDS

Base Committees, Communist Youth Union, computer system, management.

Índice de contenido

Introducción	1
Capítulo I: Fundamentación teórica-metodológicas de la gestión de procesos de la UJC.	6
I.1 Conceptos principales.	6
I.2 Funcionamiento de la UCI.	7
I.3 Análisis de soluciones homólogos vinculados a la organización.	8
I.4 Metodología, lenguajes y herramientas.	10
I.4.1 Metodología de desarrollo de software.....	10
I.4.2 Lenguajes empleados.	12
I.4.3 Herramientas empleadas.	17
Conclusiones del capítulo.....	19
Capítulo II: Descripción del sistema SIGUJC	20
II.1 Modelo Conceptual de los procesos de la UJC.	20
II.2 Diccionario de datos.	21
II.3 Modelado de proceso.	29
II.4 Descripción de Procesos de Negocio.	30
II.5 Requisitos del sistema SIGUJC.....	31
II.6 Requisitos no funcionales.....	35
II.7 Descripción de requisitos por procesos de negocio.....	36
II.7.1 Especificación del Requisito <Gestionar Crecimiento>	36
II.8 Arquitectura: Modelo-Vista-Controlador.....	38
II.8.1 Modelo de datos.	39
II.8.2 Patrones de diseño.....	40
II.9 Diagrama de despliegue.....	43
II.10 Estándares de codificación.....	44
II.11 Conclusiones del capítulo.....	46

Capítulo III:	Pruebas y validación del sistema SIGUJC.....	48
III.1	Estrategia de prueba.	48
III.2	Pruebas funcionales.	49
III.2.1	Método de Caja Negra.	49
III.3	Pruebas unitarias.....	50
III.3.1	Método de Caja Blanca.....	50
III.4	Resultados de pruebas.....	51
III.5	Conclusiones del capítulo.....	52
Conclusiones finales.....		53
Recomendaciones.....		54
REFERENCIAS BIBLIOGRÁFICAS		55
ANEXOS.....		57

Índice de tabla

Tabla 1: Sistemas homólogos.....	10
Tabla 2: Diccionario de datos de crecimiento.	22
Tabla 3: Diccionario de datos de desactivación.	24
Tabla 4: Diccionario de datos de planes trimestrales.	25
Tabla 5: Diccionario de datos de acta.....	26
Tabla 6: Diccionario de datos de Comité de Base.	27
Tabla 7: Diccionario de datos de Militante.	28
Tabla 8: Diccionario de datos de Candidato.	29
Tabla 9: Descripción del proceso Crecimiento.....	31
Tabla 10: Especificación de los requisitos funcionales.	34
Tabla 11: Descripción de los requisitos no funcionales.	36
Tabla 12: Especificación de <Gestionar Crecimiento>.	38
Tabla 13: Caso de prueba de partición equivalente del RF13 Agregar permisos del sistema.	50
Tabla 14: Caso de prueba para agregar militante.....	51
Tabla 15: No conformidades por iteración.	51
Tabla 16: Descripción del proceso Elaboración de Planes Trimestrales.	58
Tabla 17: Descripción del proceso Desactivación.	59
Tabla 18: Especificación de <Gestionar Plan Trimestral>.	62
Tabla 19: Especificación de <Gestionar Tipo de Actividad>.	64

Índice de figura

Figura 1: Modelo Conceptual de los procesos de la UJC.	21
Figura 2: Modelado del proceso de crecimiento.	30
Figura 3: Interfaz de Gestionar Militante.	39
Figura 4: Modelo de datos.	40
Figura 5: Ejemplo de experto.	41
Figura 6: Ejemplo de creador.	41
Figura 7: Ejemplo de controlador.	42
Figura 8: Ejemplo de bajo acoplamiento.	42
Figura 9: Ejemplo de contrato uniforme.	43
Figura 10: Ejemplo de patrón observador.	43
Figura 11: Diagrama de Despliegue.	44
Figura 12: Clase MilitanteAusente.	44
Figura 13: Ejemplo de los atributos de un clase.	45
Figura 14: Ejemplo de operadores.	46
Figura 15: Ejemplo de estructuras de control.	46
Figura 16: Código de Agregar Militante.	50
Figura 17: Resultado de pruebas.	52
Figura 18: Modelado del proceso de desactivación.	57
Figura 19: Modelado del proceso elaboración de planes trimestrales.	57

OPINIÓN DEL(OS) TUTOR(ES)

<Contenido de la opinión de los tutores>

AVAL DEL CLIENTE

<Contenido del aval del cliente sobre la solución desarrollada>

INTRODUCCIÓN

En los Estatutos de la Unión de Jóvenes Comunistas (UJC) se plantea que es *“...la organización juvenil del Partido Comunista de Cuba (PCC) que agrupa, dirige y coordina a la vanguardia de la juventud cubana”* (Comunistas, Estatutos de la Unión de Jóvenes Comunistas, 2014). Asume la responsabilidad de dar continuidad a la revolución y al modelo socialista, forjado por varias generaciones de cubanos.

“La Unión de Jóvenes Comunistas se guía por la dirección política del PCC, es orgánicamente independiente y tiene un carácter selectivo, por lo cual agrupa a los jóvenes de vanguardia de todos los sectores de la sociedad que voluntariamente quieran pertenecer a sus filas. Se rige por el principio del centralismo democrático que implica la elegibilidad, la dirección colectiva, la responsabilidad individual, la disciplina consciente, el ejercicio de la crítica y la autocrítica, la subordinación de los órganos y organismos inferiores a los superiores, la rendición de cuentas, la amplia discusión de los asuntos medulares de la Organización y la subordinación de la minoría a la mayoría. Encamina su labor a partir de una estrecha vinculación con las masas juveniles, a las cuales se debe y representa.” (Comunistas, Estatutos de la Unión de Jóvenes Comunistas, 2014).

Esta tiene un órgano de prensa propio, el periódico Juventud Rebelde, y agrupaciones culturales como la Brigada Hermanos Saíz y la Raúl Gómez García. En su trabajo Ideológico, se apoya en organizaciones estudiantiles, como la Federación de Estudiantes de la Enseñanza Media y la Federación Estudiantil Universitaria, e infantiles, como la Organización de Pioneros José Martí.

En la Universidad de las Ciencias Informáticas (UCI), la UJC orienta y forma a sus estudiantes y trabajadores. Teniendo en cuenta lo establecido en los estatutos de la UJC (Comunistas, Estatutos de la Unión de Jóvenes Comunistas, 2014), se determina que está organizada por Comités de Base independientes y por Comités Primarios que tienen a su vez Comités de Base. Cada Comité de Base tiene asociado a un grupo de jóvenes que forman parte del llamado universo juvenil (candidato), que están entre 14 y 32 años y aún no son militantes, pero que pueden aspirar a ingresar a las filas de la organización juvenil.

Durante la entrevista con el cliente se concluyó que, actualmente, los datos de cada joven que milita en la UJC son recogidos de forma manual, lo que implica demora a la hora de realizar alguna consulta, trámite o entrega de información; además, en muchas ocasiones, la información está duplicada apareciendo en más de un documento. El registro de la información es en formato físico o electrónico (Word o Excel) y no se encuentra disponible en todo momento, ni se puede localizar en un lugar específico, al no estar centralizada, lo que trae como consecuencia que exista pérdida de información y demoras en la consulta oportuna para la toma de decisiones. Por otra parte, la información archivada en formato físico, constantemente sufre deterioro por la periodicidad y cantidad de personas que trabajan con los documentos. Además, la planificación de actividades, su cumplimiento, los crecimientos, y chequeo de acuerdos se vuelven procesos complicados, debido a que, se extravían con facilidad los documentos donde se planifican las fechas de realización de algunas actividades o acuerdos emitidos, el responsable de estas, así como otros datos de importancia para su ejecución.

Las insuficiencias antes mencionadas constituyen **problemas de funcionamiento** de la organización juvenil, presentes en la facultad 2.

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

Por lo expuesto anteriormente se tiene el siguiente **problema a resolver**: ¿Cómo contribuir a la disponibilidad y control de la gestión de la información de los procesos permitiendo facilitar el funcionamiento de la UJC en la Facultad 2?

Para dar solución al problema antes planteado se define como **objetivo general**: Desarrollar un sistema informático que contribuya a la disponibilidad y control de la gestión de la información de los procesos referente al funcionamiento de la UJC en la Facultad 2. Se identifica como **objeto de estudio** la gestión de los procesos de la Unión de Jóvenes Comunistas, enmarcado en el **campo de acción**: los procesos de funcionamiento de la UJC en la Facultad 2.

Para dar cumplimiento al objetivo general se establecen como **objetivos específicos**:

1. Establecer los fundamentos y referentes teórico-metodológicos que sustentan la investigación para el desarrollo del sistema.
2. Diseñar una propuesta de solución que permita el acceso y control de algunos procesos dentro del funcionamiento de la UJC en la Facultad 2.
3. Implementar una propuesta de solución que permita el acceso y control de algunos procesos dentro del funcionamiento de la UJC en la Facultad 2.
4. Validar la propuesta de solución cumpliendo con las expectativas del cliente.

Para lograr los objetivos trazados, se llevan a cabo las siguientes **tareas de la investigación**:

1. Fundamentación de la investigación mediante búsquedas bibliográficas.
2. Elaboración del estado del arte mediante la comparación de sistemas similares que responden a los procesos de la UJC.
3. Modelación del flujo de la información de los procesos de la UJC de la Facultad 2.
4. Implementación del sistema de gestión de los procesos de la UJC de la Facultad 2.
5. Validación del sistema de gestión de los procesos de la UJC de la Facultad 2.

Métodos de investigación:

1-Métodos Teóricos:

Analítico-Sintético: Se utilizó para procesar la información relacionada con la organización y planificar una mejor forma de gestionar los procesos de los militantes, así como la manera de clasificar todos los procedimientos.

Modelación: Este método se empleó para la realización de los diagramas para una mejor comprensión del negocio como el modelo conceptual, diagrama de despliegue, modelo de datos, etc.

2-Métodos Empíricos:

Entrevista: Este método se usó para comprender el negocio, además de realizar una correcta captura de requisitos.

Observación: Este método permitió estudiar más de cerca el objeto de la investigación, las acciones, causas y consecuencias.

Resultados y posibles beneficios:

Se obtiene un sistema informático para la gestión de los procesos de la Unión de Jóvenes Comunistas en la Facultad 2 de la Universidad de las Ciencias Informáticas, permitiendo así obtener los siguientes beneficios:

- La agilización de algunos procesos de la UJC en la Facultad 2 para un mejor control y acceso de la información.
- La investigación detallada de los procesos de la UJC en la Facultad 2 de la UCI.
- El ahorro de materiales para el almacenamiento de la información en formato físico, que constituye un beneficio económico.

La investigación está estructurada de forma capitular, consta de introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos.

- **Capítulo 1: Fundamentación teórico-metodológica de la gestión de procesos de la UJC.** En este capítulo se analizan conceptos asociados al problema de investigación, como es el proceso de crecimiento, para una mayor comprensión de los procedimientos de la organización. Se realiza un estudio de sistemas homólogos relacionados con la UJC que respondan a problemas similares. Se presentan los lenguajes de modelado y programación, así como las herramientas y metodologías que se ajustan al desarrollo del trabajo, fundamentando su selección sobre la base del estudio realizado.
- **Capítulo 2: Descripción del sistema SIGUJC.** En este capítulo se describe la propuesta de solución. Se realiza la especificación de los requerimientos funcionales y no funcionales, la modelación del negocio y otros artefactos según la metodología seleccionada.

- **Capítulo 3: Pruebas y validación del sistema SIGUJC.** En este capítulo se describen las iteraciones llevadas a cabo durante la etapa de implementación del sistema, así como las pruebas efectuadas sobre el mismo y el manejo de la seguridad.

CAPÍTULO I: Fundamentación teórica-metodológicas de la gestión de procesos de la UJC.

La UJC, como organización política, tiene lugar en cada centro donde existan al menos cinco militantes y en la mayoría de los casos se encuentra estructurada como sucede en la Universidad de las Ciencias Informáticas (UCI), donde existe un Comité de la UJC al cual se subordinan varios Comités Primarios (CP) como estructuras intermedias entre el Comité UJC y los Comités de Base (CB). Un CP es el encargado de orientar, planificar, organizar, dirigir y controlar el trabajo de la UJC en cada una de las facultades de la UCI.

Luego de revisar la bibliografía consultada, se asumen los conceptos planteados en los Estatutos de la Unión de Jóvenes Comunistas del 2014.

I.1 Conceptos principales.

-Comité de Base: estructura más baja y principal de la organización, donde se desarrollan todos los procesos sustantivos de la UJC.

-Militante: persona que integra la organización, la cual participa en los procesos sustantivos que desarrolla la UJC.

-Candidato: miembro del universo juvenil que aspira a ser militante o es un posible candidato a ingresar a las filas de la organización.

-Secretario General: es la máxima autoridad de la UJC en la facultad, es el responsable de todas las tareas que se lleven a cabo en la organización. Se le subordinan otros cargos como el organizador, activista de funcionamiento, entre otros.

-Activista de Acta (Act. Acta): Es el encargado de confeccionar correctamente y entregar el acta de la reunión al secretario general.

-Evaluación: proceso dirigido a la búsqueda de elementos importantes sobre la actitud del aspirante a militante.

-Crecimiento: procedimiento mediante el cual se captan nuevos miembros para la organización. Militan los jóvenes cubanos de 16 a 32 años que cumplan sus estatutos y reglamentos, pertenezcan y actúen en una de sus organizaciones de base o del partido.

-Sanciones: medidas disciplinarias que se aplican a militantes que incumplen con sus deberes.

-Desactivación: procedimiento donde los militantes abandonan o dejan de pertenecer a la organización. Ocurre por sanciones o por edad. El militante que infrinja los principios y normas de la UJC o de la sociedad, deberá ser sancionado y podrá cumplir la sanción fuera o dentro de la organización. Los militantes de la UJC que al llegar a los 32 años de edad no sean procesados por el partido o, realizado el proceso no ingresen a sus filas, y las causas no estén asociadas a problemas de principios, errores o indisciplinas cometidas que impliquen la aplicación de una sanción externa, se procederá a su desactivación.

-Planes trimestrales: es un modelo que se elabora antes de realizar una actividad, en un período de 3 meses. Se redactan en las reuniones del Comité de Base y comprenden:

- a) Tema o temas.
- b) Objetivos.
- c) Factores del centro a invitar.
- d) Fecha, hora y lugar de realización.
- e) Responsable.

Y otros elementos de interés.

-Acta: documento donde queda constancia de todos los puntos analizados en la reunión.

-Actividad: conjunto de tareas propias de la organización.

I.2 Funcionamiento de la UCI.

El proceso de crecimiento puede iniciar de dos formas: por medio de la Asamblea de Ejemplares, donde se genera un acta, o por una solicitud personal del aspirante a militante de la UJC (candidato), que genera una carta. En el primero, se reúnen los miembros del Comité de Base para debatir los posibles candidatos y, en el segundo, es el propio solicitante quien se auto propone para comenzar el proceso. Posteriormente, en ambos casos, el Comité de Base revisa si cumplen con los requisitos establecidos en el reglamento de la organización, por medio de la evaluación, donde se contempla una entrevista, cuestionarios de datos socio - demográficos y otros elementos. Si determinan que no satisfacen estos, no entraran a sus filas; en caso contrario, se presentará la propuesta al Comité Primario. Este se encargará de preparar la entrega del carné de la UJC al nuevo miembro de la organización. (Comunistas, Estatutos de la Unión de Jóvenes Comunistas, 2014).

El proceso de desactivación se inicia con la reunión del Comité de Base presentándose por dos vías: por edad (desactivación natural) o por alguna indisciplina realizada por uno o más militantes. Entonces se procesa al militante y en ambos casos se genera un acta. En el primer caso: se comete alguna indisciplina, el documento será enviado al Comité Primario para determinar la sanción. Dependiendo de la gravedad de esta, el militante será sancionado con amonestación, suspensión temporal de derechos o expulsión permanente de la organización. En un segundo caso: cuando los militantes alcanzan los 32 años de edad, se procede a su desactivación de forma natural. (Comunistas, Estatutos de la Unión de Jóvenes Comunistas, 2014).

Además de estos, está el proceso de elaboración de planes trimestrales, que se lleva a cabo en una de las reuniones del Comité de Base. Se genera un acta, donde se plasman los acuerdos, dejando registros relacionados con el tema, objetivos, fecha, hora, lugar, responsable, entre otros datos de interés. Este documento se envía al Comité Primario para su aprobación, principalmente por los temas y la fecha. Una vez se tiene su consentimiento, se realizan los preparativos para cumplir con este plan. (Comunistas, Estatutos de la Unión de Jóvenes Comunistas, 2014).

I.3 Análisis de soluciones homólogos vinculados a la organización.

A nivel internacional no se tienen conocimientos precedentes de sistemas relacionados con la UJC debido a que esta es una organización creada en Cuba, con el objetivo de agrupar a la vanguardia de la juventud cubana. A nivel nacional los sistemas que existen y que pudieran analizarse como antecedentes de esta investigación, son en su mayoría portales Web y sistemas que solo informatizan una parte de un proceso sustantivo o un proceso en general. A continuación, se realiza un análisis del estudio de algunos de estos sistemas.

-Sistema integral de gestión de la UJC nacional módulo “Militante”: fue creado con la finalidad de formar parte de un sistema integral de la UJC Nacional y contribuir a la mejora de la gestión de la información de los militantes y el universo juvenil de la organización (Carlos Montenegro Amador, 2015). Este módulo automatiza los siguientes procesos:

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

- Gestión de centros.
- Gestión de los datos de cada miembro del universo juvenil.
- Realizar crecimiento del universo juvenil.

-Sistema integral de gestión de la UJC UCI módulo “Militante”: fue creado en la Facultad 1 para formar parte de un sistema integral de gestión de la UJC en la UCI y permitir introducir mejoras en la gestión de la información de los militantes y el universo juvenil (Carlos Montenegro Amador, 2015).

-Sitio web PCC: es una plataforma web del Partido Comunista de Cuba que brinda información acerca de la historia de este, las noticias más recientes, entre otras funcionalidades. (Universidad de Ciencias Informáticas, 2021)

-Sistema Informático para la gestión de la información de la Unión de Jóvenes Comunistas de la Facultad 2: este sistema permite gestionar los militantes y profesores de la facultad 2, además de generar reportes de ID2 para los Comités de Base compuestos por estos militantes. (Carlos Montenegro Amador, 2015).

Sistemas	Sistema de Gestión	Multiplataforma	Ambiente Web	Proceso de Crecimiento	Proceso de Desactivación	Proceso Elaboración de Planes Trimestrales
Sistema Integral de Gestión de la UJC Nacional_ Módulo Militante	Sí	Sí	Sí	Sí	No	No
Sistema Integral de Gestión de la UJC UCI Módulo Militante	Sí	Sí	Sí	Sí	Sí	No

Sistema Informático para la Gestión de la Información de La Unión de Jóvenes Comunistas de la Facultad 2	Sí	Sí	Sí	Sí	No	No
Sitio web PCC	Sí	Sí	Sí	No	No	No

Tabla 1: Sistemas homólogos.

Estos sistemas no cumplen con los requisitos necesarios para la informatización de los procesos de la UJC. Entre sus principales limitantes está que no se llegó al desarrollo de la solución propuesta a las investigaciones analizadas. Además, no realizan los procesos de la UJC como la desactivación de los militantes y la elaboración de los planes trimestrales, que son de interés para esta investigación. Debido a ello, se ha propuesto el desarrollo de un nuevo sistema para el acceso y control de la información de los procesos de la UJC en la Facultad 2.

I.4 Metodología, lenguajes y herramientas.

En el proceso de desarrollo de software adaptado a la situación nacional debe tener como característica fundamental que el proceso debe ser libre de costos en cuanto a patentes de desarrollo de software, asociadas a los servidores de aplicaciones y de base de datos, al sistema operativo del servidor u otras herramientas y tecnologías utilizadas.

Para el desarrollo de la solución se utilizaron herramientas, lenguajes, arquitectura y metodología que se describen en el siguiente epígrafe y se describe como interceden cada una de ellas en el proceso de desarrollo de software.

I.4.1 Metodología de desarrollo de software.

Según Pressman “Se define metodología de desarrollo de software como un conjunto de procedimientos, técnicas, herramientas y un soporte documental para el desarrollo de productos de software” (Pressman, Ingeniería del software. Un enfoque práctico. Quinta

Edición., 2005). Actualmente las metodologías más utilizadas se dividen en Metodologías Ágiles y Metodologías Tradicionales. Las ágiles promueven iteraciones durante todo el ciclo de vida del proyecto. Existen muchos métodos de desarrollo ágil; la mayoría minimiza riesgos desarrollando software en poco tiempo, debido a que en la actualidad los entornos de los sistemas son muy cambiantes y se exige reducir drásticamente los tiempos de desarrollo manteniendo una alta calidad. Las tradicionales o pesadas son metodologías con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado. Estas se centran en la definición detallada de los procesos mediante una estricta definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada del producto. (Thais Sánchez Martínez, 2015)

Después de haber estudiado las metodologías, el equipo de desarrollo decidió utilizar la metodología de desarrollo de software el Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process* en inglés en su variante UCI, apoyada en el modelo CMMI-DEV v 1.3 que constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora. Estas prácticas se centran en el desarrollo de productos y servicios de calidad.

En esencia AUP-UCI es una versión simplificada del Proceso Unificado de Rational (RUP), adaptándose a características más específicas del negocio local. Se acortan las fases del proceso de desarrollo quedando en: Inicio, Ejecución y Cierre.

Dentro de ella se escoge el escenario 3, debido a que aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad. (Alberto, 2017). Junto a la aplicación de esta metodología se emplea como lenguaje de modelado a *Unified Modeling Language* (UML), utilizado para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Permite la modelación de sistemas con tecnología orientada a objetos. Se puede aplicar en el desarrollo de software entregando 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 utilizar.

Este lenguaje de modelado formal permite tener un mayor rigor en la especificación, realizar una verificación y validación del modelo desarrollado, automatizar determinados procesos y generar código a partir de los modelos y a la inversa. Esto último permite que el modelo y el código estén actualizados.

El escenario empleado trabaja con modelo de proceso de negocio por lo que se decide emplear la notación Modelo y Notación de Procesos de Negocio (BPMN, *Business Process Management Notation*) es un estándar de modelado de procesos de negocio donde se presentan gráficamente las diferentes etapas de su proceso. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes.

Antes de conocer para qué se utiliza esta notación, es necesario resaltar en qué consiste un proceso de negocio. Este es una colección de actividades que, tomando una o varias clases de entradas, crean una salida que tiene valor para un cliente. Los procesos de negocio representan el flujo de trabajo y de información a través del negocio.

El objetivo principal de esta notación es mejorar la eficiencia a través de la gestión sistemática de los procesos de negocio que se deben modelar, automatizar, integrar, monitorizar y optimizar de forma continua. A través del modelado de las actividades y los procesos puede lograrse un mejor entendimiento del negocio. Muchas veces esto brinda un mejor enfoque lo que permite mejorarlos.

I.4.2 Lenguajes empleados.

Lenguaje de Programación Python

Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el *machine learning* (ML). En el desarrollo de software Python es renombrado gracias a su eficiencia y facilidad a la hora de aprender, además de que es multiplataforma. Python es de código abierto y sus componentes se pueden descargar gratis. (Inc. Amazon Web Services, 2023).

Se escoge como lenguaje de programación debido a su gran facilidad para realizar aplicaciones con soporte en la web, porque es de código abierto y el desarrollo empleándolo es totalmente libre de costos.

Django

Django es un *framework* utilizado para desarrollar aplicaciones web de forma rápida y eficiente. La mayoría de las aplicaciones web tienen varias funciones comunes, como la autenticación, la recuperación de información de una base de datos y la administración de *cookies*. Los desarrolladores tienen que codificar una funcionalidad similar en cada aplicación web que escriban. Django facilita su trabajo al agrupar las diferentes funciones en una gran colección de módulos reutilizables, llamada marco de aplicación web. Los desarrolladores utilizan el marco web de Django para organizar y escribir su código de manera más eficiente y reducir significativamente el tiempo de desarrollo web. (Inc. Amazon Web Services, 2023).

Fue empleado primeramente porque dentro de los *frameworks* que posee Python es uno de los más estables, además que permite realizar aplicaciones de fácil mantenimiento y el desarrollo sobre este es relativamente sencillo.

API-REST

Una REST API, o API RESTful, es una interfaz de programación de aplicaciones (API o API web) que se ajusta a los límites de la arquitectura REST y permite la interacción con los servicios web de RESTful.

Las API son conjuntos de definiciones y protocolos que se utilizan para diseñar e integrar el software de las aplicaciones. Suele considerarse como el contrato entre el proveedor de información y el usuario, donde se establece el contenido que se necesita por parte del consumidor (la llamada) y el que requiere el productor (la respuesta). Por ejemplo, el diseño de una API de servicio meteorológico podría requerir que el usuario escribiera un código postal y que el productor diera una respuesta en dos partes: la primera sería la temperatura máxima y la segunda, la mínima. (Inc. Amazon Web Services, 2023).

En otras palabras, las API permiten interactuar con una computadora o un sistema para obtener datos o ejecutar una función, de manera que el sistema comprenda la solicitud y la cumpla.

Django Rest Framework

Django Rest Framework es una aplicación Django que permite construir proyectos software bajo la arquitectura REST, incluye gran cantidad de código para reutilizar (*Views*, *Resources*, etc.) y una interfaz administrativa desde la cual es posible realizar pruebas sobre las operaciones HTTP como lo son: POST y GET. (Inc. Amazon Web Services, 2023).

Se empleó ya que se escogió Django y al ser esta una API-REST la vía idónea era haciendo uso del mismo.

PostgreSQL 14.0.1

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacional (*Object-Relational Database Management System* (ORDBMS)) que no tiene costo asociado por lo que se puede disponer de su código fuente, modificarlo y redistribuirlo libremente. PostgreSQL garantiza concurrencia, para lo cual utiliza la tecnología de Control de Concurrencia Multi-Versión (*Multiversion concurrency control* (MVCC)), con lo que se logra que ningún lector sea bloqueado por un escritor. Es extensible, soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario. Tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/PGSQL. Este lenguaje es comparable al lenguaje procedural del sistema de gestión de base de datos relacional Oracle, PL/SQL. En cuanto a sus funciones, poseen bloques de código que se ejecutan en el servidor los cuales pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos brinda. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que se intente conectar a la base de datos. Tiene una adecuada documentación, además de contar con una comunidad de usuarios y desarrolladores a los que acudir en caso de tener problemas. (Platzi, 2023).

Fue escogido por su versatilidad y fácil empleo. Además que es uno de los Sistemas de Base de Datos más populares en la actualidad y va tono con las tecnologías escogidas.

PSYCOPG2

Psycopg2 es una librería de Python que proporciona una interfaz para conectarse y trabajar con bases de datos PostgreSQL. Con ella, se pueden ejecutar consultas SQL, realizar transacciones, manejar errores entre otras funciones. (Inc. Amazon Web Services, 2023)

Fue empleado por ser el componente conector entre Django y el Sistema Gestor de Base de Datos PostgreSQL.

JavaScript

JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (*just-in-time*) con funciones de primera clase. Si bien es más conocido como un lenguaje de *scripting* (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador. JavaScript es un lenguaje de programación basado en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa. (MDN., 2023).

Fue empleado en el sistema debido a que dentro de la Web y de los lenguajes del lado del cliente es la más versátil.

Node JS

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. (Inc. Amazon Web Services, 2023)

Fue empleado ya que es el motor fundamental para que las tecnologías empleadas se comuniquen.

TypeScript

Typescript es un *superset* de JavaScript. Decimos que una tecnología es un *superset* de un lenguaje de programación, cuando puede ejecutar programas de la tecnología, Typescript en este caso, y del lenguaje del que es el *superset*, JavaScript en este mismo ejemplo. En resumen, esto significa que los programas de JavaScript son programas válidos de TypeScript, a pesar de que TypeScript sea otro lenguaje de programación. (Inc. Amazon Web Services, 2023).

Se usó para en darle vida fundamentalmente a los elementos del *Framework* AngularJS.

HTML5

HTML5 es la quinta revisión importante del lenguaje básico de la *World Wide Web*, HTML. HTML5 especifica dos variantes de sintaxis para HTML: una «clásica», HTML, y una variante XHTML conocida como sintaxis XHTML5 que deberá servirse con sintaxis XML. (Inc. Amazon Web Services, 2023)

El uso del HTML es generalizado para todos los sistemas que corran sobre la web. Se utilizó para la realización de las interfaces.

Angular

Angular es un potente *framework*, basado en el lenguaje TypeScript, mantenido por Google y modificado desde una versión anterior nombrada AngularJS. Se basa en módulos y componentes que junto al sistema de *imports* (importación de módulos) permite crear proyectos con una estructura ordenada y limpia evitando el conocido "código espagueti. Angular es un proyecto escalable, flexible y de código abierto y se ha convertido en uno de los *frameworks* más populares de los últimos tiempos. (Inc. Amazon Web Services, 2023).

Su uso permitió realizar un sistema ameno para el usuario y de fácil empleo.

Angular CLI

Angular CLI (*Command Line Interface*) es una herramienta del propio Angular que permite crear el árbol de directorios y archivos necesarios (*scaffolding*) para un nuevo proyecto Angular, esto se hace a través de una línea de comandos. (Inc. Amazon Web Services, 2023).

Se utilizó para desarrollar y estructurar el sistema directamente desde la terminal.

CSS3

Las hojas de estilo en cascada versión 3 son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (*World Wide Web Consortium*) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. Lo que se persigue con el desarrollo de CSS es separar la estructura de un documento de su presentación. (HACK A BOSS., 2023).

Se empleó para darle estilo a todo el sistema.

I.4.3 Herramientas empleadas.

Pycharm

Pycharm es un Entorno de Desarrollo Integrado (IDE) especializado en el lenguaje de programación Python y es desarrollado por JetBrains. Pycharm es el IDE más popular para Python hasta la fecha. Esta plataforma híbrida se utiliza habitualmente para el desarrollo de aplicaciones en Python, principalmente por grandes empresas como X, Facebook, Amazon y Pinterest. (alexandre., 2022).

Se uso fue muy importante para el desarrollo del sistema debido a que proporciona indicación de errores sobre la marcha y arreglos rápidos, así como refactorización de código automática y completas funcionalidades de navegación.

IntelliJIDEA

IntelliJ IDEA es un entorno de desarrollo integrado para el desarrollo de programas informáticos. Es desarrollado por JetBrains, y está disponible en dos ediciones: edición para la comunidad y edición comercial. Es especializado en el lenguaje de programación Java pero tiene soporte para varios lenguajes incluyendo JavaScript y TypeScript. (Inc. Amazon Web Services, 2023).

Esta herramienta permite sugerir opciones de finalización de código, generar mensajes de confirmación, trabajar con elementos de código de otros lenguajes incrustados en el

suyo sin notar la diferencia, y muchas más funcionalidades, por lo cual decidimos utilizarlo.

PgAdmin4

PgAdmin es una herramienta indispensable para gestionar y administrar PostgreSQL, la base de datos de código abierto más avanzada del mundo. Por lo tanto PgAdmin es la herramienta para gestionar nuestras bases de datos espaciales PostGIS. Cabe destacar que esta versión del software es proporcionada por un servicio Web. (Morales, 2017).

Se usó, en conjunto con PostgreSQL, para gestionar, administrar, desarrollar y mantener las bases de datos.

Visual Paradigm for UML

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. El software de modelado UML ayuda a la construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. (Pressman, 2005).

Permitió confeccionar los diagramas necesarios para el diseño del sistema como modelo conceptual, diagrama de despliegue, entre otros más.

Servidor Web - Apache 2.2

Un servidor web es un programa que sirve datos en forma de Páginas Web, hipertextos o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. (Bustos, 2023)

La comunicación de estos datos entre cliente y servidor se hace por medio un protocolo, concretamente del protocolo HTTP. Apache es uno de los servidores web más utilizados, posiblemente porque ofrece instalaciones sencillas para sitios pequeños y si se requiere es posible expandirlo hasta el nivel de los mejores productos comerciales. Además el

servidor HTTP y de código abierto para las plataformas Windows, Mac OS X y UNIX (GNU, BSD, etc.).

Conclusiones del capítulo

El análisis de los principales conceptos asociados al objeto y campo de la investigación permitió arribar a las conclusiones siguientes:

1. Luego de revisar documentación correspondiente a la UJC quedan definidos los procesos sustantivos a trabajar durante la investigación, así como los elementos que conforman estos procesos.
2. No existe una solución informática que satisfaga las problemáticas que generaron la investigación.

CAPÍTULO II: Descripción del sistema SIGUJC

El SIGUJC (Sistema Informático de Gestión de La Unión de Jóvenes Comunistas) está basado en los estatutos del 2014 vigentes y por los que se rige la UJC. Es un sistema web que tiene como objetivo facilitar el acceso y control de la información de los procesos de la UJC. Entre las funcionalidades del sistema están:

- La gestión del proceso de crecimiento de los militantes de la UJC.
- La gestión del proceso de desactivación de los militantes de la UJC.
- La gestión del proceso de elaboración de planes trimestrales con sus respectivas actividades.
- La asignación de cada militante a los Comités de Base.
- La gestión de las actas de las reuniones.

Para el funcionamiento del sistema debe existir al menos un Comité de Base presente en la facultad.

II.1 Modelo Conceptual de los procesos de la UJC.

En este modelo se definen cuáles son y cómo se relacionan los conceptos relevantes en la descripción del problema, en este caso describe los conceptos relacionados con el negocio del sistema SIGUJC.

El Comité de Base tiene candidatos, los cuales son aspirantes a militantes, y está integrado por militantes, quienes participan en los procesos sustantivos que desarrolla la UJC. El secretario general y el activista de acta son militantes que se encargan de manejar la información referente a los procesos de la organización. El primero, junto con demás miembros que se le subordinan, gestiona el crecimiento y requiere una evaluación del solicitante para que integre las filas de esta. También gestiona la desactivación, que puede ser producida por sanción o por edad, donde el militante abandona o es expulsado de la UJC. Además, el activista de acta confecciona el acta de las reuniones del Comité de Base y los planes trimestrales, los cuales contienen diversas actividades.

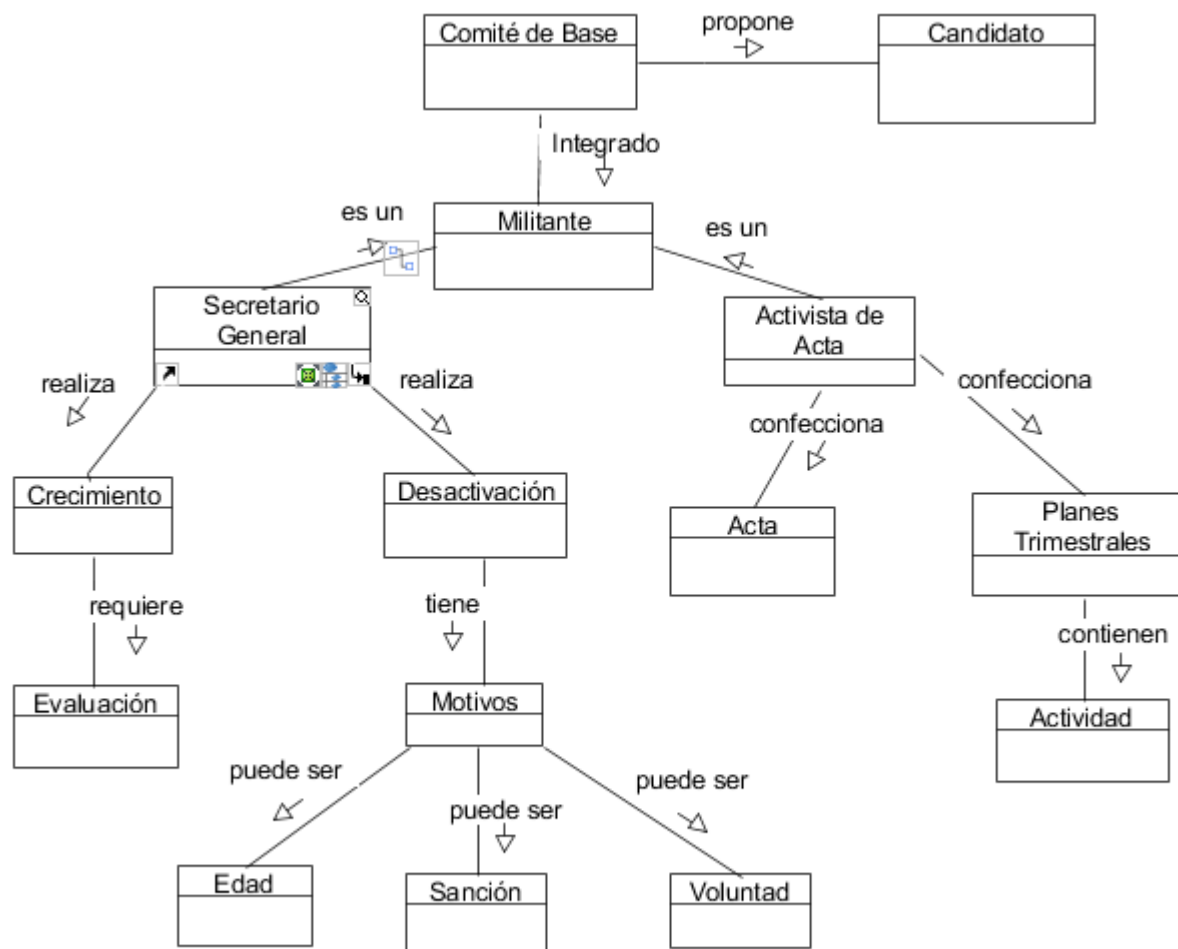


Figura 1: Modelo Conceptual de los procesos de la UJC.

II.2 Diccionario de datos.

Un diccionario de datos, o repositorio de metadatos, como lo define el IBM *Dictionary of Computing*, un repositorio centralizado de información sobre datos tales como significado, relación con otros datos, origen, uso y formato. (McDaniel, G. y International Business Machines Corporation, 2015).

A continuación, se presenta el diccionario de datos de crecimiento.

Descripción	Procedimiento mediante el cual se captan nuevos miembros para la organización. Militan los jóvenes cubanos de 16 a 32 años que cumplan
-------------	--

	sus estatutos y reglamentos, pertenezcan y actúen en una de sus organizaciones de base o del partido.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
Nombre de persona	Nombre del aspirante a militante	Cadena de caracteres	No	No	A-Z, a-z, espacio	Caracteres extraños
Comprobadores	Miembros de la UJC que procesan al aspirante a militante	Cadena de caracteres	No	No	A-Z, a-z, espacio	Caracteres extraños
Avales	Valoración integral del militante	Cadena de caracteres	No	Sí	A-Z, a-z, espacio	Caracteres extraños

Tabla 2: Diccionario de datos de crecimiento.

Descripción	Procedimiento donde los militantes abandonan o dejan de pertenecer a la organización. Ocurre por sanciones o por edad. El militante que infrinja los principios y normas de la UJC o de la sociedad, deberá ser sancionado y podrá cumplir la sanción fuera o dentro de la organización. Los militantes de la UJC que al llegar a los 32 años de edad no sean procesados por el partido o,
-------------	--

	realizado el proceso no ingresen a sus filas, y las causas no estén asociadas a problemas de principios, errores o indisciplinas cometidas que impliquen la aplicación de una sanción externa, se procederá a su desactivación.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único ?	Restricciones	
					Clases válidas	Clases no válidas
Medida propuesta	Propuesta de medida.	Cadena de caracteres	No	No	A-Z, a-z, espacio	Caracteres extraños
Militante	Militante sancionado.	Cadena de caracteres	No	No	A-Z, a-z, espacio	Caracteres extraños
Comité de Base	Comité de Base al que pertenece el militante sancionado.	Cadena de caracteres	No	Sí	A-Z, a-z, espacio	Caracteres extraños
Causa	Causa de medida.	Cadena de caracteres	No	No	A-Z, a-z, espacio	Caracteres extraños
Fecha de aplicada	Fecha en que se aplicó la medida.	Fecha	No	Sí	DD/MM /AAAA	No aplica

Tabla 3: Diccionario de datos de desactivación.

Descripción	Es un modelo que se elabora antes de realizar una actividad, en un período de 3 meses. Se redactan en las reuniones del Comité de Base y comprenden: tema o temas, objetivos, factores del centro a invitar, fecha, hora y lugar de realización, responsable y otros elementos de interés.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único ?	Restricciones	
					Clases válidas	Clases no válidas
Fecha	Fecha en que se redactó.	Fecha	No	Sí	DD/MM /AAAA	No aplica
Militante	Responsable.	Cadena de caracteres	No	No	A-Z, a-z, espacio	Caracteres extraños
Observaciones	Observaciones durante la actividad.	Cadena de caracteres	No	Sí	A-Z, a-z, espacio	Caracteres extraños
Nombre de actividad	Nombre de la actividad a realizar.	Cadena de caracteres	No	No	A-Z, a-z, espacio	Caracteres extraños
Fecha	Fecha en que se realizará.	Fecha	No	Sí	DD/MM /AAAA	No aplica

Tipo de actividad	Tipo de actividad a realizar.	Cadena de caracteres	No	Sí	A-Z, a-z, espacio	Caracteres extraños
-------------------	-------------------------------	----------------------	----	----	-------------------	---------------------

Tabla 4: Diccionario de datos de planes trimestrales.

Descripción	Documento donde queda constancia de todos los puntos analizados en la reunión.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único ?	Restricciones	
					Clases válidas	Clases no válidas
Actividad	Nombre de la actividad propuesta.	Cadena de caracteres	No	Sí	A-Z, a-z, espacio	Caracteres extraños
Visitante del nivel superior	Invitado del nivel superior.	Cadena de caracteres	No	Sí	A-Z, a-z, espacio	Caracteres extraños
Asistencia	Cantidad de militantes que asistieron a la reunión.	Numérico	No	No	No aplica	Caracteres extraños
Total	Cantidad de militantes que	Numérico	No	No	No aplica	Caracteres extraños

	deben estar en la reunión.					
Cuerpo de acta	Acuerdos a los que se arribaron.	Cadena de caracteres	No	No	A-Z, a-z, espacio	Caracteres extraños

Tabla 5: Diccionario de datos de acta.

Descripción	Estructura más baja y principal de la organización, donde se desarrollan todos los procesos sustantivos de la UJC.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único ?	Restricciones	
					Clases válidas	Clases no válidas
Nombre del Comité de Base	Nombre del Comité de Base.	Cadena de caracteres	No	Sí	A-Z, a-z, espacio	Caracteres extraños
Organismo	Organismo al que pertenece.	Cadena de caracteres	No	Sí	A-Z, a-z, espacio	Caracteres extraños
Institución	Institución al que pertenece.	Cadena de caracteres	No	Sí	A-Z, a-z, espacio	Caracteres extraños

Secretario General	Acuerdos a los que se arribaron.	Cadena de caracteres	No	Sí	A-Z, a-z, espacio	Caracteres extraños
--------------------	----------------------------------	----------------------	----	----	-------------------	---------------------

Tabla 6: Diccionario de datos de Comité de Base.

Descripción	Persona que integra la organización, la cual participa en los procesos sustantivos que desarrolla la UJC.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único ?	Restricciones	
					Clases válidas	Clases no válidas
Carnet de militante	Carnet de militancia.	Cadena de caracteres	No	Sí	No aplica	Caracteres extraños
Persona	Nombre de la persona.	Cadena de caracteres	No	Sí	A-Z, a-z, espacio	Caracteres extraños
Comité de Base	Nombre del Comité de Base.	Cadena de caracteres	No	Sí	No aplica	Caracteres extraños
Clasificador ocupacional	Ocupación.	Cadena de caracteres	No	No	No aplica	Caracteres extraños
Fecha alta	Fecha en que ingresó.	Fecha	No	No	DD/MM /AAAA	No aplica

Tabla 7: Diccionario de datos de Militante.

Descripción	Miembro del universo juvenil que aspira a ser militante o es un posible candidato a ingresar a las filas de la organización.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único ?	Restricciones	
					Clases válidas	Clases no válidas
Usuario	Nombre de usuario.	Cadena de caracteres	No	Sí	No aplica	Caracteres extraños
Nombre	Nombre del candidato.	Cadena de caracteres	No	Sí	A-Z, a-z, espacio	Caracteres extraños
Apellidos	Apellidos del candidato.	Cadena de caracteres	No	Sí	A-Z, a-z, espacio	Caracteres extraños
Carnet de identidad	Carnet de identidad del militante.	Cadena de caracteres	No	No	No aplica	Caracteres extraños
Sexo	Género del candidato.	booleano	No	Sí	F, M	Caracteres extraños
Dirección	Dirección de residencia.	Cadena de caracteres	No	Sí	A-Z, a-z,	Caracteres extraños

					espacio , número s	
Municipio	Municipio de residencia.	Cadena de caracteres	No	Sí	A-Z, a- z, espacio	Caracteres extraños
Provincia	Provincia de residencia.	Cadena de caracteres	No	Sí	A-Z, a- z, espacio	Caracteres extraños
País	País de residencia.	Cadena de caracteres	No	Sí	A-Z, a- z, espacio	Caracteres extraños

Tabla 8: Diccionario de datos de Candidato.

II.3 Modelado de proceso.

En la figura siguiente se muestra el modelado del proceso de crecimiento. Los otros se encuentran en los anexos.

Este proceso puede iniciar de dos formas: por medio de la Asamblea de Ejemplares, donde se genera un acta, o por una solicitud personal del aspirante a militante de la UJC (candidato), que genera una carta. En el primero, se reúnen los miembros del Comité de Base para debatir los posibles candidatos, además de hablar con estos para saber si están de acuerdo, y, en el segundo, es el propio solicitante quien se auto propone para comenzar el proceso. Posteriormente, en ambos casos, los comprobadores, miembros del Comité de Base, revisan si cumplen con los requisitos establecidos en el reglamento de la organización, por medio de la evaluación, donde se contempla una entrevista, cuestionarios de datos socio - demográficos y otros elementos. Si determinan que no satisfacen estos, no entraran a sus filas; en caso contrario, se presentará la propuesta al Comité Primario. Este se encargará de preparar la

entrega del carné de la UJC al nuevo miembro de la organización. (Comunistas, Estatutos de la Unión de Jóvenes Comunistas, 2014)

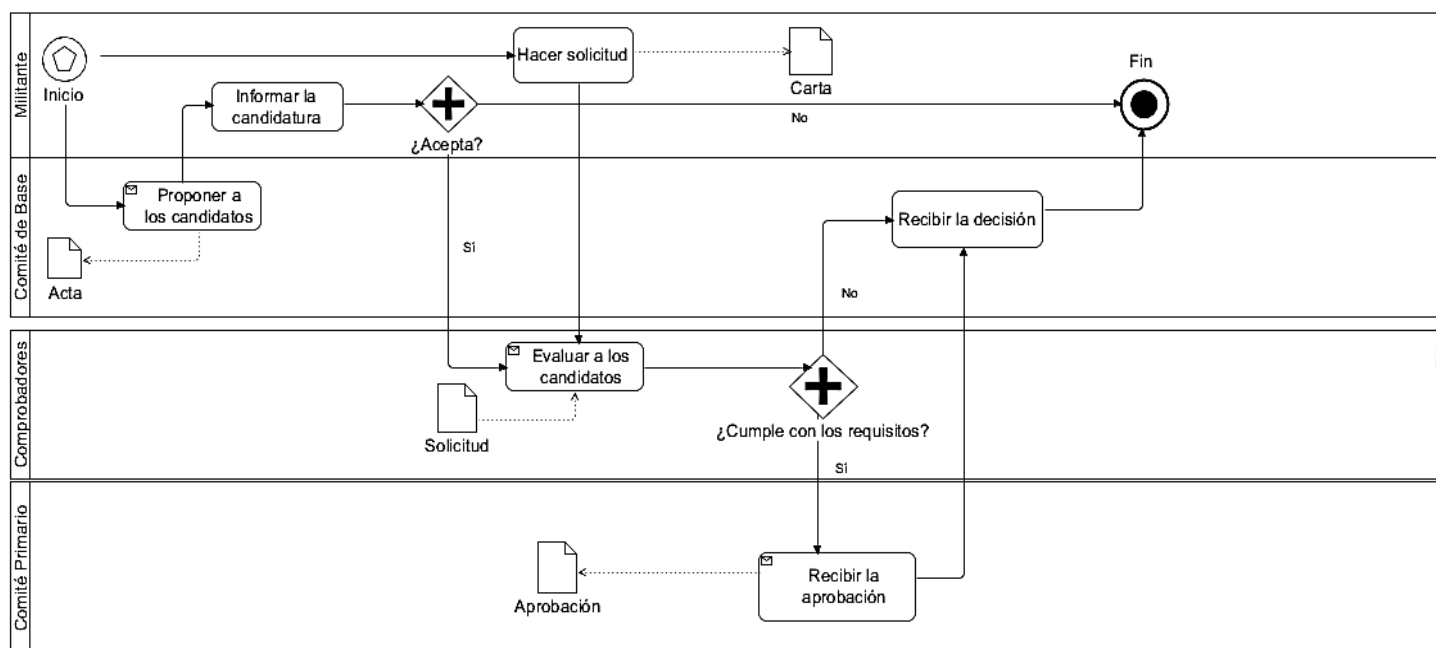


Figura 2: Modelado del proceso de crecimiento.

II.4 Descripción de Procesos de Negocio.

En la siguiente tabla se muestra la descripción del proceso Crecimiento donde se describe el mismo para un mayor entendimiento por parte del desarrollador de la presente investigación.

Nombre del proceso	Crecimiento
Objetivo	Realizar el crecimiento de un militante.
Evento(s) que lo genera(n)	Necesidad de registrar el crecimiento de un militante.
Pre condiciones	Debe existir la persona a la cual se le realizará el crecimiento.
Responsable	Secretario General
Clientes internos	Comité de Base
Clientes externos	N/A
Entradas	Datos
Flujo de eventos	

Flujo básico		
1. Se propone al candidato militante.	al a	En la Asamblea de Ejemplares, el Comité de Base presenta a los posibles militantes, después de conversarlo con ellos. El aspirante a militante solicita unirse a la organización por voluntad propia.
2. Se realiza una evaluación del candidato.	una del	Los comprobadores revisan si cumplen con los requisitos establecidos en el reglamento de la organización, teniendo en cuenta una entrevista, cuestionarios de datos socio - demográficos y otros elementos.
3. Se envía la aprobación al Comité Primario.	la al	Se encarga de preparar la entrega del carné de la UJC al nuevo miembro de la organización y se lo informa al Comité de Base, el cual se lo dice al militante.
Postcondiciones		
Se agrega un militante a la UJC.		
Salidas		
Nuevo miembro de la UJC.		
Flujos alternos		
3a.Si no cumplen con los requisitos establecidos.		
1. Se le notifica al Comité de Base, quien le informa al candidato que no puede ingresar a la organización.		
Postcondiciones		
No ingresa a las filas.		
Salidas		
No aplica.		

Tabla 9: Descripción del proceso Crecimiento.

Las demás tablas que describen los otros procesos se encuentran referenciadas en los anexos.

II.5 Requisitos del sistema SIGUJC.

Utilizando la entrevista como método, para SIGUJC se identificaron los siguientes requisitos funcionales:

No.	Nombre	Descripción	Prioridad	Complejidad
-----	--------	-------------	-----------	-------------

RF1	Autenticar usuario	Nombre, contraseña	Baja	Media
RF2	Agregar Crecimiento	Nombre de persona, comprobador 1, comprobador 2	Alta	Media
RF3	Eliminar Crecimiento		Alta	Media
RF4	Ver Crecimiento	Avales	Alta	Media
RF5	Buscar Crecimiento			
RF6	Agregar plan trimestral	Fecha, militante, observaciones	Alta	Alta
RF7	Eliminar plan trimestral		Alta	Alta
RF8	Buscar plan trimestral			
RF9	Ver plan trimestral	Nombre de actividad, fecha, tipo de actividad	Alta	Alta
RF10	Agregar actividad	Nombre de actividad, fecha, tipo de actividad	Alta	Alta
RF11	Buscar actividad			
RF12	Eliminar actividad			
RF13	Agregar permisos del sistema	Nombre del rol	Alta	Alta
RF14	Eliminar permisos del sistema		Alta	Alta
RF15	Buscar permisos del sistema			
RF16	Agregar usuario	Nombre, correo electrónico, contraseña, rol	Media	Baja
RF17	Eliminar usuario		Media	Baja
RF18	Buscar usuario			
RF19	Agregar persona	Usuario, nombre, apellidos, carnet de identidad, sexo, dirección, municipio, provincia, país	Media	Alta
RF20	Modificar persona	Nombre, apellidos, carnet de identidad, sexo, dirección,	Media	Alta

		municipio, provincia, país		
RF21	Eliminar persona		Media	Alta
RF22	Ver persona	Usuario, nombre, apellidos, carné de identidad, sexo, dirección, municipio, provincia, país	Media	Alta
RF23	Buscar persona			
RF24	Agregar clasificadores ocupacionales	Nombre del clasificador	Baja	Baja
RF25	Eliminar clasificadores ocupacionales		Baja	Baja
RF26	Buscar clasificadores ocupacionales			
RF27	Agregar tipo de actividad	Nombre de actividad	Media	Media
RF28	Eliminar tipo de actividad		Media	Media
RF29	Buscar tipo de actividad			
RF30	Agregar causa de medida	Nombre de la causa	Alta	Media
RF31	Eliminar causa de medida		Alta	Media
RF32	Buscar causa de medida			
RF33	Agregar Comité de Base	Nombre del Comité de Base, organismo, institución, Secretario General	Media	Media
RF34	Modificar Comité de Base	Nombre del Comité de Base, organismo, institución, Secretario General	Media	Media
RF35	Eliminar Comité de Base		Media	Media
RF36	Ver Comité de Base	Carnet de militante, carnet de identidad, nombre y apellidos	Media	Media

RF37	Buscar Comité de Base			
RF38	Agregar acta	Actividad, visitante del nivel superior, asistencia, total, cuerpo de acta	Alta	Alta
RF39	Modificar acta	Cuerpo del acta	Alta	Alta
RF40	Eliminar acta		Alta	Alta
RF41	Buscar acta			
RF42	Ver acta	Cuerpo del acta	Alta	Alta
RF43	Agregar medida	Medida propuesta, militante, Comité de Base, causa, fecha de aplicada	Baja	Media
RF44	Eliminar medida		Baja	Media
RF45	Buscar medida			
RF46	Agregar ausencia	Militante, causa de la ausencia, actividad, fecha	Baja	Media
RF47	Eliminar ausencia		Baja	Media
RF48	Buscar ausencia			
RF49	Agregar militante	Carnet de militante, persona, Comité de Base, clasificador ocupacional, fecha alta	Alta	Media
RF50	Modificar militante	Carnet de militante, persona, Comité de Base, clasificador ocupacional	Alta	Media
RF51	Eliminar militante		Alta	Media
RF52	Buscar militante			
RF53	Ver militante	Carnet de militante, persona, Comité de Base, clasificador ocupacional, fecha alta	Alta	Media

Tabla 10: Especificación de los requisitos funcionales.

A continuación, son especificados los requisitos no funcionales con que debe cumplir el sistema que se propone, agrupados en las siguientes categorías:

II.6 Requisitos no funcionales

Número	Requisito	Tipo
RNF 1	Las interfaces gráficas implementadas deben concebirse con un ambiente sencillo y de navegación fácil para el usuario.	Usabilidad
RNF 2	El acceso a la aplicación debe estar regido por la autenticación del usuario y estar en correspondencia con los permisos definidos por la organización.	Seguridad
RNF 3	La autenticación del sistema será la primera acción del usuario, el cual deberá proporcionar un usuario único y una contraseña, los cuales serán de uso exclusivo del propio usuario.	
RNF 4	Las diferentes áreas del sistema se encontrarán protegidas contra acceso no autorizado utilizando roles.	
RNF 5	El administrador del sistema como política de seguridad podrá restringir el acceso a las diferentes áreas del sistema a los usuarios.	
RNF 6	El sistema deberá estar disponible las 24 horas del día durante los 365 días del año, salvo sea necesario su mantenimiento o actualización.	
RNF 7	El tiempo de respuesta ante cada acción del usuario debe ser lo más mínimo posible, no puede sobrepasar los 10 segundos, debido a la importancia de los procesos que se realizan.	Rendimiento

RNF 8	Las interfaces del sistema contendrán los datos de forma estructurada, permitiendo la interpretación correcta de la información.	Interfaz de usuario
RNF 9	La entrada incorrecta de datos será mostrada al usuario claramente, a través de mensajes, detallando los campos donde se encuentra el error y mostrando como título el detalle del error.	
RNF 10	Como medio de almacenamiento se utilizará una base de datos soportada sobre el gestor de base de datos PostgreSQL.	Soporte

Tabla 11: Descripción de los requisitos no funcionales.

II.7 Descripción de requisitos por procesos de negocio.

A continuación se presenta la descripción de uno de los requisitos funcionales del sistema. Los otros están en los anexos.

II.7.1 Especificación del Requisito <Gestionar Crecimiento>

Pre condiciones	El usuario debe estar autenticado en el sistema y cuenta con los permisos necesarios para ejecutar esta acción.
Flujo de eventos	
Sección 1	
Flujo básico Agregar Crecimiento	
1. Se muestra la interfaz principal del sistema y se presiona la pestaña “Crecimiento”.	
2. El usuario presiona la opción “Agregar crecimiento”.	
3. Se muestra la interfaz “Agregar crecimiento” para que el usuario inserte los datos: nombre de la persona, comprobador 1, comprobador 2 y avales, y se presiona el botón “Aceptar”.	
4. Se comprueban los datos introducidos.	

5. Si los datos introducidos están correctos se adiciona la información a la interfaz correspondiente.
6. Concluye el requisito.
Postcondiciones
Se muestra la información en la interfaz correspondiente.
Flujos alternativos
Flujo alternativo 3.a Campos vacíos
1. No se introduce el nombre de la persona o algún otro dato.
2. Se presiona el botón de Aceptar.
3. El sistema muestra un mensaje de error “No deben haber campos vacíos”.
4. Volver al paso 3 del flujo básico.
Postcondiciones
N/A
Flujo alternativo 3.b Datos incorrectos
1. Se introduce nombre con números.
Poscondiciones
No se adiciona al sistema.
Sección 2
Flujo básico Ver Crecimiento
1. Se muestra la interfaz principal del sistema y se presiona la pestaña “Crecimiento”.
2. El usuario presiona la opción “Ver crecimiento”.
3. Concluye el requisito.
Postcondiciones
Se muestra la información de la interfaz “Avaes”.
Flujos alternativos
N/A
Sección 3
Flujo básico Eliminar Crecimiento
1. Se muestra la interfaz principal del sistema y se presiona la pestaña “Crecimiento”.

2. El usuario busca la persona que desea eliminar y presiona la opción “Eliminar”.
3. Concluye el requisito.
Postcondiciones
Se elimina la persona de la lista.
Flujos alternativos
Flujo alternativo 2.a La persona no existe
1. El usuario busca la persona que desea eliminar pero no aparece.
Postcondiciones
N/A
Sección 4
Flujo básico Buscar Crecimiento
1. Se muestra la interfaz principal del sistema y se presiona la pestaña “Crecimiento”.
2. El usuario presiona la opción “Buscar crecimiento”.
3. El usuario rellena el campo con alguno de los datos como el nombre de la persona.
4. Concluye el requisito.
Postcondiciones
Se muestran los resultados que coinciden con ese dato.
Flujos alternativos
Flujo alternativo 3.a La persona no existe
1. El usuario rellena el campo con alguno de los datos como el nombre de la persona.
Postcondiciones
Se muestra un mensaje “No se encontró”.

Tabla 12: Especificación de <Gestionar Crecimiento>.

II.8 Arquitectura: Modelo-Vista-Controlador.

Para el desarrollo de la aplicación web se decide el empleo de la arquitectura: Modelo Vista Controlador (MVC).

El Modelo-Vista-Controlador es un patrón de diseño de arquitectura de software que separa la parte lógica de una aplicación de su presentación. (Arquitectura de software: Qué es y qué tipos existen., 2022).

Al emplear este patrón se facilita agregar nuevos tipos de datos según sean requeridos por la aplicación ya que son independientes del funcionamiento de otras capas y facilita el mantenimiento en caso de errores.

Modelo: representa las estructuras de datos. Las clases del modelo de clases contendrán funciones para consultar, insertar y actualizar información de la base de datos.

Vista: es la información presentada al usuario y para su interacción con él. En aplicaciones Web una vista puede ser una página web o una parte de una ella.

Controlador: actúa como intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para generar una página web. Es el responsable de recibir los eventos de entrada desde la vista.

Figura 3: Interfaz de Gestionar Militante.

II.8.1 Modelo de datos.

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

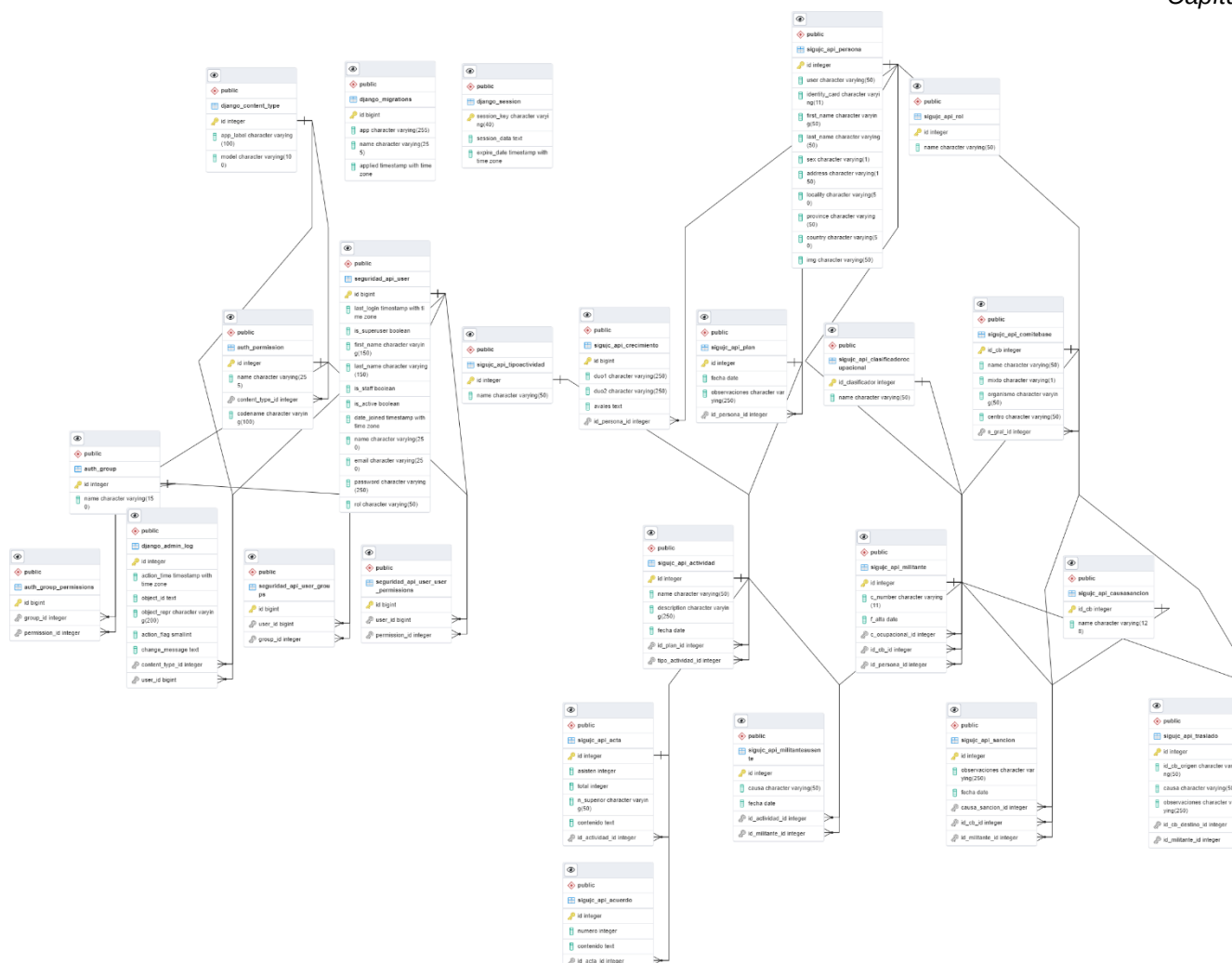


Figura 4: Modelo de datos.

II.8.2 Patrones de diseño.

Un patrón de diseño es una descripción de un problema y su solución, el cual recibe un nombre y puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en circunstancias diversas. Los patrones no se proponen descubrir ni expresar nuevos principios de la ingeniería del software. Al contrario, estos intentan codificar el conocimiento, las expresiones y los principios ya existentes.

Patrones GRASP

Los patrones GRASP (del inglés: *General Responsibility Assignment Software Patterns*) son patrones generales de software para asignación de responsabilidades. Estos describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en

forma de patrones.

Experto: se basa en asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

```
refreshUsersList(){
  this.ListUsuarios = [];
  this.http.get<any[]>({ url: 'http://127.0.0.1:8000/seguridad/users' }).subscribe((item:any)=>{
    item.forEach(user=>{
      this.ListUsuarios.push(user);
    });
  });
}
```

Figura 5: Ejemplo de experto.

Creador: es el encargado de identificar quién debe ser el responsable de la creación o instanciación de nuevos objetos o clases.

```
addPersona(){
  if(this.user_name === '' || this.identity_card === '' ||
    this.first_name === '' || this.last_name === '' || this.sex === '' || this.address === '' ||
    this.locality === '' || this.province === '' || this.country === ''){
    alert("No pueden existir campos vacíos");
  }else {
    if (this.sex === 'Femenino') {
      this.sex = 'F';
    }
    if (this.sex === 'Masculino') {
      this.sex = 'M';
    }
    var persona = {
      identity_card: this.identity_card,
      first_name: this.first_name,
      last_name: this.last_name,
      sex: this.sex,
      address: this.address,
      locality: this.locality,
      province: this.province,
      country: this.country,
      img: this.identity_card,
      user: this.user_name
    };
    console.log(this.sex);
    this.conexion.addMethod(this.action, persona).subscribe(res => {
      alert("Persona agregada correctamente");
    });
    this.user_name = '';
    this.identity_card = '';
    this.first_name = '';
    this.last_name = '';
  }
```

Figura 6: Ejemplo de creador.

Controlador: sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es el controlador quien recibe los datos del usuario y quien los envía a las distintas clases según el método llamado.

```

1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { Observable } from 'rxjs';
4
5 const API_URL = 'http://127.0.0.1:8000/api';
6 const API_SECURITY_URL = 'http://127.0.0.1:8000/seguridad';
7
8 @Injectable({
9   providedIn: 'root'
10 })
11 export class ConexionService {
12
13   constructor(private http: HttpClient) { }
14
15   getMethod(action: any): Observable<any[]>{
16     return this.http.get<any[]>({ url: API_URL + '/' + action + '/' });
17   }
18
19   addMethod(action: any, val: any){
20     return this.http.post({ url: API_URL + '/' + action + '/', val });
21   }
22
23   updateMethod(action: any, val: any){
24     return this.http.put({ url: API_URL + '/' + action + '/', val });
25   }
26
27   deleteMethod(action: any, val: any){
28     return this.http.delete({ url: API_URL + '/' + action + '/' + val });
29   }
30 }

```

Figura 7: Ejemplo de controlador.

Bajo Acoplamiento: se encarga de medir el nivel en que una clase depende o está conectada a otra, basándose en la idea de desligar la relación entre las clases.

```

class MilitanteAusente(models.Model):
    id = models.AutoField(primary_key=True)
    id_militante = models.ForeignKey(Militante, on_delete=models.DO_NOTHING)
    causa = models.CharField(max_length=50)
    id_actividad = models.ForeignKey(Actividad, on_delete=models.DO_NOTHING)
    fecha = models.DateField()

```

Figura 8: Ejemplo de bajo acoplamiento.

Patrones GOF

Los patrones GOF (del inglés: *Gang of Four*) son patrones de diseño software que solucionan problemas de creación de instancias. Se dividen en tres tipos fundamentales, agrupados según el tipo de soluciones que aplican: creacionales, estructurales y de comportamiento.

Contrato Uniforme: Es un patrón utilizado exclusivamente en servicios web, ya que plantea el uso de los cuatro verbos principales del protocolo HTTP en tareas específicas.

```

@csrf_exempt
def PersonaApi(request, id=0):
    if request.method == 'GET':
        element = Persona.objects.all()
        element_serializer = PersonaSerializer(element, many=True)
        return JsonResponse(element_serializer.data, safe=False)

    elif request.method == 'POST':
        element_data = JSONParser().parse(request)
        element_serializer = PersonaSerializer(data=element_data)
        if element_serializer.is_valid():
            element_serializer.save()
            return JsonResponse( data: "Agregado correctamente", safe=False)
        return JsonResponse( data: "Fallo al ejecutar operacion", safe=False)

    elif request.method == 'PUT':
        element_data = JSONParser().parse(request)
        element = Persona.objects.get(id=element_data['id'])
        element_serializer = PersonaSerializer(element, data=element_data)
        if element_serializer.is_valid():
            element_serializer.save()
            return JsonResponse( data: "Agregado correctamente", safe=False)
        return JsonResponse( data: "Fallo al ejecutar operacion", safe=False)

    elif request.method == 'DELETE':
        element = Persona.objects.get(id=id)
        element.delete()

```

Activar Windows

Figura 9: Ejemplo de contrato uniforme.

Patrón observador: define de una a muchas dependencias entre objetos de forma que cuando un objeto cambia de estado, todos sus dependientes son notificados y actualizados de forma automática.

```

editPersona(){
    if(this.sex === 'Femenino') {this.sex = 'F';}
    if(this.sex === 'Masculino'){ this.sex = 'F';}
    var persona = {id: this.personaShow.id, identity_card: this.identity_card, first_name: this.first_name, last_name: this.last_name,
    sex: this.sex, address: this.address, locality: this.locality, province: this.province, country: this.country, img: this.identity_card, user: this.user_name};
    this.conexion.updateMethod(this.action, persona).subscribe(res=>{
        alert("Persona modificada correctamente");
    });
    this.refreshListPersonas();
}
}

```

Figura 10: Ejemplo de patrón observador.

II.9 Diagrama de despliegue.

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Además, muestra cómo se configuran las instancias de los componentes y los procesos para la ejecución en las instancias de los nodos de proceso.



Figura 11: Diagrama de Despliegue.

II.10 Estándares de codificación.

Estándar de código, convención de código, o estilo de programación es un término que describe convenciones para escribir código fuente en cierto lenguaje de programación. Las convenciones de código son importantes para los programadores debido a que el 80% del coste del código de un programa va a su mantenimiento. (Pérez, 2017)

El estándar de codificación utilizado en el desarrollo del sistema se muestra a continuación:

- Los nombres de las clases serán con mayúscula, en caso de ser un nombre compuesto las siguientes palabras se escribirán de igual forma.

```
class MilitanteAusente(models.Model):
    id = models.AutoField(primary_key=True)
    id_militante = models.ForeignKey(Militante, on_delete=models.DO_NOTHING)
    causa = models.CharField(max_length=50)
    id_actividad = models.ForeignKey(Actividad, on_delete=models.DO_NOTHING)
    fecha = models.DateField()
```

Figura 12: Clase MilitanteAusente.

- Los identificadores para las variables y los parámetros serán con letras en minúsculas y en caso de ser un nombre compuesto las siguientes palabras se escribirán con minúscula separadas por un guión bajo.

```

addPersona(){
  if(this.user_name === '' || this.identity_card === '' ||
    this.first_name === '' || this.last_name === '' || this.sex === '' || this.address === '' ||
    this.locality === '' || this.province === '' || this.country === ''){
    alert("No pueden existir campos vacios");
  }else {
    if (this.sex === 'Femenino') {
      this.sex = 'F';
    }
    if (this.sex === 'Masculino') {
      this.sex = 'M';
    }
  }
  var persona = {
    identity_card: this.identity_card,
    first_name: this.first_name,
    last_name: this.last_name,
    sex: this.sex,
    address: this.address,
    locality: this.locality,
    province: this.province,
    country: this.country,
    img: this.identity_card,
    user: this.user_name
  };
  console.log(this.sex);
  this.conexion.addMethod(this.action, persona).subscribe(res => {
    alert("Persona agregada correctamente");
  });
  this.user_name = '';
  this.identity_card = '';
  this.first_name = '';
  this.last_name = '';
}

```

Figura 13: Ejemplo de los atributos de un clase.

- Operadores: Todos los operadores binarios (operadores que se encuentran entre dos valores), como +, -, =, >!, ==, no necesariamente deben tener un espacio antes y después de que el operador.

```

1  import { Injectable } from '@angular/core';
2  import { HttpClient } from '@angular/common/http';
3  import { Observable } from 'rxjs';
4
5  const API_URL = 'http://127.0.0.1:8000/api';
6  const API_SECURITY_URL = 'http://127.0.0.1:8000/seguridad';
7
8  @Injectable({
9    providedIn: 'root'
10  })
11  export class ConexionService {
12
13
14    constructor(private http: HttpClient) { }
15
16
17    getMelthod(action: any):Observable<any[]>{
18      return this.http.get<any[]>({ url: API_URL + '/' + action + '/' });
19    }
20
21    addMelthod(action: any, val: any){
22      return this.http.post({ url: API_URL + '/' + action + '/', val});
23    }
24
25    updateMelthod(action: any, val: any){
26      return this.http.put({ url: API_URL + '/' + action + '/', val});
27    }
28
29    deleteMelthod(action: any, val: any){
30      return this.http.delete({ url: API_URL + '/' + action + '/' + val});
31    }
32
33  }

```

Figura 14: Ejemplo de operadores.

- Estructuras de control: Estructuras de control, que incluyen if, for, while, switch.

```

addUser(){
  if(this.form.getRawValue().name === '' || this.form.getRawValue().email === '' || this.form.getRawValue().password === '' || this.form.getRawValue().rol === ''){
    alert("No pueden existir campos vacios")
  }
  else {
    if (this.form.getRawValue().rol === 'Administrador' && this.checkAdmin()) {
      alert('Solamente puede existir un usuario con rol Administrador');
    }
    else {
      this.http.post({ url: 'http://localhost:8000/seguridad/register', this.form.getRawValue()}).subscribe(res => {
        alert("Usuario agregado correctamente");
      });
      this.form.reset();
      this.refreshUsersList()
    }
  }
}

```

Figura 15: Ejemplo de estructuras de control.

II.11 Conclusiones del capítulo

Al final del capítulo se concluye que:

1. Para el desarrollo del sistema se deben implementar 53 requisitos funcionales y tener en cuenta 10 requisitos no funcionales que se clasifican en usabilidad, soporte, interfaz de usuario y rendimiento.
2. Se documentó el análisis y diseño de los requisitos implementados.
3. Una vez descrito cómo el sistema será realizado a partir de las funcionalidades previstas y las restricciones impuestas (requisitos), se indica con precisión lo que se debe programar.

CAPÍTULO III: Pruebas y validación del sistema SIGUJC.

El proceso de Prueba se realiza en un determinado momento del ciclo de desarrollo de las funcionalidades. Para poner a prueba todos estos componentes desarrollados, es necesario entrar principalmente en la evaluación de la calidad del producto, para lo cual se hace necesario que el sistema funcione como está previsto y que la validación de los requisitos se aplique correctamente.

III.1 Estrategia de prueba.

El **objetivo** que se persigue con esta estrategia es medir el grado en que el sistema de gestión de la información de los procesos cumple con los requisitos del sistema. Para llevarla a cabo, se tuvieron en cuenta diferentes tipos de prueba como pruebas de unidad, pruebas de funcionalidad y pruebas de aceptación.

Prueba de Unidad o unitarias: centra el proceso de verificación en la menor unidad del diseño del software: el componente de software o módulo. (Ruiz, 2010).

Pruebas de Funcionalidad: aseguran que los componentes de un sistema cumplan con las funcionalidades y operen correctamente durante su operación.

Pruebas de aceptación: las realiza el usuario final en lugar del responsable del desarrollo del sistema, una prueba de aceptación puede ir desde un informal paso de prueba hasta la ejecución sistemática de una serie de pruebas bien planificadas. (Ruiz, 2010).

Para la validación de la propuesta de solución, se realizaron las pruebas unitarias usando el método de caja blanca, la cual permite inspeccionar la estructura interna del programa, y la técnica camino básico, que obtiene una medida de la complejidad lógica de un diseño. Además, se efectuaron pruebas funcionales con el método de caja negra teniendo en cuenta el método de particiones equivalentes.

Este método divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. Según Pressman *“El diseño de casos de prueba para la partición*

equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada” (Pressman, Ingeniería del software. Un enfoque práctico. Quinta Edición., 2005). También se realizaron las pruebas de Aceptación por parte del cliente. Además, gracias a las iteraciones se pudieron corregir todos los errores presentes.

III.2 Pruebas funcionales.

III.2.1 Método de Caja Negra.

Las pruebas de caja negra, también denominadas, pruebas de comportamiento, se concentran en los requisitos funcionales del software. Es decir, permiten al ingeniero de software derivar conjuntos de condiciones de entrada que ejercitarán por completo todos los requisitos funcionales de un programa.

Leyenda: V: válido, I: inválido, N/A: no aplica.

Escenario	Descripción	Nombre del rol	Respuesta del sistema	Flujo central
EC 1.1 Introducir correctamente un nombre de rol.	El usuario introduce correctamente el nombre del rol.	V	Se agrega exitosamente al sistema.	-Seleccionar la opción Agregar permiso. -Introducir un nombre de rol correctamente. -Seleccionar la opción Guardar.
		Militante		
EC 1.2 Introducir incorrectamente un nombre de rol.	El usuario introduce incorrectamente el nombre del rol.	I	Muestra el mensaje “Credenciales incorrectas”.	-Seleccionar la opción Agregar permiso. -Introducir un nombre de rol incorrectamente. -Seleccionar la opción Guardar.
		Militante14		
EC 1.3 No introducir un nombre de rol.	El usuario no introduce el nombre del rol.	I	Muestra el mensaje “No pueden haber campos	-Seleccionar la opción Agregar permiso.

		N/A	vacíos”.	-No introducir el nombre del rol. -Seleccionar la opción Guardar.
--	--	-----	----------	--

Tabla 13: Caso de prueba de partición equivalente del RF13 Agregar permisos del sistema.

III.3 Pruebas unitarias.

III.3.1 Método de Caja Blanca.

El método de caja blanca, en ocasiones llamada pruebas de cristal, es un método que usa la estructura de control descrita como parte del diseño al nivel de componentes para derivar los casos de prueba.

Dichas pruebas fueron desarrolladas utilizando la herramienta *Android JUnit*. Luego, se usó la técnica camino básico para obtener una medida de la complejidad de un algoritmo y un conjunto básico de caminos de ejecución de este, utilizados luego para obtener los casos de prueba. (Fuentes, 2019)

```

addMilitante(){
    if( this.cb_name == '' || this.persona_name == '' || this.c_number == '' || this.f_alta == '' || this.c_ocupacional == '' ){
        alert('No pueden existir campos vacíos');
    } else {
        var id_c_ocupacional = this.getIDbyNameClasificador(this.c_ocupacional);
        var id_persona = this.getPersonalIDByName(this.persona_name);
        var id_cb = this.getIDbyName(this.cb_name);
        var militante = {
            c_number: this.c_number,
            f_alta: this.f_alta,
            id_persona: id_persona,
            id_cb: id_cb,
            c_ocupacional: id_c_ocupacional
        };

        this.conexion.addMethod(this.action, militante).subscribe(res => {
            alert("Militante agregado correctamente");
        });

        this.persona_name = '';
        this.cb_name = '';
        this.c_number = '';
        this.f_alta = '';
        this.c_ocupacional = '';
        this.refreshListMilitantes();
    }
}

```

Figura 16: Código de Agregar Militante.

A continuación, se definen los pasos que se realizaron:

1. Confeccionar el grafo de flujo.
2. Calcular la complejidad ciclomática.

3. Determinar un conjunto básico de rutas linealmente independientes.

4. Obtención de casos de pruebas.

Luego de realizar el grafo se calculó la complejidad ciclomática, dando como resultado 5.

Posteriormente, se obtuvieron conjuntos de básicos de rutas independientes, donde cada una es un caso de prueba a realizar, de forma que los datos introducidos provoquen que se visiten las sentencias vinculadas a cada nodo del camino. A continuación, se muestra uno de los casos de pruebas:

Caso de uso	Agregar Militante
Descripción	A partir de la entrada de los datos incorrectos se muestra un mensaje de error.
Camino	1, 2, 3, 8
Entrada	Carnet de Militante: CU-T2345, Persona: Sergio Luis Perez Reyna, Comité de Base: “ ”, Clasificador Ocupacional: Estudiante, Fecha alta: “ ”.
Resultado esperado	Se muestra el mensaje: No deben haber campos vacíos.

Tabla 14: Caso de prueba para agregar militante.

III.4 Resultados de pruebas.

Para evaluar la solución se probó el software íntegramente, prestando gran atención a evaluar las funcionalidades que presenta. La realización de las pruebas permitió establecer un conjunto de No Conformidades (NC) en las diferentes iteraciones.

Iteraciones	No conformidades detectadas			
	Errores de contenido	Funcionalidad	Usabilidad	Total
1	4	7	7	24
2	2	4	2	8
3	0	2	1	3
Prueba final	0	0	0	0

Tabla 15: No conformidades por iteración.

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

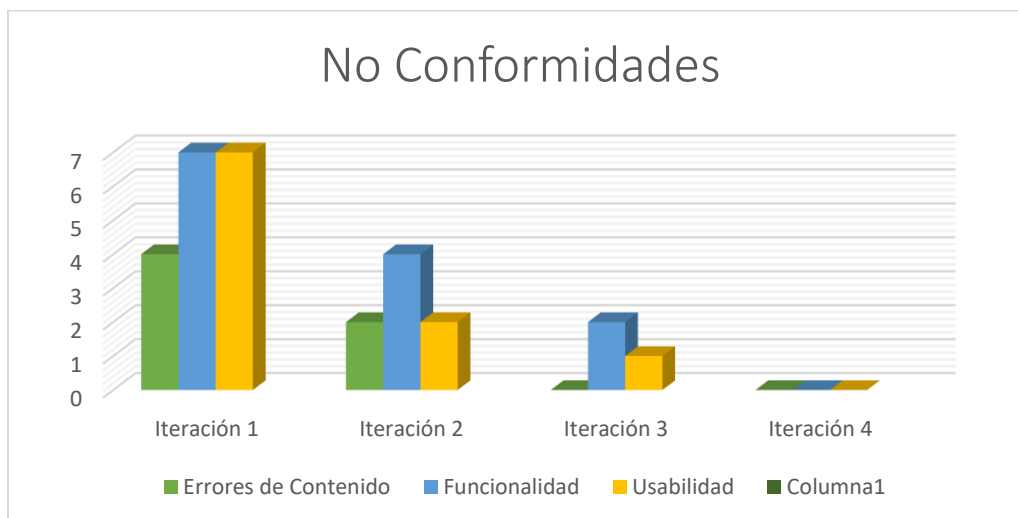


Figura 17: Resultado de pruebas.

III.5 Conclusiones del capítulo

Los artefactos generados en la implementación del sistema, contribuyeron a lograr obtener las siguientes conclusiones:

1. La realización de pruebas de caja blanca y caja negra al sistema obtenido permitió comprobar el correcto funcionamiento de la misma, evaluar las clases diseñadas y descubrir errores una vez implementadas.

Conclusiones finales

Con el desarrollo del sistema SIGUJC en la Facultad 2 se dio respuesta al problema a resolver planteado, obteniendo las siguientes conclusiones:

- Se realizó un estudio detallado que permitió tener un conocimiento de la situación actual y las tendencias de los sistemas de gestión relacionados a la organización, demostrando la necesidad de desarrollar un sistema que fuese capaz de dar solución a los problemas de funcionamiento presentes.
- Al realizar las entrevistas con los clientes se obtuvieron datos relevantes sobre los procesos de negocio así como los requisitos funcionales deseados por este que sirvieron de base para realizar una propuesta de solución basada en las necesidades de los procesos de gestión de la UJC en la Facultad 2.
- Las tecnologías y herramientas definidas posibilitaron la creación de un sistema adaptable a diferentes entornos de despliegue y capaz de gestionar los procesos referentes a la UJC tales como: la gestión de acta, sanciones, planes, militantes y Comités de Base.
- Se aplicaron las pruebas internas, caja negra y caja blanca, utilizando la técnica partición de equivalencia y camino básico; validando tanto la interfaz como el adecuado funcionamiento interno del software.

Recomendaciones

Con el objetivo de mejorar y seguir perfeccionando este sistema, se recomienda la implementación de más funcionalidades que contribuyan a la disponibilidad y control de la información de los procesos de la UJC como:

- Mostrar Comités de Base que incumplen el cronograma de reuniones.
- Mostrar pases al PCC.
- Gestión de Cotización.
- La opción de que el aspirante a militante se auto proponga como candidato.

REFERENCIAS BIBLIOGRÁFICAS

(2014). *Estatutos de la Unión de Jóvenes Comunistas*. Cuba.

Alberto, R. M. (Junio de 2017). API para el desarrollo de aplicaciones clientes de escritorio con el marco de trabajo XEGFORT. La Habana, Cuba.

alexandre. (21 de noviembre de 2022). *PyCharm: Todo sobre el IDE de Python más popular*. Formation Data Science. Obtenido de Todo sobre el IDE de Python más popular. Formation Data Science: <https://datascientest.com/es/pycharm>

Arquitectura de software: Qué es y qué tipos existen. (24 de agosto de 2022). Obtenido de Qué es y qué tipos existen.: <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>

B., G. (26 de septiembre de 2023). *¿Qué es Apache? Descripción completa*. Obtenido de Descripción completa: <https://www.hostinper.es/tutoriales/que-es-apache>

Carlos Montenegro Amador, Y. M. (Junio de 2015). “Sistema Integral para la Gestión de la Información de los Procesos Sustantivos de la UJC en la UCI. SIGPS_UCI”. La Habana, Cuba.

Comunistas, U. d. (2014). *Reglamento de la Unión de Jóvenes Comunistas*. Cuba.

Comunistas, U. d. (2014). *Estatutos de la Unión de Jóvenes Comunistas*. Cuba.

Fuentes, B. S. (2019). Módulo de gamificación basado en ranking para la plataforma educativa ZERA. La Habana, Cuba.

HACK A BOSS. (11 de octubre de 2023). *¿Qué es CSS? Para qué se usa en diseño y programación web*. Obtenido de Para qué se usa en diseño y programación web: <https://www.hackaboss.com/blog/que-es-css>

Inc. Amazon Web Services. (11 de octubre de 2023). *¿Qué es Django? Explicación del software Django*. Obtenido de Explicación del software Django : <https://aws.amazon.com/es/what-is/django/>

Inc. Amazon Web Services. (11 de octubre de 2023). *¿Qué es Python? Explicación del lenguaje Python*. Obtenido de Explicación del lenguaje Python: <https://aws.amazon.com/es/what-is/python/>

Inc. Amazon Web Services. (11 de octubre de 2023). *Angular y Angular CLI*. Obtenido de <https://bahiaxip.com/entrada/angular-y-angular-cli>

Inc. Amazon Web Services. (11 de octubre de 2023). *Funcionalidades—IntelliJ IDEA*. Obtenido de IntelliJ IDEA.: <https://www.jetbrains.com/es-es/idea/features/>

- Inc. Amazon Web Services. (11 de octubre de 2023). *Lenguaje HTML5—HTML en español*. Obtenido de HTML en español: <https://lenguajehtml.com/html/>
- Inc. Amazon Web Services. (11 de octubre de 2023). *Node.js*. Obtenido de <https://nodejs.org/es/about>
- Inc. Amazon Web Services. (6 de febrero de 2023). *Psycopg2 para acceder a una base de datos PostGIS con Phyton*. Obtenido de <https://geoinnova.org/blog-territorio>
- Inc. Amazon Web Services. (11 de octubre de 2023). *Qué es TypeScript*. Obtenido de <https://codigofacilito.com/articulos/typescript>
- Inc. Amazon Web Services. (11 de octubre de 2023). *REST API*. Obtenido de <https://www.redhat.com/es/topics/api/what-is-a-rest-api>
- McDaniel, G. y International Business Machines Corporation. (2015). *IBM Dictionary of Computing*. Obtenido de <https://www-03.ibm.com/ibm/history/documents/pdf/glossary.pdf>
- MDN. (24 de julio de 2023). *JavaScript*. Obtenido de <https://developer.mozilla.org/es/docs/Web/JavaScript>
- Morales, A. (15 de noviembre de 2017). *Descubre el nuevo pgAdmin 4 para trabajar con PostGIS*. Obtenido de <https://mappinggis.com/2017/11/descubre-el-nuevo-pgadmin-4-para-trabajar-con-postgis/>
- Pérez, J. P. (2017). “Capa de servicios web para la gestión de la información generada en un punto de venta”. La Habana, Cuba.
- Platzi. (11 de octubre de 2023). *Descubre PostgreSQL: Qué es, cómo funciona y ventajas*. Obtenido de Qué es, cómo funciona y ventajas.: <https://platzi.com/blog/que-es-postgresql/>
- Pressman, R. S. (2002). *Ingeniería de Software, un enfoque práctico. Quinta edición*.
- Pressman, R. S. (2005). *Ingeniería del software. Un enfoque práctico. Quinta Edición*.
- Ramírez, Y. A. (s.f.). Sistema Integral de Gestión de la UJC Nacional: Módulo Militante. Cuba.
- Ruiz, R. T. (2010). Las Pruebas de Software y su Importancia en las Organizaciones.
- Thais Sánchez Martínez, R. Q. (junio de 2015). Sistema para la Gestión de la Información de la UJC en la Facultad 3. La Habana, Cuba.
- Universidad de Ciencias Informáticas. (2021). *PCC*. Obtenido de <http://pcc.cu>

ANEXOS

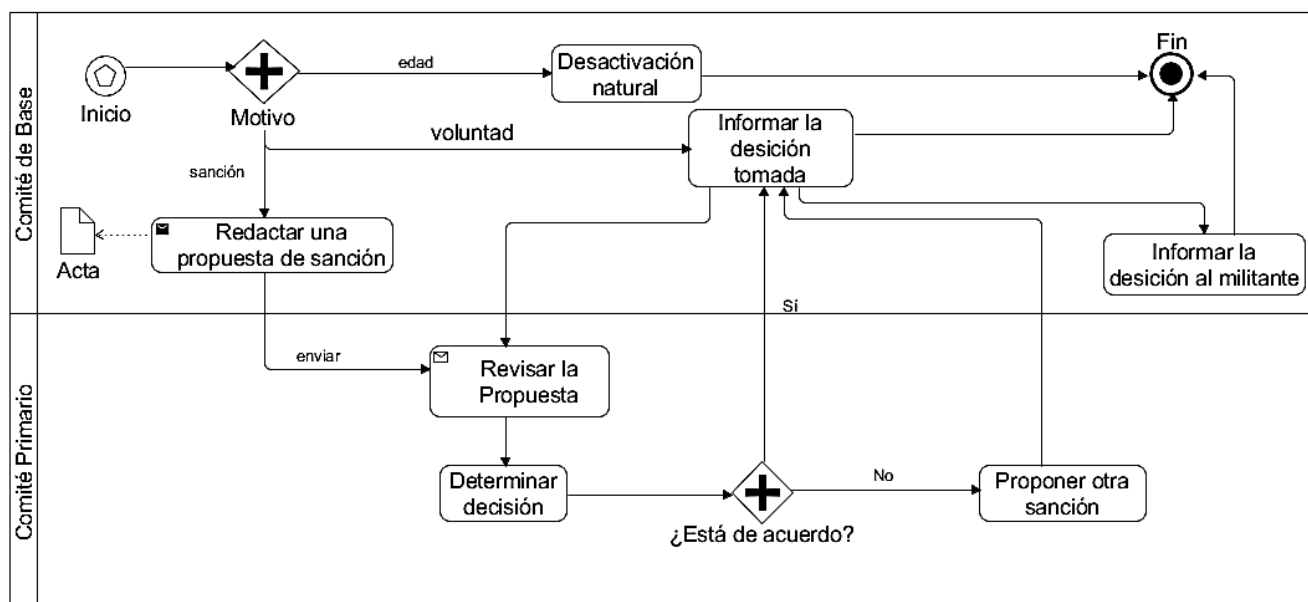


Figura 18: Modelado del proceso de desactivación.

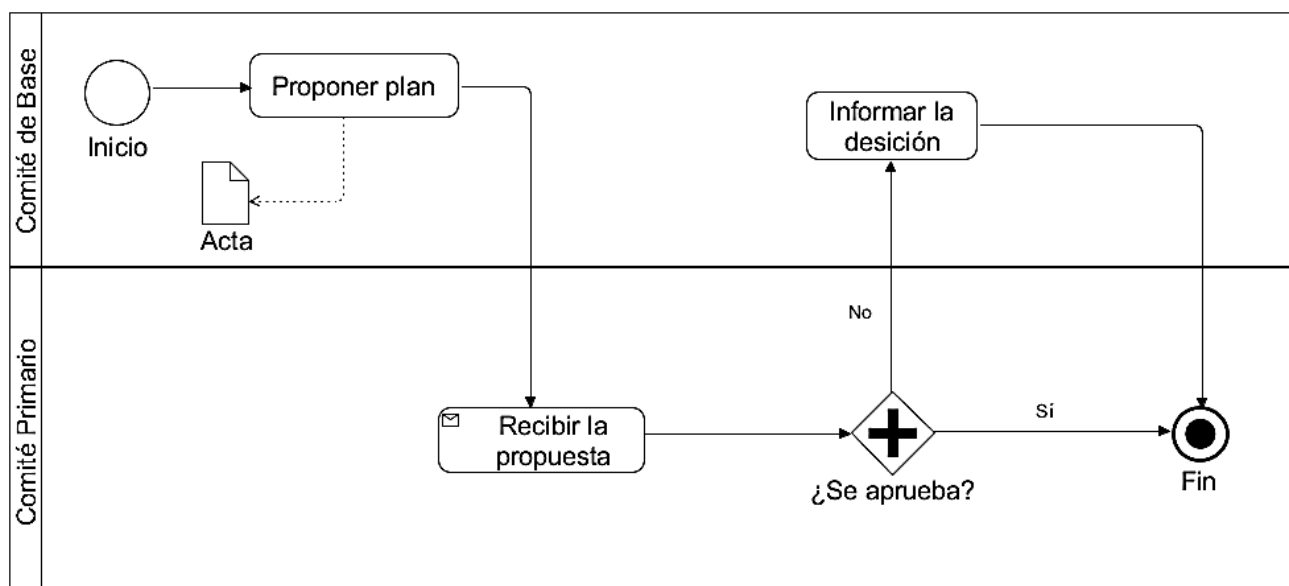


Figura 19: Modelado del proceso elaboración de planes trimestrales.

Objetivo	Elaborar planes trimestrales.
Evento(s) que lo genera(n)	Reunión del Comité de Base.

Pre condiciones	Deben existir los acuerdos pactados en la reunión.
Responsable	Activista de Acta
Clientes internos	Comité de Base
Clientes externos	N/A
Entradas	Datos
Flujo de eventos	
Flujo básico	
1. Se realiza una propuesta de plan trimestral.	En la reunión del Comité de Base se proponen una serie de actividades que comprenden un plan trimestral.
2. Se envía la propuesta al Comité Primario.	El Comité Primario revisa la propuesta.
3. Se encarga de los preparativos.	El Comité Primario aprueba la propuesta y comienza los preparativos.
Postcondiciones	
Se agrega un nuevo plan trimestral.	
Salidas	
Nuevo plan trimestral.	
Flujos alternos	
2a.No se aprueba la propuesta.	
1. Se le informa la decisión al Comité de Base.	
Postcondiciones	
No se realizan las actividades.	
Salidas	
No aplica.	

Tabla 16: Descripción del proceso Elaboración de Planes Trimestrales.

Objetivo	Realizar desactivación de un militante.
Evento(s) que lo genera(n)	El militante ha sido sancionado. El militante ha llegado a los 32 años de edad. El militante decide abandonar la organización.
Pre condiciones	Debe existir el militante al cual se le realizará la desactivación.
Responsable	Secretario General

Cientes internos	Comité de Base
Cientes externos	N/A
Entradas	Datos
Flujo de eventos	
Flujo básico	
1. Se redacta documento del motivo de la desactivación.	El militante le comenta a su Comité de Base que quiere abandonar la organización. El militante ha llegado a los 32 años de edad. El militante ha cometido una indisciplina y el Comité de Base propone una sanción.
2. Se envía la propuesta al Comité Primario.	El Comité Primario revisa la propuesta.
3. Aprueba la propuesta.	El Comité Primario aprueba la propuesta y se lo notifica al Comité de Base.
Postcondiciones	
Se expulsa al militante de la organización.	
Salidas	
El militante abandona la UJC.	
Flujos alternos	
2a.No se aprueba la propuesta.	
1. El Comité Primario propone otra medida.	
Postcondiciones	
El Comité de Base acata la orden.	
Salidas	
El militante no se desactiva.	

Tabla 17: Descripción del proceso Desactivación.

Pre condiciones	El usuario debe estar autenticado en el sistema y cuenta con los permisos necesarios para ejecutar esta acción.
Flujo de eventos	
Sección 1	
Flujo básico Agregar plan trimestral	

1. Se muestra la interfaz principal del sistema y se presiona la pestaña “Planes Trimestrales”.
2. El usuario presiona la opción “Agregar plan trimestral”.
3. Se muestra la interfaz “Agregar trimestral” para que el usuario inserte los datos: fecha, militante y observaciones, y se presiona el botón “Aceptar”.
4. Se comprueban los datos introducidos.
5. Si los datos introducidos están correctos se adiciona la información a la interfaz correspondiente.
6. Concluye el requisito.
Postcondiciones
Se muestra la información en la interfaz correspondiente.
Flujos alternativos
Flujo alternativo 3.a Campos vacíos
1. No se introduce la fecha o algún otro dato.
2. Se presiona el botón de Aceptar.
3. El sistema muestra un mensaje de error “No deben haber campos vacíos”.
4. Volver al paso 3 del flujo básico.
Postcondiciones
1. N/A
Sección 2
Flujo básico Ver Plan Trimestral
1. Se muestra la interfaz principal del sistema y se presiona la pestaña “Planes Trimestrales”.
2. El usuario presiona la opción “Ver plan trimestral”.
3. Concluye el requisito.
Postcondiciones
Se muestra la información de la interfaz “Relación de actividades”.
Flujos alternativos

N/A
Sección 3
Flujo básico Eliminar Plan Trimestral
1. Se muestra la interfaz principal del sistema y se presiona la pestaña “Planes Trimestrales”.
2. El usuario busca el plan que desea eliminar y presiona la opción “Eliminar”.
3. Concluye el requisito.
Postcondiciones
1. Se elimina el plan de la lista.
Flujos alternativos
Flujo alternativo 2.a El plan no existe
1. El usuario busca el plan que desea eliminar pero no aparece.
Postcondiciones
N/A
Sección 4
Flujo básico Buscar Plan Trimestral
1. Se muestra la interfaz principal del sistema y se presiona la pestaña “Planes Trimestrales”.
2. El usuario presiona la opción “Buscar plan trimestral”.
3. El usuario rellena el campo con alguno de los datos como el militante a cargo.
4. Concluye el requisito.
Postcondiciones
Se muestran los resultados que coinciden con ese dato.
Flujos alternativos
Flujo alternativo 3.a El plan no existe
1. El usuario rellena el campo con alguno de los datos como el militante a cargo.
Postcondiciones
Se muestra un mensaje “No se encontró”.

Tabla 18: Especificación de <Gestionar Plan Trimestral>.

Pre condiciones	El usuario debe estar autenticado en el sistema y cuenta con los permisos necesarios para ejecutar esta acción.
Flujo de eventos	
Sección 1	
Flujo básico Agregar tipo de actividad	
1. Se muestra la interfaz principal del sistema y se presiona la pestaña “Tipos de Actividades”.	
2. El usuario presiona la opción “Agregar tipo de actividad”.	
3. Se muestra la interfaz “Agregar tipo de actividad” para que el usuario inserte los datos: nombre de la, y se presiona el botón “Aceptar”.	
4. Se comprueban los datos introducidos.	
5. Si los datos introducidos están correctos se adiciona la información a la interfaz correspondiente.	
6. Concluye el requisito.	
Postcondiciones	
Se muestra la información en la interfaz correspondiente.	
Flujos alternativos	
Flujo alternativo 3.a Campos vacíos	
5. No se introduce el nombre.	
6. Se presiona el botón de Aceptar.	
7. El sistema muestra un mensaje de error “No deben haber campos vacíos”.	
8. Volver al paso 3 del flujo básico.	
Postcondiciones	
2. N/A	
Flujo alternativo 3.b Datos incorrectos	
1. Se introduce el nombre con números.	
Poscondiciones	

No se adiciona al sistema.
Sección 2
Flujo básico Eliminar tipo de actividad
4. Se muestra la interfaz principal del sistema y se presiona la pestaña “Tipos de Actividades”.
5. El usuario busca el tipo de actividad que desea eliminar y presiona la opción “Eliminar”.
6. Concluye el requisito.
Postcondiciones
2. Se elimina el tipo de actividad de la lista.
Flujos alternativos
Flujo alternativo 2.a El tipo de actividad no existe
2. El usuario busca el tipo de actividad que desea eliminar pero no aparece.
Postcondiciones
N/A
Sección 3
Flujo básico Buscar tipo de actividad
5. Se muestra la interfaz principal del sistema y se presiona la pestaña “Tipos de Actividades”.
6. El usuario presiona la opción “Buscar tipo de actividad”.
7. El usuario rellena el campo con alguno de los datos.
8. Concluye el requisito.
Postcondiciones
Se muestran los resultados que coinciden con ese dato.
Flujos alternativos
Flujo alternativo 3.a El tipo de actividad no existe
2. El usuario rellena el campo con alguno de los datos.
Postcondiciones
Se muestra un mensaje “No se encontró”.

Tabla 19: Especificación de <Gestionar Tipo de Actividad>.