



**Universidad de las Ciencias Informáticas
Facultad 1**

**Módulo de funcionamiento para el Sistema de Control de
Militantes del Comité Central del PCC**

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

Autor: Yenisley Porta Rodríguez

Tutor: Ing. Osay González Fuentes

La Habana, Cuba
junio 2019

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo: Yenisley Porta Rodríguez, con carné de identidad 96100311775, soy la autora principal del trabajo de diploma que se titula: Módulo de funcionamiento para el Sistema de Control de Militantes del Comité Central del PCC, el cual ha sido desarrollado como parte del trabajo en el centro CISED de la Facultad 1. Autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, se firma la presente declaración jurada de autoría en La Habana, a los _____ días del mes de _____ del año 2019.

Tutor

Ing. Osay González Fuentes

Autor

Yenisley Porta Rodríguez

Datos de contacto

Tutor: Ing. Osay González Fuentes

Ciudadanía: cubana.

Institución: Universidad de la Ciencias Informáticas Centro de Identificación y Seguridad Digital

Títulos: Ingeniera en Ciencias Informáticas.

E-mail: ogonzalezf@uci.cu

Dedicatoria

A mis padres y mi hermano pequeño porque quiero ser su ejemplo a seguir.

Agradecimientos

Yenisley:

A la UCI por darme la oportunidad de conocer a las personas más influyentes en mi preparación.

A mis mejores amigos Duriet, Stephanie, Rafa, Melissa, Yulina, Armando, Enier, Lio, David, Laura, Emilio, Adam, Johan, Ivan, Alejandro, Bryan, Anthuan.

Al gran amor de mi vida Deniel, que ha sido mi compañero en muchas tareas como lo es este logro, por acompañarme siempre en mis mejores momentos y en los malos, por aguantarme que sé que es una tarea difícil, por cuidarme y protegerme, por ser mi amigo en momentos que necesito un abrazo de un amigo y no de un novio. Gracias mi amor por estar a mi lado. Te amo.

A mi familia que son mi razón ser, el motor impulsor para hacer posible este sueño, mi mamá, mi papá, mis hermanos, mi abuelo, mi abuela, aunque hoy no está aquí comenzó este sueño conmigo, mi tío, a Idálmis, a mi mejor amiga Stephanie que es parte de mi familia.

Resumen

El objetivo principal de esta investigación es desarrollar un módulo de funcionamiento para el SICOM, que permita gestionar la información de las reuniones de los núcleos del PCC del país. Surge ante la necesidad de contribuir al proceso de la gestión de las reuniones de los núcleos del partido. Actualmente para realizar este proceso no se cuenta con un sistema informático que permita desarrollar todos los procesos que se desarrollan dentro del funcionamiento del PCC, como son: la gestión de las reuniones, la elaboración de los reportes y la presentación de los temas que se van a tratar en la reunión, esto provoca que se vuelva muy engorroso elaborar informes a nivel central.

El módulo fue desarrollado bajo tecnologías libres, las cuales fueron seleccionadas teniendo en cuenta la integración con el sistema SICOM. Como entorno de desarrollo *Django* en su versión 1.11, como lenguaje de programación *python* en su versión 3.5, como entorno de desarrollo integrado (IDE) el *PyCharm* en su versión 2017.2.2, como gestor de base de datos *PostgreSQL* en su versión 9.4 y para el modelado *Visual Paradigm* en su versión 8.0.

Palabras clave: módulo, información, funcionamiento, reunión, actas.

Índice de contenidos

Introducción	1
Capítulo #1. Fundamentación teórica del módulo de funcionamiento.....	- 6 -
1.1 Conceptos asociados al dominio de la investigación (Módulo de funcionamiento)	- 6 -
1.1.1 Gestión de la Información	- 6 -
1.1.2 Captura de la Información	- 7 -
1.1.3 Procesamiento de la Información	- 8 -
1.1.4 Visualización de la Información	- 8 -
1.1.5 Gestión de reunión	- 9 -
1.2 Análisis de soluciones existentes.....	- 9 -
1.2.1 Análisis de los sistemas homólogos.....	- 14 -
1.3 Entorno de desarrollo, lenguaje de programación y herramientas	- 15 -
1.3.1 Herramientas	- 16 -
1.3.2 Herramientas CASE	- 16 -
1.4 Metodología	- 17 -
1.5 Conclusiones del capítulo 1	- 19 -
Capítulo #2. Descripción del módulo de funcionamiento	- 20 -
2.1 Modelo Conceptual.	- 20 -
2.2 Especificación de Requisitos	- 21 -
2.2.1 Requisitos Funcionales	- 22 -
2.2.2 Requisitos no funcionales.....	- 23 -
2.3 Definición de los actores del Sistema.....	- 24 -
2.3.1 Diagrama de casos de uso	- 25 -

2.4 Arquitectura MVT.....	- 30 -
2.5 Diagrama de clase de diseño.....	- 31 -
2.6 Patrones de Diseño	- 32 -
2.7 Conclusiones Parciales	- 36 -
Capítulo #3. Implementación y prueba del módulo de funcionamiento.....	- 37 -
3.1. Introducción.....	- 37 -
3.2. Diagrama de componente.....	- 37 -
3.3. Diagrama de despliegue	- 38 -
3.4. Estándares de codificación	- 39 -
3.5 Pruebas de Software	- 41 -
3.5.1 Pruebas de caja negra	- 41 -
3.5.2 Pruebas de caja blanca	- 45 -
3.5.3 Pruebas de Integración	- 47 -
3.5.4 Pruebas de aceptación	- 48 -
3.6 Conclusiones del capítulo	- 49 -
Conclusiones Generales	- 50 -
Recomendaciones	- 51 -
Bibliografía referenciada	- 52 -
Anexos.....	- 55 -

Índice de Ilustraciones

<i>Ilustración 1: Sistema GOC.....</i>	- 11 -
<i>Ilustración 2: logo del sistema actius.....</i>	- 13 -
<i>Ilustración 3: Modelo conceptual (elaboración propia)</i>	- 20 -
<i>Ilustración 4: Diagrama de caso de uso.....</i>	- 25 -
<i>Ilustración 5: Arquitectura MVT (Elaboración Propia).....</i>	- 30 -
<i>Ilustración 6: Arquitectura MVT (código).....</i>	- 31 -
<i>Ilustración 7: Diagrama de clase de diseño.....</i>	- 32 -
<i>Ilustración 8: Representación de patrones (alta cohesión, bajo acoplamiento y controlador)</i>	- 34 -
<i>Ilustración 9: Representación del patrón experto</i>	- 35 -
<i>Ilustración 10: Login</i>	- 36 -
<i>Ilustración 11: Diagrama de componente</i>	- 38 -
<i>Ilustración 12: Diagrama de despliegue (Diagrama de despliegue).....</i>	- 39 -
<i>Ilustración 13: Grafo de flujo</i>	- 45 -
<i>Ilustración 14: Prueba de integración</i>	- 48 -

Índice de tablas

<i>Tabla 1: Requisitos Funcionales</i>	<i>- 22 -</i>
<i>Tabla 2: Actores del sistema.....</i>	<i>- 24 -</i>
<i>Tabla 3: Descripción del RF Gestionar Reunión</i>	<i>- 25 -</i>
<i>Tabla 4: Diseño del caso de prueba Gestionar reunión</i>	<i>- 42 -</i>
<i>Tabla 5: Resultados de las pruebas de caja negra.....</i>	<i>- 44 -</i>
<i>Tabla 6: Resultados estadísticos de la prueba de aceptación</i>	<i>- 49 -</i>

Introducción

El Partido Comunista de Cuba (PCC), marxista-leninista, vanguardia organizada de la nación cubana, es la fuerza dirigente superior de la Sociedad y el Estado, que organiza y orienta los esfuerzos comunes hacia los altos fines de la construcción del socialismo y el avance hacia la sociedad comunista. El Partido Comunista de Cuba es una organización de unidad revolucionaria y su objetivo final es la construcción del comunismo, expresado en sus estatutos y reglamentos. De acuerdo con dichos estatutos, el ingreso al PCC es voluntario. Se realiza mediante las asambleas de elección de trabajadores ejemplares y por selección individual. La voluntad del Partido es fortalecer e incrementar las medidas para la prevención y el combate contra las manifestaciones de corrupción, indisciplina, ilegalidades y otras conductas negativas, que ponen en peligro a la Revolución (CubaEduca, 2019).

Como parte de las tareas del PCC para resolver los problemas internos de la organización y consolidar una nueva moral en la sociedad cubana, cimentada en la ideología de la Revolución, la solidaridad, la igualdad y la justicia social, se realizan las reuniones por núcleo, que pueden ser ordinaria y extraordinarias. La gestión de dichas reuniones que incluye la gestión de los temas y los reportes son procesos asociados al funcionamiento del partido. Estas reuniones son realmente necesarias pues tienen el objetivo de que cada militante plantee sus inquietudes. Además, demuestra la justeza de la línea política del PCC, crea el vínculo con el pueblo, escucha e incorpora a la mayoría a la lucha por los objetivos de la Revolución.

Dichas reuniones son preparadas con 72 horas de antelación con vista a que todos sus militantes estén suficientemente informados. La dirección del núcleo es la encargada de elaborar el proyecto del orden del día de las reuniones. En ella se discuten los aspectos más importantes que tienen relación con el cumplimiento de los planes de producción, servicios, docencia, o la actividad específica del centro o la comunidad en cuestión; la situación política e ideológica del colectivo de trabajadores o la población, sus condiciones de trabajo y de vida; la actividad interna del núcleo; la labor del comité de base de la Unión de Jóvenes Comunistas, la sección sindical y demás organizaciones. En el transcurso de la discusión de dichos aspectos, se hacen intervenciones y se toman acuerdos para dar solución a los problemas que puedan existir.

Todos estos acuerdos deben quedar recogidos en el acta de manera precisa y clara, indicando los responsables de su cumplimiento y los plazos fijados para ello. Actualmente la confección de esta acta se realiza de forma manual pues no existe un sistema capaz de realizar dicho proceso. Sin embargo, el Comité Central del PCC (CCPCC) pretende mejorar, mediante la informatización, la manera en que se realiza el llenado y revisión de dichas actas, pues como plantea el presidente del Consejo de Estado y de Ministros, Miguel Díaz Canel Bermúdez, (...) “la informatización es una de las tareas que la población más agradecerá, porque ahorrará trámites, tiempo, intermediarios y costos. Se supone, que seamos más eficientes, que haya más productividad, más rendimiento y, por tanto, menos gente dedicada al papeleo” (Hernández, 2018). Con el fin de ampliar la digitalización de sus procesos, el CCPCC acude a la Universidad de Ciencias Informáticas (UCI).

Esta universidad surge en el año 2002 con la visión del futuro de nuestro comandante Fidel Castro, su objetivo principal es formar profesionales comprometidos con su Patria, altamente calificados en la rama de la informática, y producir aplicaciones y servicios informáticos, a partir del vínculo docencia - investigación - producción como modelo de formación, sirviendo de soporte a la industria cubana del software. (Berenguer, 2018)

En la actualidad en el Centro de Identificación y Seguridad Digital (CISED) de la UCI se desarrolla un sistema informático para el control de la militancia (SICOM) el cual lleva la gestión y control de los militantes del PCC, dicho sistema cuenta con un conjunto de funcionalidades como son el registro de los militantes, las sanciones según la indisciplina cometida, el proceso de avalados entre otros.

Sin embargo, el actual SICOM no contempla el proceso de llenado y revisión de las actas de las reuniones de los núcleos del PCC, por lo que este proceso se realiza de forma manual, lo cual provoca las siguientes deficiencias, no se cumple con el tiempo establecido (48 horas) para la entrega del acta de las reuniones a un nivel superior, lo que provoca deficiencias en la inmediatez de la información, además no existe retroalimentación de las indicaciones lo que afecta el proceso de análisis para el perfeccionamiento del sistema. Los aspectos que se recogen en el acta son: el control de los presentes, ausentes, causas de ausencias, visitantes a la reunión, intervenciones, acuerdos y principales temas tratados. Durante el llenado del acta, los secretarios de acta no suelen cumplir con el formato establecido, por lo que no se recoge de manera homogénea el acta de cada núcleo, lo que trae consigo dificultades para la confección de los posteriores informes de control de la militancia o el chequeo de acuerdos.

Con respecto a los acuerdos, estos deben estar enumerados de manera tal que los números de la reunión de cada mes sean consecutivos con respecto a los de la reunión del mes anterior, es decir, si el último acuerdo de la reunión del mes anterior fue el número 28, el primero de los acuerdos del mes actual debe ser el 29. Los secretarios de acta no suelen respetar esta norma de numeración. También, suelen omitir los responsables y la fecha de cumplimiento de los acuerdos.

Por otra parte, una vez que el acta es enviada a una instancia superior, los funcionarios deben revisar la cantidad de ausentes que hubo por núcleo y recoger los datos estadísticos manualmente de todos los núcleos, esto es un proceso que provoca desgaste de la fuerza de trabajo ya que son demasiadas las actas que llegan a la provincia o a la nación. Otro proceso que aún se realiza de forma manual es el reporte que se realiza al final de cada reunión donde se tienen en cuenta los temas que se trataron, los acuerdos, los ausentes a la reunión, causas de ausencia y el tema más tratado en el mes lo que en ocasiones puede afectar el intercambio de información por parte de los militantes y el comité primario de partido, ya que en ocasiones estos reportes no cumplen con el formato establecido ni con el tiempo de entrega.

A partir de la situación problemática descrita anteriormente surge el siguiente **problema de la investigación**: ¿Cómo contribuir a la gestión de la información obtenida durante las reuniones de un núcleo del Partido Comunista de Cuba?

El **objeto de estudio** de la investigación se centra en el proceso de gestión de la información.

La investigación se enmarca en el **campo de acción**: Proceso de gestión de reuniones como parte del funcionamiento del Partido Comunista de Cuba.

Para dar solución al problema planteado se traza como **objetivo general**: Desarrollar el módulo de funcionamiento para el Sistema de Control de la Militancia que permita gestionar la información de las reuniones de los núcleos del Partido Comunista de Cuba.

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

- Determinar los referentes bibliográficos asociados al desarrollo de sistemas de gestión de información para la elaboración del marco teórico metodológico.

- Identificar las actividades que puedan ser desarrolladas como funcionalidades del sistema para su implementación.
- Diseñar el módulo de funcionamiento para el SICOM.
- Implementar el módulo de funcionamiento para el SICOM.
- Validar las funcionalidades del módulo de funcionamiento para el SICOM.

Después de haber definido los objetivos específicos se tiene como:

Preguntas científicas.

¿Cómo investigar los referentes teóricos para el módulo de funcionamiento asociado al sistema SICOM?

¿Cómo identificar las funcionalidades del sistema para su implementación?

¿Cómo diseñar el módulo de funcionamiento para el SICOM?

¿Cómo llevar a cabo el proceso de implementación del módulo de funcionamiento?

¿Qué resultados se obtendrían durante la validación del módulo de funcionamiento?

¿Cómo gestionar la información de las reuniones de los núcleos del partido?

Los **métodos de investigación** utilizados para dar solución a la presente investigación son:

Teóricos:

- **Analítico-Sintético:** permitió llevar a cabo un análisis bibliográfico de todo lo referente al módulo de funcionamiento y así de esta forma obtener conocimiento para luego sintetizarlo.
- **Modelación:** facilitó la representación mediante diagramas de las características, procesos y componentes del módulo de funcionamiento, así como la relación existente entre ellos.

Empíricos:

- **Entrevista:** se realizó a través de encuentros con el cliente para conocer todos los requisitos funcionales y poder satisfacer sus necesidades. (Anexo 1)

La presente investigación está estructurada de la siguiente forma:

Capítulo 1: Fundamentación Teórica del módulo de funcionamiento.

En este capítulo se realiza un estudio de los referentes teórico-metodológicos asociados al módulo de funcionamiento. Se hace un análisis de todos los procesos internos del PCC. Se seleccionan la metodología, herramientas y tecnologías a utilizar en el desarrollo del módulo de funcionamiento.

Capítulo 2: Descripción del módulo de funcionamiento

En este capítulo se documenta todo el proceso de análisis y diseño del sistema de manera más detallada. Se describen los requisitos de acuerdo a lo establecido en la metodología utilizada, diagrama de clase, diagrama de despliegue, así como los patrones de diseño utilizados.

Capítulo 3: Implementación y validación del sistema.

En este capítulo se detalla la propuesta de solución del módulo de funcionamiento al problema planteado. Se describe la organización del sistema en un diagrama de componentes y se especifican los estándares de codificación a utilizar. Se realizan las pruebas definidas para el módulo garantizando su correcto funcionamiento.

Capítulo #1. Fundamentación teórica del módulo de funcionamiento

Con el objetivo de ampliar más sobre el tema en cuestión, se exponen en el presente capítulo todos los conceptos asociados a la investigación, así como también se establece una descripción de los diferentes sistemas homólogos. Se caracteriza la metodología y la plataforma a utilizar en el desarrollo del módulo.

1.1 Conceptos asociados al dominio de la investigación (Módulo de funcionamiento)

Con el propósito de entender los temas que se van a tratar en este capítulo se exponen a continuación una serie de conceptos asociados al objeto de la investigación y que tienen relación directa con el problema de la investigación.

1.1.1 Gestión de la Información

Según (Martínez, 2018) en términos generales cuando se piensa en una estrategia de codificación los autores suelen referirse a gestionar la “información” en sus aspectos más tradicionales, es decir capturar “el conocimiento”, “codificarlo” y almacenarlo en bases de datos que posibiliten su posterior acceso. Este tipo de “gestión de la información” debería ser denominada más exactamente “gestión tecnológica de la información”.

Según (Cruz, 2016), la gestión de información, es el proceso mediante el cual se obtienen, despliegan o utilizan recursos básicos (económicos, físicos, humanos, materiales) para manejar información para la sociedad a la que sirve. Tiene como elemento básico, la gestión del ciclo de vida de este recurso y se desarrolla en cualquier organización.

Según (Arévalo, 2017), la finalidad de la Gestión de la información es ofrecer mecanismos que permitan a la organización adquirir, producir y transmitir, al menor coste posible, datos e informaciones con una calidad, exactitud y actualidad suficientes para servir a los objetivos de la organización. Sería, conseguir la información adecuada, para la persona que lo necesita, en el momento que lo necesita, al mejor precio posible para tomar la mejor de las decisiones.

Luego de haber analizado los conceptos sobre gestión de información se asume en la presente investigación la definición de (Arévalo, 2017), ya que teniendo en cuenta el objetivo del módulo este recoge todos los aspectos que se quieren coseguir con el manejo de la gestión de información, desde el

mejoramiento de los canales de comunicación y acceso a la información, hasta determinar las necesidades de información en correspondencia a sus funciones y actividades.

1.1.2 Captura de la Información

(Rafael Montero, 2016), plantean que para la captura de la información se debe desarrollar programas de cómputo que faciliten la tarea de la captura, los cuales deberán ser previamente probados y ajustados, además, deben permitir generar los tabulados de salida de cada una de las variables y cruces de variables considerados en el estudio.

Según (SoftDoit, 2017), define la captura de la información como eficacia y eficiencia en la extracción de la información que necesitan los documentos en papel y documentos electrónicos, formularios en papel o electrónicos, correos electrónicos, bases de datos y aplicaciones de terceros.

La captura de datos conlleva muchos beneficios, como son:

- **Se puede detectar lo visto anteriormente:** cuando optamos por la digitalización y la captura de datos podemos conseguir, al digitalizar los diferentes documentos y capturar los datos más relevantes, ver elementos o cosas que se nos hayan escapado en otra situación por no tener todos los documentos juntos.
- **Enriquecimiento de las bases de datos:** la digitalización de los documentos permite mejorar los datos que se tienen en la base de datos. No es lo mismo tener dos documentos en un archivo que tenerlos digitalizados, pues se puede mejorar las bases de datos y navegar por ellos.
- **Permiten comparar:** al poder navegar por todos los datos de los documentos que se han digitalizado, se puede comparar mejor las diferentes informaciones que consideramos relevantes, como pueden ser porcentajes o términos que se repiten.
- **Permiten almacenar mejor:** al capturar los datos y digitalizarlos se tiene toda esa información mejor almacenada, protegida y estructurada que si estuviesen en diferentes soportes físicos que podrían extraviarse, quemarse o ser dañados de forma más fácil.

- **Permiten compartirlo mejor:** al disponer de toda la información en formato digital y haber capturado todos los datos, se puede compartir de forma más fácil en ficheros, emails y de otras formas que no sería posible antes.
- **Permiten relacionarlos:** además, se puede relacionar un dato o un fichero concreto con otro; por ejemplo, unificar informes y fotografías al vincular uno a la otra.
- **Se pueden añadir metadatos:** esta es otra de las ventajas de haber digitalizado los datos que se capturan, ya que se puede enriquecer el archivo introduciendo usted mismo, metadatos que juzguemos relevante tener.
- **No es caro:** la digitalización y la captura de datos es algo realmente barato para la mayoría de las empresas y no supone un gran esfuerzo económico.

Se asume el concepto de (SoftDoit, 2017) ya que es más amplio, y recoge de forma clara aspectos por lo que se hace importante la captura de la información.

1.1.3 Procesamiento de la Información

Según (Instituto Nacional de Estadística , 2017), el procesamiento de la información se define como la serie de actividades mediante las cuales se ordenan, almacenan y preparan los archivos con la información captada, asegurando su congruencia con el fin de proceder a su explotación para la presentación de resultados estadísticos.

1.1.4 Visualización de la Información

Según (Pérez-Montoro, 2018) la Visualización de la Información es una disciplina transversal que se encarga de la representación visual de contenidos proposicionales mediante el uso de diagramas, gráficas y esquemas para facilitar el almacenamiento, la aprehensión, la interpretación, la transformación y la comunicación de esos contenidos a través de esas representaciones visuales.

El término de "Visualización de la información", se refiere a la forma o manera de representar la información, con el objetivo de facilitar la comprensión y que sea visible e inteligible en el menor tiempo posible. Cuando se presenta la información se obtiene de un repositorio de datos (Archivos o BDs) y se presenta en pantalla en forma organizada, evitando sobrecarga (grandes cantidades de información desorientada) (Ávila, 2018).

Se asume el concepto de Jaime Ávila pues se enfoca en términos informáticos que guarda relación con la investigación, mientras que el concepto de Dr. Mario Pérez-Montoro se va más a un nivel estratégico.

1.1.5 Gestión de reunión

Según (Kausen, 2015), la gestión de reuniones es el conjunto de mecanismos de los cuales dispone el líder o conductor de una reunión con el fin de garantizar una dinámica productiva y eficaz. Las habilidades más importantes son la administración de los tiempos, el reconocimiento y reconducción del tono de la reunión, la escucha reflexiva y la forma de generar pensamiento grupal.

Los principales factores que tiene que definir el líder de la reunión son:

1. Listado de asistentes: Según los tipos de reunión y el tema, hay que definir con precisión el listado de asistentes.
2. Convocatoria: Conviene enviarla por escrito con la máxima antelación. Debe incluir la siguiente información:
 - ✓ Data, hora y lugar.
 - ✓ Duración: Es recomendable establecer pausas después de cada
 - ✓ Hora de reunión porque está demostrado que la atención decae.
 - ✓ Listado de asistentes.
 - ✓ Propósito/Orden del Día.

Actas:

Son una buena manera de hacer explícitas las decisiones del grupo y evitar ambigüedades, solapamientos de funciones o malentendidos en la distribución de tareas. Tienen que ser sintéticas e informativas. Se pueden enviar por e-mail a todos los asistentes o entregarse al principio de la siguiente reunión. Si hay tareas asignadas, el acta debe recoger claramente quién tiene que hacer qué y en qué plazo.

1.2 Análisis de soluciones existentes

En una sociedad envuelta en las tecnologías de la información y comunicación, cada vez se hace más importante la necesidad de utilizar herramientas de software que permitan al usuario inexperto disponer de web completas, potentes pero ágiles, intuitivas en su utilización y sencillas en su administración. El sistema SICOM con la integración del módulo de funcionamiento deben disponer dichas características

que hará más fácil el trabajo de los militantes del partido, ahorrando tiempo y personal. Hay que tener en cuenta que en el mundo existen cientos de sistemas que poseen características similares pero que no resuelven el problema actual, para ampliar un poco más se hizo un estudio de los sistemas GOC (Gestión de órganos colegiados) – Software para organizar reuniones eficientes y actius que es una aplicación en la nube diseñada para gestionar todas las reuniones y la documentación de una organización.

GOC: Gestión de órganos colegiados

4TIC (Tecnología de la información y la comunicación) es una empresa especializada en el desarrollo de innovadoras soluciones tecnológicas para distintos tipos de proyectos que integra herramientas de mercado con software propio, tanto para empresas, universidades y administraciones públicas. El objetivo de esta empresa es ser referencia en el desarrollo e integración de soluciones tecnológicas tanto en España como en Sudamérica dentro del ámbito universitario y de las Administraciones Públicas, ofreciendo una gran variedad de herramientas y soluciones a medida que se adapten a las necesidades y exigencias de cada entidad u organización que cuente con su servicio.

Debido al ritmo de trabajo al que se enfrenta esta empresa día a día provoca que se tengan que acortar al máximo los plazos para poder ofrecer un servicio eficaz a sus clientes en cualquier proyecto. Es por esta razón que, en la actualidad, han surgido distintas herramientas para organizar reuniones de equipo que cuentan con las últimas novedades en cuanto a tecnologías de la información y la comunicación.

En 4TIC son conscientes de esta necesidad y es por eso que han desarrollado GOC, una plataforma para reuniones cuyo objetivo es organizar y gestionar al máximo el tiempo invertido en este tipo de encuentros de trabajo. Esta herramienta permite gestionar todas las reuniones de equipo de su entidad en una página web en la que toda la información necesaria estará accesible para todos sus integrantes.

Esta herramienta para reuniones eficientes es una tecnología diseñada y desarrollada por 4TIC, en colaboración con la Universitat Jaume I y el resto de universidades públicas valencianas, que permite el acceso a toda la información relativa a la reunión desde cualquier ordenador, lo que evita tener que enviar copias y notificaciones a las personas convocadas a la misma.

Entre las principales funcionalidades de GOC se encuentra la gestión de la fecha, los integrantes o del orden del día, el envío de notificaciones a los asistentes, la reproducción del acta de manera automática,

la integración de todos los sistemas en la herramienta para poder aprovechar la información cada vez que se convoque una nueva reunión y, poder acceder y utilizar la información directamente desde la nube con total seguridad.

Funcionamiento:

El primer punto es dar de alta o integrar tanto los órganos a convocar como los miembros de éstos. Una vez dados de alta, se pueden asignar “Autorizados”, que serán las personas que tendrán acceso a poder realizar las convocatorias de los órganos seleccionados. El administrador de la plataforma es quién asigna estos permisos.

El autorizado crea la reunión, añade el orden del día y notifica a los asistentes, que automáticamente reciben un correo electrónico con los datos de la convocatoria. Estos pueden confirmar la asistencia, indicar un sustituto y añadir comentarios al orden del día recibido. Una vez realizada la reunión, se generan los acuerdos y deliberaciones y se incluyen en el acta, la cual puede firmarse electrónicamente y archivarse para posteriores consultas. (4TIC, 2018)

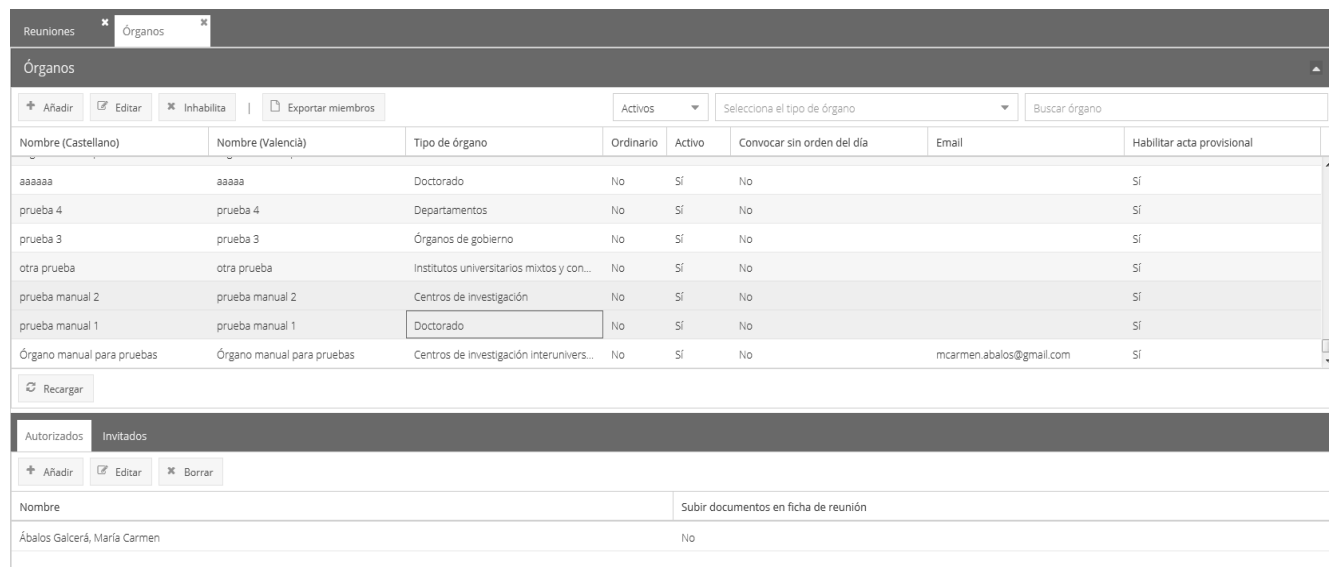


Ilustración 1: Sistema GOC

actius

Es una aplicación en la nube diseñada para gestionar todas las reuniones y la documentación de una organización. Siempre accesible desde cualquier dispositivo con conexión a Internet. Tiene el contenido de la próxima reunión en cualquier momento, el acceso a toda la documentación de los acuerdos tomados en las reuniones, incluso permite saber el estado de las diferentes líneas de trabajo creadas a partir de un acuerdo en concreto.

actius se maneja fácilmente, de forma intuitiva y con una presentación ordenada y atractiva de los temas a tratar. Esto hace de ella el recurso más eficaz y económico para el seguimiento de las decisiones en una organización compleja. Adaptada a cualquier dispositivo móvil, para que los miembros del equipo puedan seguir la reunión con su Tablet o Móvil, o consultarla más tarde. Permite buscar por palabras de cualquier campo y en archivos adjuntos cualquier tema tratado, la localización de las decisiones tomadas con anterioridad la hace en cuestión de segundos.

Funcionamiento:

1. Creación de la reunión

En primer lugar, se crea la reunión, donde se especifica la fecha, lugar y la hora de dicha reunión. Ya se puede empezar a incluir los puntos a tratar en la convocatoria y a anexar cada uno los archivos adjuntos correspondientes. Todos los convocados tendrán acceso a estos archivos sin necesidad de enviar correos. Los asistentes podrán participar y proponer puntos a debatir.

2. Cerrar orden del día

Con sólo pulsar un botón la aplicación enviará un email a todos los convocados avisando de la reunión. Cada asistente podrá acceder al orden del día en la plataforma desde cualquier lugar y consultarán los archivos anexados. actius controla los plazos de las convocatorias, indicando cuando se debe notificar la reunión para un total cumplimiento de los Estatutos de la entidad.

3. Inicio de sesión

Durante la sesión, se podrá ir anotando las deliberaciones y acuerdos adoptados. Podrá crear tareas para los miembros de la organización y empleados y hacer el seguimiento de las mismas en un completo Gestor de tareas.

Esto minimiza toda la labor administrativa posterior, acelerando los procesos y los tiempos de la entidad. En momentos de dificultades hay que preparar la organización o empresa para que sea más versátil y eficaz.

4. Acta en elaboración

Una vez terminada la reunión queda terminar de redactar correctamente los acuerdos, anotar deliberaciones, afinar conceptos tratados. Durante este proceso se podrá enviar a los asistentes emails desde la aplicación para que comprueben los puntos tratados en la misma. Esto despeja el camino a la aprobación del acta, puesto que está a disposición de los asistentes en todo momento.

5. El acta

Una vez los asistentes hayan dado el visto bueno a la redacción final, el acta se aprueba. El acta final, con el formato de la empresa, queda almacenada y firmada electrónicamente. La labor de custodia de las mismas queda así asegurada.

El nivel de transparencia y control de la documentación es total puesto que existirá un acta privada para los asistentes y otra acta pública donde se cuidarán los aspectos personales protegidos por la Ley de protección de Datos. De forma opcional el acta puede publicarse de forma automática para los asociados de la entidad en la página web corporativa (nexius, 2018).



Ilustración 2: logo del sistema actius

1.2.1 Análisis de los sistemas homólogos

En este epígrafe se hace un resumen de los sistemas homólogos estudiados a partir de los aspectos importantes que se van a tener en cuenta en el desarrollo del módulo de funcionamiento y las causas del porque no se pueden integrar en la plataforma SICOM.

Sistema GOC

Este sistema no puede ser unificado en la plataforma SICOM ya que utiliza un servidor privado en la nube, el acta de las reuniones se firma de forma digital, sin embargo, este sistema posee características que pueden ser utilizadas como base para la implementación del módulo de funcionamiento, como es la gestión de la fecha, los integrantes y el orden del día y la reproducción del acta de manera manual.

Sistema actius

Este sistema no puede ser integrado en la plataforma SICOM pues al igual que el sistema GOC utiliza un servidor privado en la nube, además está diseñado solo para sistema operativo Android, no obstante, posee elementos que sirven de apoyo para el desarrollo del módulo de funcionamiento ya que tiene acceso al contenido de la próxima reunión en cualquier momento, además tiene acceso a toda la documentación de las reuniones y se maneja fácilmente, de forma intuitiva y con una presentación ordenada.

Conclusiones

Con el análisis realizado de los sistemas actius y GOC, se permitió tener una panorámica de cómo gestionar la información de todos los temas que se tratan en la reunión, así como el orden del día y los acuerdos, en el caso del módulo de funcionamiento a través de framework de desarrollo djangando y lenguaje de programación python, mientras que los sistemas estudiados utilizaron otras tecnologías. Debido a las características que poseen los sistemas estudiados la UCI se ve en la necesidad de realizar un nuevo módulo que se adapte a las exigencias del cliente.

1.3 Entorno de desarrollo, lenguaje de programación y herramientas

Para el desarrollo del módulo de funcionamiento del SICOM se utilizarán tecnologías, metodología, lenguajes y herramientas específicas. Se establecieron, de acuerdo a las características del proyecto, la integración con el sistema SICOM y en aras de implementar un producto con calidad.

Como entorno de desarrollo más conocido como framework de desarrollo se utilizó **Django** en su versión 1.11, ya que es el que se utiliza en la plataforma SICOM, además, es el mayor *framework* web basado en *Python*. Cuenta con una potente interfaz de administración, ofrece formularios model-based, tiene su propio lenguaje de plantillas y cuenta con una excelente documentación que está disponible de manera gratuita (Django, 2019).

Python es el lenguaje de programación que utiliza *Django*, es un lenguaje de alto nivel que simplifica la sintaxis para escribir programas a partir de unas instrucciones en lenguaje natural o en pseudocódigo, Es muy eficiente y ayuda el proceso de aprendizaje de programación por su claridad en escribir las estructuras algorítmicas, por disponer de una gran cantidad de módulos o librerías de funciones (Python, 2019).

Técnicamente, **Django** utiliza un ORM (Object Relational Mapper) para asignar sus objetos a tablas de base de datos. El mismo código funciona con diferentes bases de datos y facilita la portabilidad entre diferentes tipos de bases de datos, **PostgreSQL**, MySQL, SQLite y Oracle.

El tipo de base de datos utilizado en el desarrollo del módulo es **PostgreSQL**, es un potente sistema de base de datos relacional de objetos de código abierto que utiliza y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de forma segura las cargas de trabajo de datos más complicadas (The PostgreSQL Global Development Group, 2019).

Otros lenguajes utilizados en el desarrollo del módulo son **CSS** y **HTML**, CSS es un lenguaje diseñado para separar el contenido de las páginas web de su presentación. Así, un mismo contenido podría verse de distintas maneras o con distintos aspectos aplicándole distintos estilos, es muy intuitivo y sencillo, ya que para su definición siempre se hace uso de un identificador de etiqueta HTML (como por ejemplo <H1>), y luego indicamos con qué aspecto queremos que se muestren todas las etiquetas <H1> que aparezcan en un documento. Podemos definir cómo queremos que se muestren las distintas partes del

documento HTML, pudiendo en cada caso definir sus propiedades (color, tipo de fuente, tamaño, espacio de márgenes, bordes, imagen de fondo, etc.) con algún determinado valor deseado.

HTML es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable. La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.) así como los diferentes efectos que se quieren dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado).

1.3.1 Herramientas

El Lenguaje Unificado de Modelado (UML, *Unified Modeling Language*) fue adoptado en noviembre de 1997 por Grupo de Administración de Objetos (OMG, *Object Management Group*) como una de sus especificaciones y desde entonces se ha convertido en un estándar de facto para visualizar, especificar y documentar los modelos que se crean durante la aplicación de un proceso de software. UML ha ejercido un gran impacto en la comunidad del software, tanto a nivel de desarrollo como de investigación.

UML es empleado para la documentación de proyectos, actualmente se encuentra en la versión 2.5. No dispone de una metodología de desarrollo, por lo cual se apoya en metodologías de terceros para el desarrollo de proyectos. UML modela diagramas de actividad, clases, comunicación, componentes, estructura compuesta, despliegue, interacción, objetos, paquetes, perfil, máquina de estado, tiempo y casos de uso (Lionel R. Baquero Hernandez, 2017).

1.3.2 Herramientas CASE

Actualmente existen una gran variedad de herramientas CASE (*Computer Aided Software Engineering o Ingeniería de Software Asistida por Computadora*) para el proceso de desarrollo de software. Las herramientas CASE, están tomando cada vez más relevancia en la planeación y ejecución de proyectos que involucren sistemas de información, pues suelen inducir a sus usuarios a la correcta utilización de metodologías que le ayudan a llegar con facilidad a los productos de software construidos.

Todas las herramientas CASE prestan soporte a un lenguaje de modelado para acompañar la metodología y es lógico suponer, que un alto porcentaje de ellas soportan el Lenguaje de Modelado Unificado (UML). Esto se debe a la amplia aceptación de este lenguaje, su valor conceptual y visual, así

como su facilidad para ser extendido para representar elementos particulares a determinados tipos de aplicaciones.

Para el modelado se utilizará *Visual Paradigm* for UML 8.0 por ser una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Agiliza la construcción de aplicaciones con calidad y a un menor coste. Posibilita la generación de bases de datos, transformación de diagramas de Entidad-Relación en tablas de base de datos, así como ingeniería inversa de bases de datos (Montes, 2017).

1.4 Metodología

Metodología AUP-UCI Proceso Unificado Ágil (AUP, por sus siglas en inglés) de Scott Ambler es una versión simplificada del Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés). Esta describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que se mantienen válidos en RUP (UCI, 2017).

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable, se utilizó como metodología de desarrollo la variación del Proceso Unificado Ágil en unión con el Modelo CMMI 1-DEV v1.3 definida para la actividad productiva de la UCI. Esta metodología es la escogida pues el módulo de funcionamiento que se va a realizar se va a integrar al sistema SICOM que utiliza esta misma metodología.

Descripción de las fases:

De las cuatro fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, se unifican las restantes tres fases de AUP en una sola, a la que se llamará Ejecución y se agrega la fase de Cierre.

Las disciplinas definidas en la variación AUP para la UCI son:

- Modelado de negocio
- Requisitos
- Análisis y diseño

- Implementación
- Pruebas internas

Existen tres formas de encapsular los requisitos [Casos de Uso del Sistema (CUS), Historias de usuario (HU) y Descripción de requisitos por proceso (DRP)], para definir cual se debe usar según el tipo de proyecto surgen cuatro escenarios para modelar el sistema.

Características por escenarios

Escenario No 1: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que puedan modelar una serie de interacciones entre los trabajadores del negocio/actores del sistema (usuario), similar a una llamada y respuesta respectivamente, donde la atención se centra en cómo el usuario va a utilizar el sistema. Es necesario que se tenga claro por el proyecto que los CUN muestran cómo los procesos son llevados a cabo por personas y los activos de la organización.

Escenario No 2: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información.

Escenario No 3: 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. Se debe tener presente que este escenario es muy conveniente si se desea representar una gran cantidad de niveles de detalles y la relaciones entre los procesos identificados.

Escenario No 4: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una HU no debe poseer demasiada información.

A partir de toda la información recopilada y los conocimientos adquiridos en la investigación de la metodología AUP-UCI, se decide utilizarla, ya que el presente trabajo es un módulo que será integrado al

sistema SICOM. En este caso particular se estará desarrollando el escenario 2, pues es necesario desarrollar un sistema de gestión de información que muestre al cliente todos los datos recogidos en el sistema.

1.5 Conclusiones del capítulo 1

En el presente capítulo se abordaron los elementos teóricos que dan sustento a la propuesta de solución del problema planteado, en tal sentido se puede arribar a las siguientes conclusiones:

- La definición de los principales conceptos asociados al dominio de la presente investigación permitió alcanzar una mejor comprensión de los referentes teóricos-metodológicos asociados al módulo de funcionamiento.
- El análisis de los sistemas homólogos, permitió identificar diferentes tendencias informáticas como por ejemplo (los sistemas actius y GOC) para la gestión de reunión y se definen características de estos que impiden que sean utilizados en la plataforma SICOM.
- El análisis de la metodología de desarrollo, así como las herramientas, tecnologías y lenguajes de programación utilizados en su implementación, permitió especificar el ambiente de desarrollo para la propuesta de solución.

Capítulo #2. Descripción del módulo de funcionamiento

Para el desarrollo del software fue necesario establecer un entendimiento entre el cliente y el equipo de trabajo para desarrollar un correcto análisis y diseño del sistema. El objetivo de este capítulo es presentar los resultados que se obtuvieron como parte del análisis y el diseño según propone la metodología AUP-UCI. Se especifican las características del sistema, los requisitos funcionales y no funcionales, y se presentan los artefactos generados para dar solución al problema planteado en la investigación.

2.1 Modelo Conceptual.

En esta fase se presentan las características generales del sistema y se realiza un análisis de los conceptos fundamentales asociados al proceso de funcionamiento del PCC los cuales se relacionan a través de un modelo conceptual.

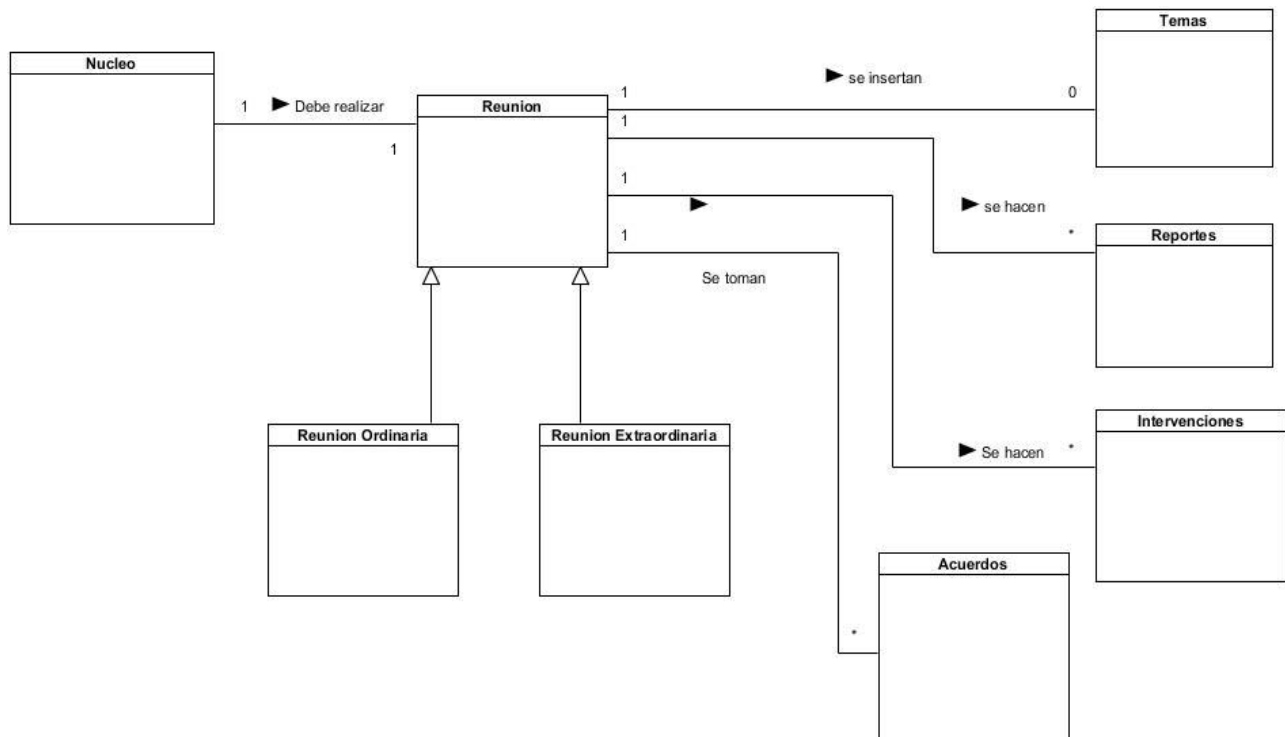


Ilustración 3: Modelo conceptual (elaboración propia)

Núcleo: El núcleo constituye el vínculo indisoluble de la vanguardia con los trabajadores y el pueblo en general y lleva a la práctica la política del partido en el lugar donde actúa. Los núcleos del partido son responsables directos de velar por la ejemplaridad de sus militantes, y tienen el deber de adoptar, oportunamente, las medidas adecuadas cuando conozcan afectaciones o pérdida de esta condición.

Reunión Ordinaria: Las reuniones ordinarias es un espacio donde los militantes del PCC se reúnen para discutir y plantear aspectos tanto positivos como negativos que son importantes para la organización, donde se debe informar con antelación los temas que se van a tratar en dicha reunión.

Reunión Extraordinaria: En el caso de la reunión extraordinaria se realizan solo en caso particular que no se logren abordar todos los temas en la reunión ordinaria o se oriente por el organismo superior la realización de dicha reunión para el análisis de un tema puntual a tratar que no se haya realizado en la Reunión Ordinaria, por tanto, se planifica informando con antelación.

Acuerdos: Resolución tomada por unanimidad o por mayoría de votos sobre cualquier asunto. Decisión reflexionada. Recuerdo, memoria de algo. Además, de significar resolución, el acuerdo es el concierto de dos o más voluntades o inteligencia de personas que llevan a un mismo fin.

Intervención: En su uso más general y amplio, intervención implica la acción y efecto de intervenir algo, una situación, una oficina, un estado entre otros. Normalmente este sentido se emplea para dar cuenta de la interposición de una autoridad en algún área o institución que demanda por fuerza mayor un orden, que se ha perdido por alguna situación extraordinaria.

Temas: Los temas de la reunión, son los aspectos que los militantes van a discutir en la reunión, acerca de ellos se van a plantear inquietudes y se van a tomar acuerdos para tratar de resolver el problema.

Reportes: Los reportes es el resumen de los elementos principales que guardan relación con la reunión como, por ejemplo: reporte de ausente, reporte de causas de ausencia, reporte de tema entre otros.

2.2 Especificación de Requisitos

Modelar los procesos de negocio es una parte esencial de cualquier proceso de desarrollo de software. Permite capturar el esquema general y los procedimientos que gobiernan el negocio. Este modelo provee una descripción de dónde se va a ajustar el sistema de software considerado dentro de la estructura

organizacional y de las actividades habituales. También, provee la justificación para la construcción del sistema de software al capturar las actividades manuales y los procedimientos automatizados habituales que se incorporarán en el nuevo sistema (Sommerville, 2019). En esta fase se presentan los requisitos funcionales y no funcionales del sistema y los respectivos artefactos generados, a partir de la metodología seleccionada.

2.2.1 Requisitos Funcionales

Requisito: Un requerimiento es una descripción de una condición o capacidad que debe cumplir un sistema, ya sea derivada de una necesidad de usuario identificada, o bien, estipulada en un contrato, estándar, especificación u otro documento formalmente impuesto al inicio del proceso.

Requisitos Funcionales

Los requisitos funcionales son los que definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Es importante que se describa el ¿Qué? y no el ¿Cómo? se deben hacer esas transformaciones. Estos requerimientos al tiempo que avanza el proyecto de software se convierten en los algoritmos, la lógica y gran parte del código del sistema.

Tabla 1: Requisitos Funcionales

Requisito	Prioridad	Requisito	Prioridad
RF1: Crear reunión	Alta	RF11: Generar reportes de causas de ausencias a la reunión.	Alta
RF2: Eliminar reunión	Media	RF12: Generar reportes de cantidad de acuerdos tomados en la reunión.	Alta
RF3: Listar reunión	Baja	RF13: Buscar si en un mes determinado no se trató un determinado tema en un núcleo.	Alta
RF4: Editar reunión	Media	RF14: Crear acuerdo de la reunión	Alta

RF5: Generar reporte de cantidad de ausentes a la reunión.	Alta	RF15: Eliminar acuerdo de la reunión	Media
RF6: Generar reportes de principales temas tratados en la reunión.	Alta	RF16: Listar acuerdo de la reunión	Baja
RF7: Buscar temas que más se discuten en un mes.	Media	RF17: Editar acuerdo de la reunión	Media
RF8: Eliminar intervención de la reunión	Alta	RF18 Crear intervención de la reunión	Alta
RF9: Listar intervención de la reunión	Media	RF19: Generar reporte de la cantidad de ausentes por núcleo a la reunión	Media
RF10: Editar intervención de la reunión	Baja		

2.2.2 Requisitos no funcionales

Se obtuvo un total de siete (6) requisitos no funcionales, distribuidos en especificaciones de usabilidad, confiabilidad, eficiencia, funcionalidad y mantenibilidad, los cuales se relacionan a continuación:

RNF1: Del lado del cliente debe instalarse como navegador web Firefox en su versión 35 o superior o Chrome en su versión 32 o superior.

RNF2: Del lado del servidor debe instalarse el intérprete de Python en su versión 3.5, el marco de trabajo Django en su versión 1.11 y el SGBD PostgreSQL en su versión 9.4.

RNF3: El sistema debe ser una aplicación web que puede ser utilizada por personas con poco conocimiento informático.

RNF4: Los tiempos de respuesta del módulo deben ser rápidos, no mayores de 5 segundos para cualquier operación realizada por el usuario.

RNF5: Solo se mostrarán a los usuarios aquellas acciones o informaciones a las que por su responsabilidad o rol dentro del negocio necesitan acceder.

RNF6: El sistema debe presentar elementos identificativos del contexto donde se aplica (Logo PCC).

2.3 Definición de los actores del Sistema

Un actor es quien interactúa con el sistema, inicializa los casos de uso, este puede ser un humano, una máquina o un software. En el caso específico del producto que se propone será un Estadístico Municipal, que se encarga de gestionar toda la información referente con las reuniones ordinarias y extraordinarias, luego interviene el Estadístico Provincial y el Estadístico Nacional, estos dos roles solo se encargan de consultar los reportes para realizar el análisis correspondiente en su nivel.

Tabla 2: Actores del sistema

Actor	Descripción
Estadístico municipal	El estadístico municipal puede insertar en el sistema temas de interés que se deseen tratar en las reuniones en el municipio, recibe la información de cada reunión realizada y las inserta en el sistema, elabora los reportes asociado a su municipio y realiza los análisis pertinentes sobre la calidad de los datos suministrados. Generan reportes de: cantidad de ausentes, % de asistencia, participación del organismo superior, principales temas tratados, cantidad de acuerdos tomados.
Estadístico	Este rol solo se encarga de consultar los reportes para realizar el análisis correspondiente en su nivel. A pesar de esto en cada uno de estos niveles pueden insertar temas a tratar en las reuniones de los núcleos que sean del interés del nivel correspondiente. Generan reportes de cantidad de ausentes, % de asistencia, participación del organismo superior, principales temas tratados, cantidad de acuerdos tomados a su nivel. El estadístico es formado por el estadístico nacional y el estadístico provincial.

2.3.1 Diagrama de casos de uso

Concluido el proceso de levantamiento de requisitos, se continúa con la realización del modelo de casos de uso del sistema. Un modelo de casos de uso describe las funcionalidades propuestas del producto a desarrollar, representa la interacción entre un usuario y el sistema. Para lograr un mejor entendimiento sobre los elementos que componen este modelo.

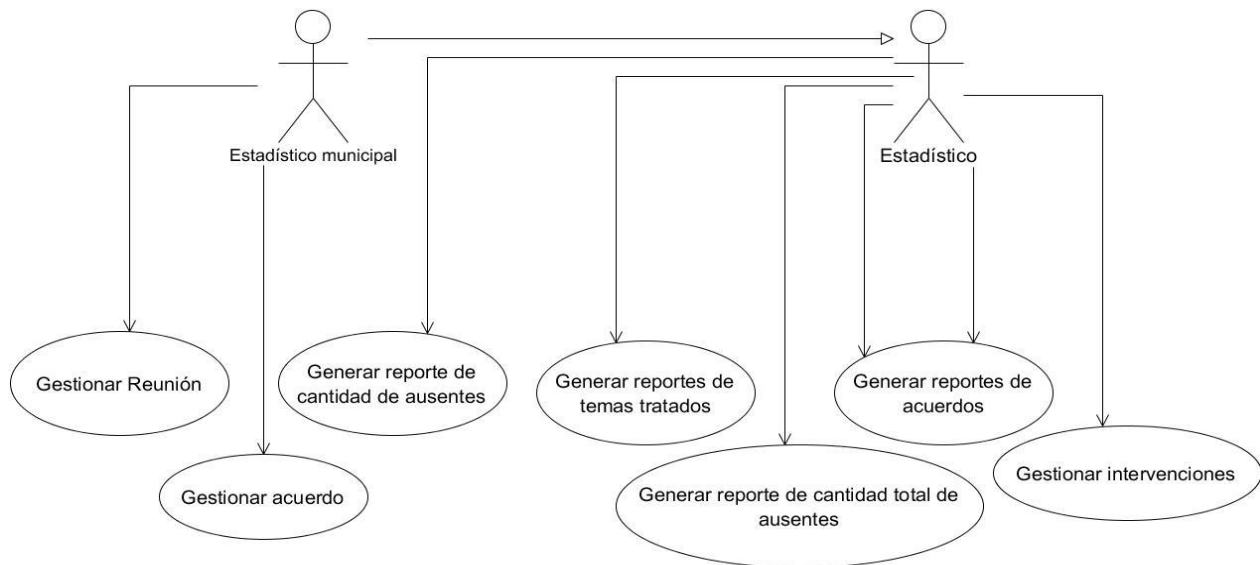


Ilustración 4: Diagrama de caso de uso

Tabla 3: Descripción del RF Gestionar Reunión

Caso de Uso	Gestionar Reunión
Actor	Estadístico Municipal
Resumen	<p>El Proceso comienza cuando el estadístico municipal desea realizar alguna de las siguientes acciones:</p> <ul style="list-style-type: none"> • Insertar una nueva reunión • Editar una reunión • Eliminar una reunión • Visualizar una reunión

Precondiciones	En dependencia de la opción que el estadístico municipal desea realizar: <ul style="list-style-type: none"> • Se Inserta una reunión. • Se edita una reunión. • Se elimina una reunión. • Se visualiza una reunión.
Referencia	RF1, RF2, RF3, RF4
Prioridad	Crítico
Flujo Básico de Eventos	
Acción del actor	Respuesta del Sistema
<p>1. El estadístico municipal selecciona cuál es la acción que desea realizar:</p> <p>1. Se Inserta una reunión.</p> <p>2. Se edita una reunión.</p> <p>3. Se elimina una reunión.</p> <p>4. Se visualiza una reunión.</p>	
	<p>2. 2 En dependencia de la acción seleccionada por el estadístico municipal el Sistema muestra la interfaz correspondiente:</p> <p>1. Insertar una nueva reunión: Ir a la sección “insertar reunión”.</p> <p>2. Modificar datos de una reunión: Ir a la sección “editar reunión”.</p> <p>3. Eliminar una reunión: Ir a la sección “eliminar reunión”.</p>

	4. Visualizar reunión: Ir a la sección “ Visualizar reunión ”.
Sección 1: “Eliminar Reunión”	
Acción del actor	Respuesta del Sistema
1. El estadístico municipal selecciona la opción eliminar reunión.	
	2. El Sistema muestra un mensaje de confirmación para eliminar la reunión
3. El estadístico municipal selecciona la opción eliminar.	
	4. El sistema elimina la reunión
Sección 2: “Visualizar Reunión”	
Acción del actor	Respuesta del Sistema
1. El estadístico municipal selecciona la opción visualizar.	
	2. El sistema ejecuta la opción visualizar reunión.
	3. El sistema muestra un listado con todas las reuniones que existen en la base de datos.
Sección 3: “Insertar Reunión”	
Acción del actor	Respuesta del Sistema
1. El estadístico municipal selecciona la opción insertar.	

	2. El sistema muestra la interfaz para insertar una nueva reunión con los campos (fecha, tiempo de duración, invitado, tipo, acuerdos, intervenciones, cantidad de ausentes, causas de ausencia, temas).
3. El estadístico municipal inserta los datos.	
	4. El sistema verifica que no exista ningún campo vacío.
	5. El sistema verifica que no halla datos incorrectos.
	6. El sistema inserta satisfactoriamente la nueva reunión.
Flujos Alternos al paso 4: “existe campos vacíos”	
Acción del actor	Respuesta del Sistema
	1. El sistema muestra un mensaje “Existen campos vacíos”.
Flujos Alternos al paso 5: “existe datos incorrectos”	
	1. El sistema muestra un mensaje “Existen campos incorrectos”
Sección 4: “Editar Reunión”	
Acción del actor	Respuesta del Sistema
1. El estadístico municipal selecciona la opción editar.	
	2. El sistema muestra un listado con todas las

	reuniones.
3. El estadístico municipal selecciona la reunión que desea editar.	
4. El estadístico municipal edita el campo (fecha, tema, cantidad de participante, entre otros) de la reunión que desea modificar y guarda los cambios.	
	4.El sistema valida completitud de los datos.
	5. El sistema valida que los datos estén correctos.
	6. El sistema actualiza la reunión.
Flujos Alternos al paso 3: “Cancelar Modificación”	
Acción del actor	Respuesta del Sistema
1. Si el estadístico municipal no indica guardar los cambios, se cancela la operación y finaliza así el caso de uso.	
Flujos alternados al paso 5: “Existen campos incorrectos”	
	1. El sistema muestra un mensaje “Existen campos incorrectos”.
Flujos alternados al paso 4: “Existen campos obligatorios vacíos”	
	1. El sistema muestra un mensaje “Existen campos vacíos” y va a la sección 3
Poscondiciones	Gestionar Reunión ha sido evaluado y corregido todos

	los errores en el proceso de modelado.
--	--

2.4 Arquitectura MVT

La arquitectura de software es la estructura de un sistema o bien, la forma en que va a estar organizado este; incluye los elementos que lo conforman, sus propiedades visibles desde el exterior y las relaciones que existen entre ellos

La arquitectura MVT (Modelo de plantilla de vista) es un patrón de diseño de software. Es una colección de tres componentes importantes Modelo Vista y Plantilla. El modelo ayuda a manejar la base de datos. Es una capa de acceso a datos que maneja los datos. La plantilla es una capa de presentación que maneja la parte de la interfaz de usuario completamente. La vista se utiliza para ejecutar la lógica empresarial e interactuar con un modelo para llevar datos y renderizar una plantilla.

Aunque *Django* sigue el patrón MVC, mantiene sus propias convenciones. No hay un controlador separado y la aplicación completa se basa en la Vista de modelo y la Plantilla. Es por eso que se llama aplicación MVT (Holovaty, 2017).

Vea el siguiente gráfico que muestra el flujo de control basado en MVT.

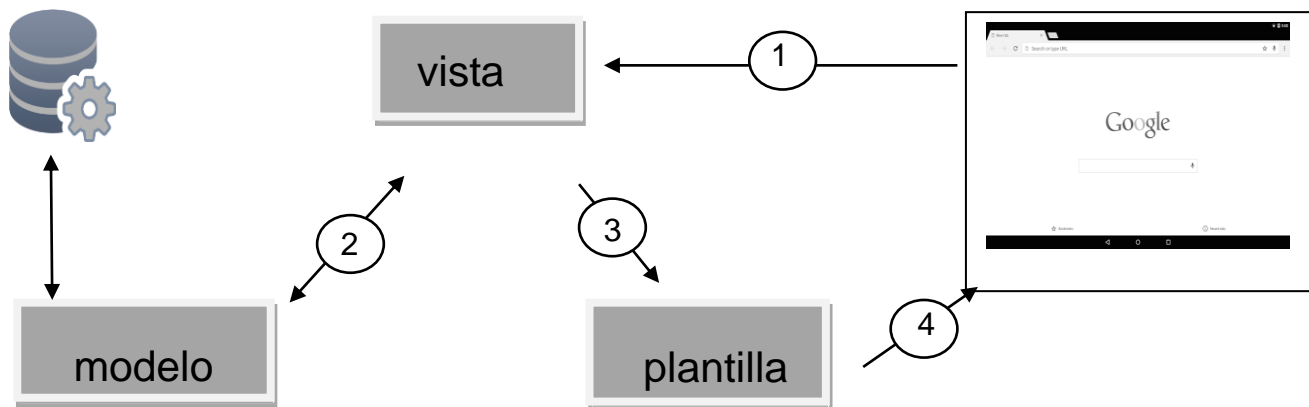


Ilustración 5: Arquitectura MVT (Elaboración Propia)



Ilustración 6: Arquitectura MVT (código)

2.5 Diagrama de clase de diseño

A diferencia del modelo de dominio, un diagrama de clases de diseño muestra definiciones de entidades más que conceptos del mundo real. A pesar de que la metodología empleada en la presente investigación no requiere la realización del diagrama de clases del diseño, se muestra este para un mejor entendimiento de la relación existente entre los casos de uso.

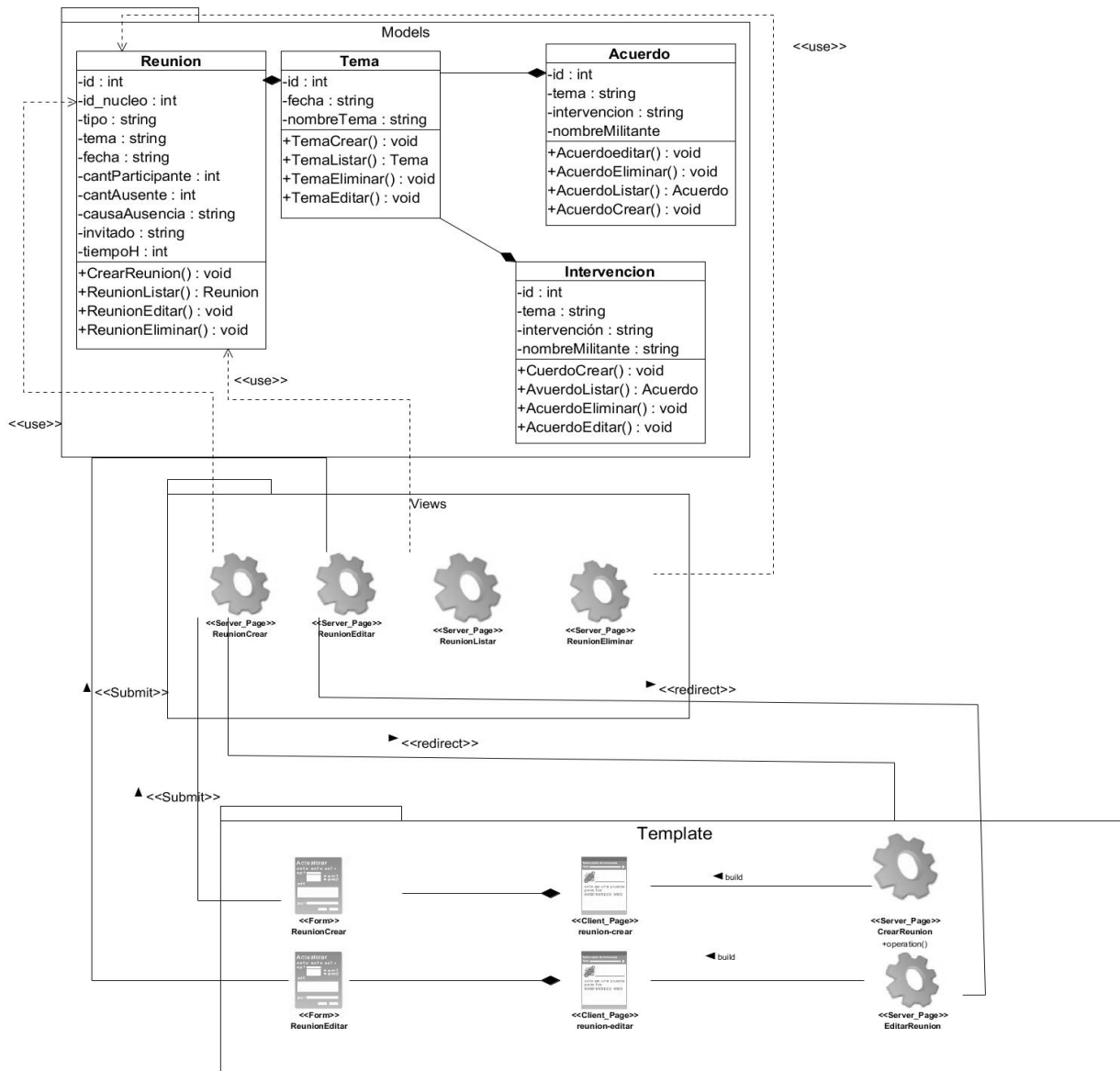


Ilustración 7: Diagrama de clase de diseño

2.6 Patrones de Diseño

Los patrones de diseño de software son una parte importante en el diseño de la arquitectura y en el desarrollo de software. Estos patrones tienen su origen en el diseño de sistemas con el paradigma de la programación orientada a objetos y que son muy recurrentes en la programación de software. Los

patrones de diseño ayudan a mantener un código reutilizable y tener mayor control sobre los problemas que son recurrentes, de ahí que sean una solución factible para su implementación en múltiples aplicaciones (S.Pressman, 2019).

Los patrones de diseño se dividen en dos grupos GRASP (*General Responsibility Assignment Software Patterns* o lo que es lo mismo patrones generales de software para asignar responsabilidades) y GOF (*Gang-of-Four*, o lo que es lo mismo Estrategia y Factoría). Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Los patrones GOF desempeñan varios roles en el proceso de desarrollo orientado a objetos: proveen un vocabulario común para diseño; constituyen una base de experiencia para construir software reusable; y actúan como elementos básicos a partir de los cuales pueden construirse diseños más complejos. Durante el desarrollo del presente trabajo de diploma se hace uso de los siguientes patrones:

GRASP

Alta Cohesión: La cohesión es la medida en la que un componente se dedica a realizar solo la tarea para la cual fue creado, delegando las tareas complementarias a otros componentes. Una clase debe de hacer lo que respecta a su entidad, y no hacer acciones que involucren a otra clase o entidad. Usar este patrón simplifica el mantenimiento y favorece el bajo acoplamiento.

Bajo Acoplamiento: El acoplamiento es la medida en que los cambios de un componente tienden a necesitar cambios de otro componente. Cuando modificamos los atributos de una clase, se tienen que modificar los atributos de otra clase. Este patrón se evidencia en todas las aplicaciones web que funcionen sobre Django, pues cada pieza de las aplicaciones tiene un propósito clave, que puede modificarse sin afectar otras piezas.

Controlador: Es el responsable de asignar a clases específicas la responsabilidad de controlar la salida de sucesos en el sistema, a través de este se permite interactuar con las demás clases del componente.

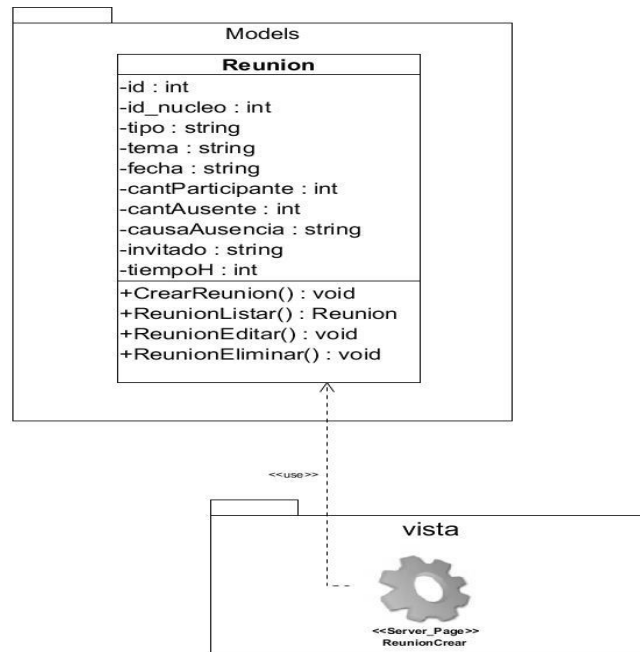


Ilustración 8: Representación de patrones (alta cohesión, bajo acoplamiento y controlador)

Ejemplo: class ReunionCrear (SuccessMessageMixin, CreateView):

```
model = Reunion

form_class = ReunionForm

template_name = 'modulo/reunion_form.html'

success_url = reverse_lazy('URL_modulo: listarReunion_modulo')

success_message = "Se ha registrado satisfactoriamente"
```

Experto: Este patrón es el encargado de asignar una responsabilidad al experto en información, es decir, la clase que posee la información necesaria y suficiente para ejecutar la responsabilidad asignada.

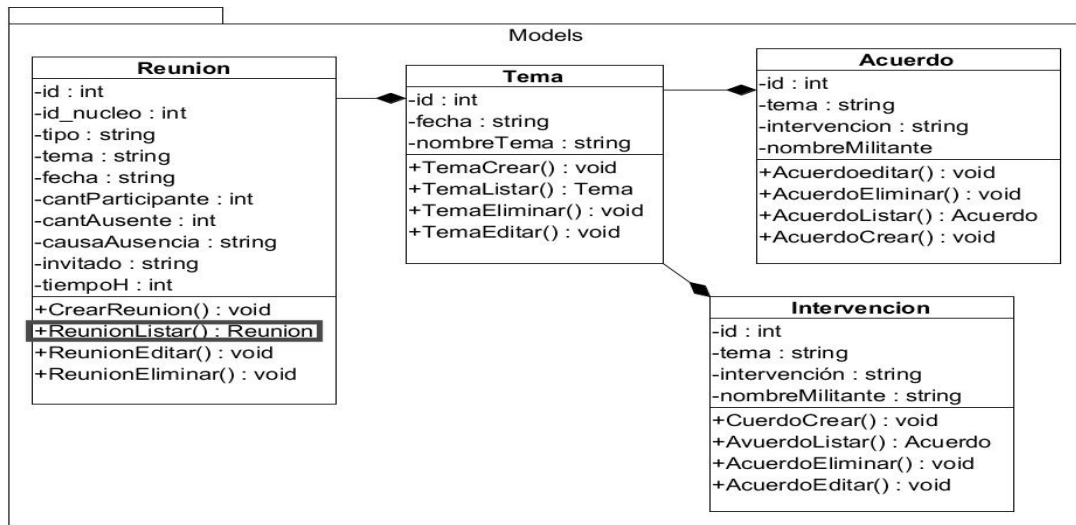


Ilustración 9: Representación del patrón experto

GOF

Decorador: Agrega responsabilidades a un objeto de manera dinámica. Provee una alternativa a la herencia cuando deseamos agregar funcionalidad. Este patrón se usa cuando:

- Se quiere agregar responsabilidades a objetos individuales de manera dinámica y transparente, sin afectar otros objetos.
- Se desea implementar responsabilidades que pueden removerse.
- Cuando la extensión utiliza herencia, es impracticable, por ejemplo, pues provocaría demasiadas variaciones y un gran número de clases.

```
Ejemplo: def Login(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate (username=username, password=password)
    if user is not None:
        login (request, user)
    return render (request, 'base/base.html')
```

```
else: messages.error (request, 'El usuario o contraseña no son correctos. ¡Verifique!!!')
return render (request, 'modulo/login.html')
return render (request, 'modulo/login.html')
```



Ilustración 10: Login

2.7 Conclusiones Parciales

Después de realizado el análisis y diseño de la propuesta de solución y haber generado los diferentes artefactos que dispone la metodología AUP-UCI, se puede concluir lo siguiente:

- El análisis de las características del sistema y la modelación del dominio, permitieron identificar los principales requisitos funcionales y no funcionales del módulo de funcionamiento.
- La identificación de los patrones de diseño y el estilo arquitectónico del sistema, permitió evaluar y mejorar el diseño del módulo de funcionamiento, así como también, resolver problemas en dicho diseño.
- El diseño del diagrama de clases de diseño, facilitó la visión del modelado de una clase y las relaciones entre las clases del módulo.

Capítulo #3. Implementación y prueba del módulo de funcionamiento

3.1. Introducción

Con el propósito de garantizar la corrección de errores se definen tareas que proporcionen la calidad requerida en todas las fases del flujo de trabajo. Las pruebas de software son el proceso de identificar las fallas o defectos en el sistema con el objetivo de asegurar el cumplimiento de los requisitos expuestos por el cliente. En este capítulo se brindan los aspectos referentes a la implementación y se recogen los resultados obtenidos luego de realizar las pruebas necesarias al módulo de funcionamiento, evidenciándose las diferentes técnicas de prueba que permiten comprobar la eficiencia del mismo, para satisfacer así las necesidades del cliente.

3.2. Diagrama de componente

Un diagrama de componentes es un diagrama del Lenguaje Unificado de Modelado. El diagrama de componentes proporciona una visión física de la construcción del sistema de información. Muestra la organización de los componentes software, sus interfaces y las dependencias entre ellos. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software, pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema. Estos diagramas pueden incluir paquetes que permiten organizar la construcción del sistema de información en subsistemas (Pressman, 2019).

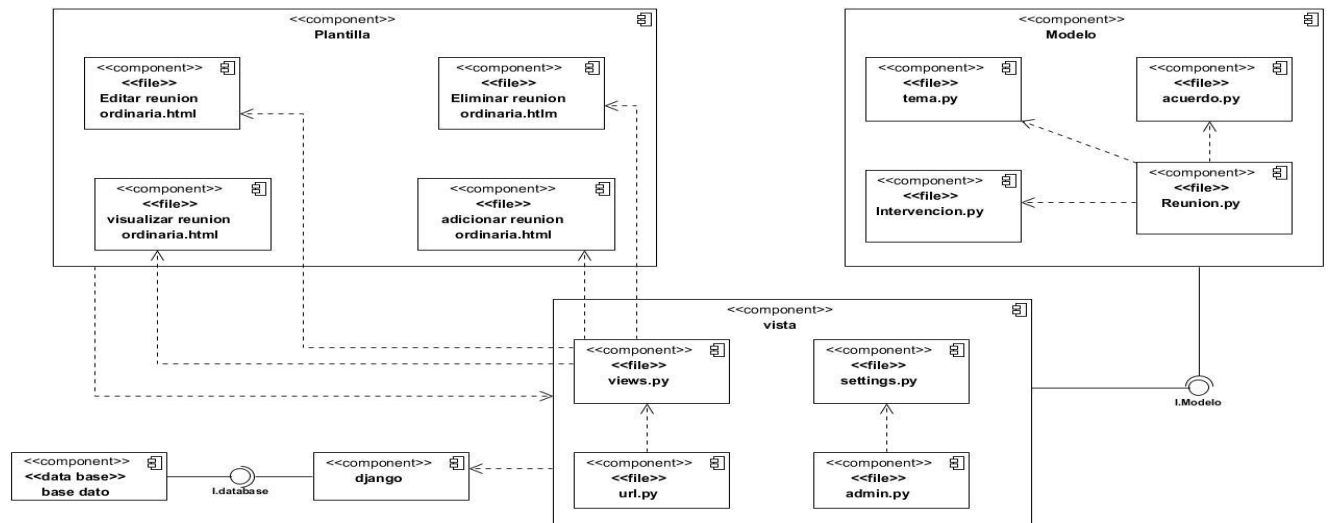


Ilustración 11: Diagrama de componente

3.3. Diagrama de despliegue

Los elementos del diseño del despliegue indican la forma en la que se acomodarán la funcionalidad del software y los subsistemas dentro del ambiente físico de la computación. Los diagramas de despliegue comienzan en forma de descriptor, donde el ambiente de despliegue se describe en términos generales, luego se utilizan formas de instancia y se describen explícitamente los elementos de la configuración. Un diagrama de despliegue muestra el *hardware* de su sistema y el software de ese hardware. Los diagramas de implementación son útiles cuando la solución de software se despliega en varios equipos, cada uno con una configuración única (ESCOBAR, 2017).

Teniendo en cuenta que lo que se está desarrollando es un módulo que se va a integrar al SICOM, el diagrama de despliegue sigue el mismo principio de este sistema que ya existe.

Para el despliegue del módulo se hace necesario un servidor de base de datos y uno web, los mismos se van a comunicar a través del protocolo TCP/IP¹. El servidor web va a ser el que se comunica con las PC

¹ *Transmission Control Protocol/Internet Protocol*, o lo que es lo mismo Protocolo de Control de Transmisión/Protocolo de Internet

de los clientes a través del protocolo de comunicaciones HTTPS². Como se puede observar en el diagrama a continuación.



Ilustración 12: Diagrama de despliegue (Diagrama de despliegue)

PC-Usuario: El nodo PC Cliente representa las estaciones de trabajo de los usuarios que se conectan al sistema.

Servidor de Base de Datos: Representa el servidor de base de datos PostgreSQL que almacenará la información del Módulo.

Servidor Web: Representa el servidor web apache en donde se montará el módulo, el mismo recibe peticiones del cliente y realiza peticiones al servidor de base de datos.

3.4. Estándares de codificación

El propósito fundamental de los estándares de codificación es que tenga una arquitectura y un estilo consistente, independiente, con lo cual el sistema resulte fácil de entender y por supuesto fácil de mantener. Las convenciones de código son importantes para los programadores por un gran número de razones (Prezi, 2019):

- Las convenciones de código mejoran la lectura del software, permitiendo entender código nuevo mucho más rápidamente y más a fondo.
- Si distribuyes tu código fuente como un producto, necesitas asegurarte de que está bien hecho y presentado como cualquier otro producto.

A continuación, se definen los estándares de codificación a utilizar en la implementación del módulo:

² *HyperText Transfer Protocol Secure*, o lo que es lo mismo Protocolo de transferencia de hipertexto

- Los nombres de cada uno de los elementos del programa deben ser significativos.
- No manejar en los programas más de una instrucción por línea.
- Declarar las variables en líneas separadas.
- La mayoría de los elementos se deben nombrar usando sustantivos.
- Los atributos deben comenzar con letra minúsculas.
- Los métodos deben comenzar con letra mayúsculas.
- Para nombrar las clases, interfaces y archivos se deberá de utilizar un sustantivo singular, con la primera letra en mayúscula y las demás en minúsculas.
- Si el nombre de la clase está compuesto por más de una palabra la primera letra de cada una debe de ser mayúscula. Letras mayúsculas seguidas no son permitidas. Para separar palabras se usará la diferenciación entre mayúsculas, minúsculas y guión bajo (CamelCase, 2019)

A continuación, se definen estilos generales de codificación para python:

- Definir variables con nombres que tengan algún sentido o que puedas identificar fácilmente
- Las líneas de código no deben ser muy largas, como mucho 72 caracteres. Si se tiene una línea larga, se puede cortar con una barra invertida (\) y continuar en la siguiente línea.
- Dentro de paréntesis, corchetes o llaves, no dejar espacios inmediatamente dentro de ellos.
- Justo después de coma, punto y coma y punto, separar con un espacio.
- Aunque en Python se pueden hacer varias declaraciones en una línea, se recomienda hacer sólo una en cada línea.
- Además, de los comentarios entre el código explicando cómo funciona, el elemento básico de documentación de *Python* es el *Docstring* (cadena de documentación).
- Nunca mezclar tabulaciones y espacios.
- Separar funciones de alto nivel y definiciones de clase con dos líneas en blanco, mientras que las definiciones de métodos dentro de una clase son separadas por una línea en blanco.
- Las importaciones deben estar en líneas separadas.

3.5 Pruebas de Software

Luego del desarrollo de la propuesta de solución, se hace necesario verificar y validar el sistema implementado a través de una estrategia de pruebas que permita comprobar el cumplimiento de las especificaciones del diseño y de la codificación e identificar los posibles errores cometidos. Las pruebas de software intentan demostrar que un programa hace lo que se intenta que haga, así como descubrir defectos en el programa antes de usarlo.

Al probar el software, se ejecuta un programa con datos artificiales. Hay que verificar los resultados de la prueba que se opera para buscar errores, anomalías o información de atributos no funcionales del programa. Se diseñó una estrategia de prueba para el módulo de funcionamiento, en función de garantizar su seguridad, verificar la conformidad de los requisitos establecidos y validar que el módulo hace lo que el usuario espera.

3.5.1 Pruebas de caja negra

Las pruebas de caja negra, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjunto de condiciones de entradas que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca. En vez de ello, es un enfoque complementario que es probable que descubra una clase de errores diferentes que los métodos de caja blanca (Pressman, 2019).

Las pruebas de caja negra intentan encontrar errores en las categorías siguientes:

1. Funciones incorrectas o faltantes.
2. Errores de interfaz.
3. Errores en la estructura de datos o en el acceso a bases de datos externas.
4. Errores en el comportamiento o rendimiento.
5. Errores de inicialización y terminación.
6. Esta prueba de caja negra se aplica durante las últimas etapas de pruebas.

Diseño de caso de prueba

Un caso de prueba se diseña según las funcionalidades descritas en los casos de uso. El propósito que se persigue con este artefacto es lograr una comprensión común de las condiciones específicas que la

solución debe cumplir. Se comienza con la descripción de los casos de uso del sistema, como apoyo para las revisiones. Cada planilla de caso de prueba recoge la especificación de un caso de uso, dividido en secciones y escenarios, que detalla las funcionalidades descritas en él y cada variable que recoge el caso de uso.

Diseño del caso de prueba del caso de uso Gestionar reunión

Este caso de uso inicia cuando el usuario decide insertar una reunión ya sea ordinaria o extraordinaria, el sistema muestra una interfaz con los campos a llenar, el usuario introduce los datos correctamente, y se inserta la reunión, en caso que desee editar algún campo de la reunión, regresa a la reunión que desee editar, cambia el campo seleccionado correctamente e inserta la reunión, y desea eliminar una reunión, solo se debe seleccionar la reunión y eliminarla.

Gestionar reunión

Tabla 4: Diseño del caso de prueba Gestionar reunión

Nombre de la sección	Escenario	Acción realizada	Respuesta del sistema	Resultado de la prueba
SC1: Insertar reunión	EC1.1 Introducir datos de la reunión correctamente	El usuario introduce los datos de la reunión satisfactoriamente Datos (fecha, temas, cantidad de ausentes, causas de ausencia, invitado, cantidad de participantes, tipo)	El sistema, verifica que todos los campos están llenos y muestra un mensaje indicando que la reunión ha sido insertada correctamente	Resultado satisfactorio
	EC1.2 Introducir datos de la reunión	El usuario introduce los datos de la reunión de forma incorrecta	El sistema muestra una interfaz indicando	Resultado satisfactorio

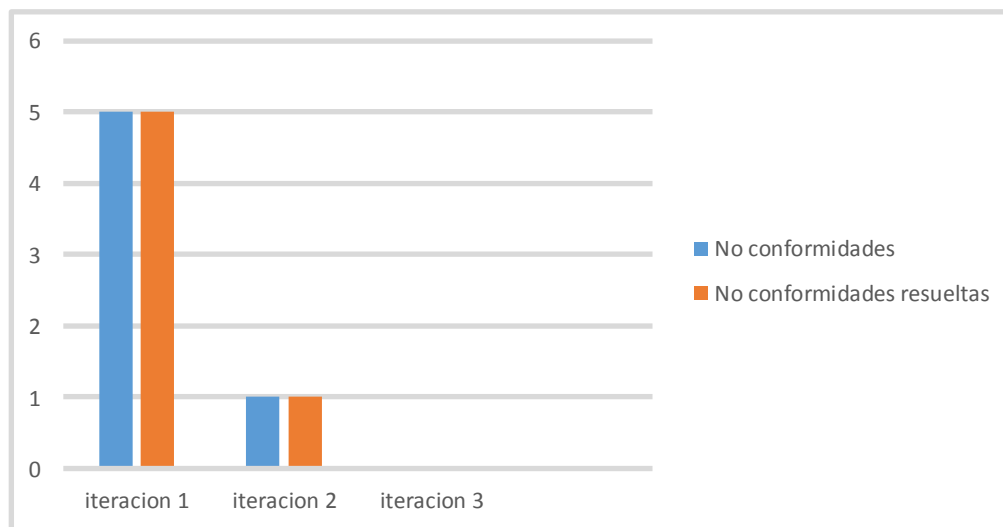
	incorrectamente	Datos incorrectos (tiempo, cantidad de ausentes)	que el campo es incorrecto	
	EC1.3 Insertar Reunión sin haber llenado todos los datos.	El usuario inserta la reunión, dejando campos vacíos campo vacío: (fecha)	El sistema muestra una interfaz indicando el campo vacío	Resultado satisfactorio
SC2 Editar reunión	EC2.1 Editar campo correctamente	El usuario edita el campo que desee satisfactoriamente Campo editado (cantidad de participantes)	El sistema muestra un mensaje indicando que la reunión ha sido insertada satisfactoriamente	Resultado satisfactorio
	EC2.2 Editar datos de la reunión incorrectamente	El usuario edita los datos de la reunión de forma incorrecta Campo incorrecto (tiempo de duración)	El sistema muestra una interfaz indicando que el campo es incorrecto	Resultado satisfactorio
	EC2.3 Editar Reunión sin haber llenado todos los datos	El usuario edita la reunión, dejando campos vacíos Campos vacíos (tipo)	El sistema muestra una interfaz indicando el campo vacío	Resultado satisfactorio

SC3 Eliminar reunión	EC3.1 Eliminar reunión	El usuario elimina una reunión	El sistema muestra un mensaje indicando que la reunión ha sido eliminada	Resultado satisfactorio
----------------------	------------------------	--------------------------------	--	-------------------------

Resultado de las pruebas de caja negra

Las pruebas se desarrollaron en 3 iteraciones en la primera iteración se obtuvieron cinco no conformidades, que fueron resueltas satisfactoriamente. En la segunda iteración donde se encontró una no conformidad que fue resuelta. Para la tercera iteración no se detectaron nuevas no conformidades. En el gráfico se muestran los resultados de la aplicación de las pruebas funcionales al sistema.

Tabla 5: Resultados de las pruebas de caja negra



(Ver anexo 2)

3.5.2 Pruebas de caja blanca

La prueba del camino básico es una técnica de prueba de la Caja Blanca propuesta por Tom McCabe (1976). La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes. Un camino independiente es aquel que introduce por lo menos una sentencia de procesamiento (o valor de condición) que no estaba considerada.

Para obtener el conjunto de caminos independientes se construirá el Grafo de Flujo asociado y se calculará su Complejidad Ciclomática. Para obtener el conjunto de caminos independientes se construirá el Grafo de Flujo asociado y se calculará su Complejidad Ciclomática.

La complejidad ciclomática brinda la cantidad de caminos independientes que deben probarse, puede calcularse de tres maneras

$$V(G) = A - N + 2 \quad A: \text{aristas} \quad N: \text{nodos}$$

$$V(G) = P + 1 \quad P: \text{nodo predicado}$$

$$V(G) = \# \text{Regiones}$$

Las regiones son las áreas delimitadas por las aristas y nodos. También se incluye el área exterior del grafo, contando como una región más. Las regiones se enumeran y la cantidad de regiones es equivalente a la cantidad de caminos independientes del conjunto básico de un programa. Ver código Anexo (3)

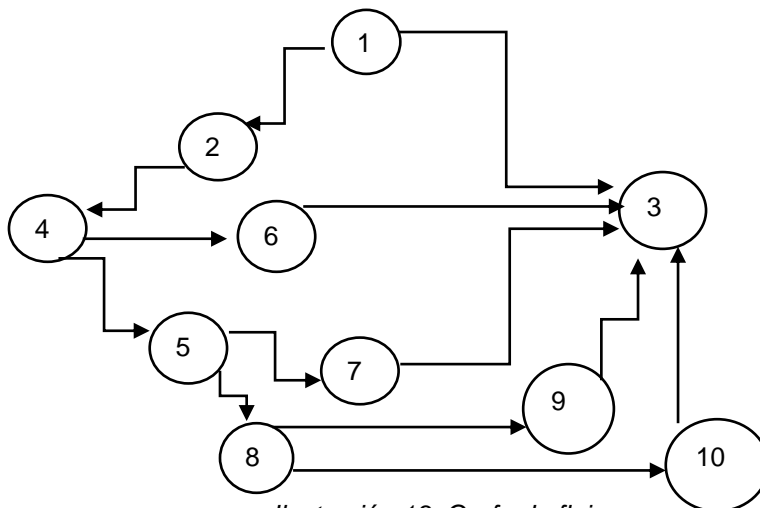


Ilustración 13: Grafo de flujo

Complejidad ciclomática

$$V(G) = NA \text{ (Número de Aristas)} - ND \text{ (Número de Nodos)} + 2$$

$$V(G) = 13 - 10 + 2$$

$$V(G) = 5$$

$$V(G) = P \text{ (Nodos predicados)} + 1$$

$$V(G) = 4 + 1$$

$$V(G) = 5$$

$$V(G) = R \text{ (Número de regiones)}$$

$$V(G) = 5$$

Caminos Básicos

CB1: 1-3

CB2: 1-2-4-6-3

CB3: 1-2-4-5-7-3

CB4: 1-2-4-5-8-9-3

CB5: 1-2-4-5-8-10-3

Estrategia de Prueba para el camino básico 1

```
if request.method == 'POST':
```

```
    return render(request,
```

Resultado esperado: Se ha registrado la reunión correctamente.

Ver anexo (4)

También, se realizaron pruebas de caja blanca con *unittest*, también llamado **PyUnit**, este forma parte de una familia de herramientas conocida colectivamente como *xUnit*, un conjunto de *frameworks* basados en el software. Ver anexo (7)

3.5.3 Pruebas de Integración

Las pruebas de integración permiten probar el sistema en su conjunto y con otros sistemas con los que se relaciona para verificar que las especificaciones funcionales y técnicas se cumplen. Su objetivo es detectar errores de interfaces y relaciones entre componentes (It-Mentor, 2015).

Las pruebas de integración son una técnica sistemática para construir la arquitectura del software mientras se llevan a cabo pruebas para descubrir errores asociados con la interfaz. El objetivo es tomar los componentes probados de manera individual y construir una estructura de programa que se haya dictado por diseño (Pressman, 2005)

Existen dos tipos de integración: no incremental e incremental. La integración no incremental combina todos los componentes y se prueba el programa en su conjunto. La integración incremental consiste en probar el programa en pequeños segmentos para facilitar el aislamiento y la corrección de errores. En la propuesta de solución se escogió el enfoque incremental para la realización de las pruebas de integración mediante la estrategia integración descendente, aplicando la técnica de integración primero en anchura. Esta técnica incorpora todos los componentes directamente subordinados en cada nivel y se mueve horizontalmente a través de la estructura.

La estructura del proyecto permite definir como se integran y se relacionan los componentes con los diferentes directorios. Mediante la aplicación de la técnica integración primero en anchura se estableció un diagrama de tres niveles con los principales componentes que se integran.

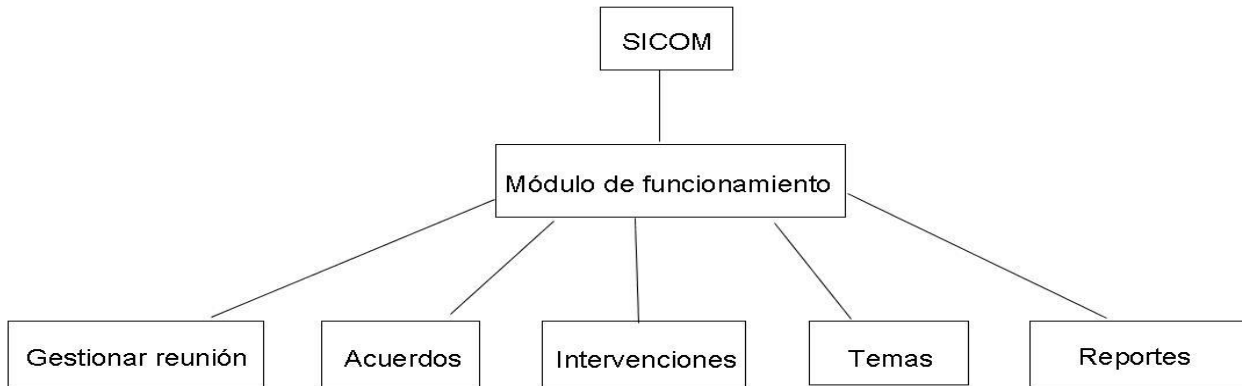


Ilustración 14: Prueba de integración

Los componentes Gestionar Reunión, Acuerdos, Intervenciones, Temas y los diferentes tipos de reporte se integrarían primero al módulo de cotización este a su vez al siguiente nivel de control, el SICOM. La integración de este módulo se evidencia cuando se importa del SICOM la lista de los militantes y la lista de los núcleos.

3.5.4 Pruebas de aceptación

Las pruebas de aceptación, también, llamadas pruebas del cliente, son especificadas por el cliente y se centran en las características y funcionalidad generales del sistema que son visibles y revisables por parte del cliente. En las pruebas de aceptación se prueba una versión del sistema que podría ser entregada al cliente, quien valida que el sistema satisface sus necesidades. Si la entrega es lo suficientemente buena, el cliente puede entonces aceptarla para su uso.

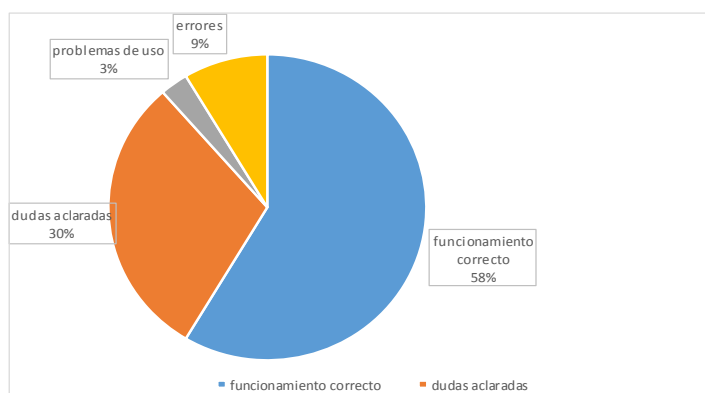
El módulo de funcionamiento fue sometido a un proceso de aceptación en el centro CISED por parte de militantes del PCC durante este proceso se destacaron los siguientes resultados:

- El módulo obtenido ayuda a mitigar pérdidas de tiempo en la organización.
- Constituye una herramienta informática que elimina el trabajo manual, y facilita el proceso de reunión.
- Toda la información que se recoge luego de la reunión queda especificada en el módulo, eliminando así errores innecesarios.

- Los informes de control de la militancia quedan conformados automáticamente que concluya la reunión.
- En general el módulo evita errores en la confección de los informes y gasto de esfuerzo por parte de la organización.

La gráfica que se muestra a continuación muestra los resultados estadísticos luego de haber realizado la prueba de aceptación.

Tabla 6: Resultados estadísticos de la prueba de aceptación



3.6 Conclusiones del capítulo

En este capítulo se definieron todos los elementos para la implementación del módulo de funcionamiento, además fueron realizadas las pruebas de software, lo cual permitió arribar a las siguientes conclusiones.

- La elaboración del diagrama de componente facilitó la organización de los componentes software, sus interfaces y las dependencias entre ellos.
- El correcto uso de los estándares de codificación permitió una mejor comprensión y organización del código.
- La implementación del módulo presentó un producto funcional, adaptado a las necesidades del cliente.
- La estrategia de prueba elaborada, permitió la detección y corrección de errores que surgieron en la propuesta de solución, que a su vez fueron solucionados.

Conclusiones Generales

El módulo de funcionamiento asociado al sistema SICOM permitió mejorar la gestión de la información obtenida durante las reuniones de un núcleo del PCC, una vez cumplido los objetivos trazados en la investigación se concluye lo siguiente:

- El estudio de los principales aspectos teóricos relacionados con los sistemas de Gestión de Información permitió sentar las bases para el posterior desarrollo del módulo de funcionamiento.
- El diseño del módulo de funcionamiento permitió obtener los artefactos necesarios para guiar el proceso de desarrollo del mismo.
- Como resultado de la implementación se obtuvo el módulo de funcionamiento asociado al SICOM para los militantes del PCC que cumple con todos los requisitos funcionales identificados en la fase de ejecución.
- Las pruebas internas realizadas permitieron probar el correcto funcionamiento del Módulo de funcionamiento y los resultados satisfactorios de las mismas otorgaron validez a la presente investigación, siendo esto la base para que el sistema pueda favorecer los procesos que tienen lugar en el CCPCC.

Recomendaciones

El objetivo general trazado en el presente trabajo de diploma fue alcanzado; no obstante, se recomienda:

- El módulo de funcionamiento debería tener como funcionalidad la exportación de las actas en pdf o cualquier formato duro.

Bibliografía referenciada

4TIC. 2018. GOC. [En línea] 2018. <https://www.4tic.com/software-reuniones>.

Alonso, Concha Aparicio. 2017. envejecer ACTIVOS. *env*

Use the "Insert Citation" button to add citations to this document.

ejecer ACTIVOS. [En línea] 2017. <https://envejeceractivos.com/procesamiento-de-la-informacion/>.

Ávila, Jaime. 2018. Prezi. *Prezi*. [En línea] 2018. <https://prezi.com/3ukr6egk76/tema-4-visualizacion-de-la-informacion/>. Use the "Insert Citation" button to add citations to this document.

Berenguer, Abel Díaz. 2018. Universidad de las Ciencias Informáticas. *Universidad de las Ciencias Informáticas*. [En línea] UCI, 2018. <https://www.uci.cu/investigacion-y-desarrollo/centros-de-desarrollo/centro-de-identificacion-y-seguridad-digital-cised>.

Bos, Bert. 2019. Cascading Style Sheets. [En línea] 2019. <https://www.deepl.com/translator#en/es/Cascading%20Style%20Sheets%20Ahome%20page>.

CamelCase. 2019. CamelCase. [En línea] 2019. <https://www.neumoytoraxpanama.org/cloud/resources/documentos/camelcase.pdf>.

Cisnero. 2018. conceptos-basicos-y-generales-de-la-programacion. *conceptos-basicos-y-generales-de-la-programacion*. [En línea] 2018. <https://www.evirtualcisneros.xyz/articulos/8-tecnologia/25-conceptos-basicos-y-generales-de-la-programacion>.

Cruz, Rojas López. 2016. *Sistema de Gestión Electoral*. 2016.

CubaEduca. 2019. CubaEduca. [En línea] CINESOFT, 2019. <http://www.cubaeduca.cu>.

Delgado, Lic. Sofía Mabel Manso. 2018. Radio Rebelde. *Radio Rebelde*. [En línea] 2018. <http://www.radiorebelde.cu/50-revolucion/historia/partido-comunista.html>.

Django Software Foundation. 2019. Django. [En línea] Django, 2019. <https://www.djangoproject.com/>.

ESCOBAR, ING. CARLOS AUGUSTO MENESES. 2017. *INFORME PR*. 2017.

Figueroba, Alex. 2017. *La Teoría del Procesamiento de la Información*. 2017.

Gomis, Pedro. 2018. *Fundamentos de Programación en python*. 2018.

Hernández, Leticia Martínez. 2018. Hacia un mejor uso de las Tecnologías de la Información y la Comunicación. *Hacia un mejor uso de las Tecnologías de la Información y la Comunicación*. [En línea]

Granma, 2018. <http://www.granma.cu/cuba/2018-09-14/hacia-un-mejor-uso-de-las-tecnologias-de-la-informacion-y-la-comunicacion-14-09-2018-21-09-38?page=1>.

Holovaty, Adrian. 2017. *django*. 2017.

Lionel R.Baquero Hernandez, Dayana Mendoza Peña, Osviel Rodríguez Valdésmar. 2017. *Herramientas UML*. 2017.

Martínez, María Isabel Rueda. 2018. *La Gestión del Conocimiento y la Ciencia de la Información*. s.l. : Departamento de Biblioteconomía y Documentación , Facultad de humanidades, Comunicación y Documentación, 2018.

Montes, María. 2017. *Herramientas case*. 2017.

nexius. 2018. *actius. actius*. [En línea] 2018. <http://actius.net/>.

Pérez, Víctor Avendaño y Urbáez, Matilde Flores. 2018. *Modelos teóricos de gestión del conocimiento*. 2018.

Pérez-Montoro, Dr. Mario. 2018. *Visualización de la Información*. 2018.

Prezi. 2019. Prezi. [En línea] 2019. <https://prezi.com/ngfpqli1fklt/estandares-de-codificacion>.

python. 2019. python. [En línea] 2019. <https://www.python.org>.

Radio Rebelde. 2019. Historia del Partido Comunista de Cuba. [En línea] Radio Rebelde, 2019. <http://www.radiorebelde.cu/50-revolucion/historia/partido-comunista.html>.

Rafael Montero, Víctor Fajardo. 2016. *Captura de Información*. 2016.

S.Pressman, Roger. 2019. *Ingeniería de software*. 2019.

Sommerville, Ian. 2019. *SOFTWARE ENGINEERING*. 2019.

The PostgreSQL Global Development Group. 2019. PostgreSQL. [En línea] 2019. <https://www.postgresql.org/>.

UCI. 2017. *Metodología de desarrollo para la Actividad productiva de la UCI*. 2017.

W3C. 2019. HTML5. [En línea] 2019. <https://www.w3.org/html/>.

Suárez, Pablo. 2016. Materia. *Materias*. [En línea] 2003. [Citado el: 5 de Mayo de 2015.] http://materias.fi.uba.ar/7507/content/20101/lecturas/documentacion_pruebas.pdf.

Venete, Adriana. 2017. Ingeniería del Software. *Ingeniería del Software*. [En línea] 18 de Mayo de 2011. [Citado el: 13 de Diciembre de 2014.] <http://mahara.uji.es/view/artefact.php?artefact=54800&view=4648>.

Visual-Paradigm. 2018. Visual-Paradigm. *Visual-Paradigm*. [En línea] Visual-Paradigm.com, 2014. [Citado el: 15 de Diciembre de 2014.] <http://www.visual-paradigm.com/editions/community.jsp>.

WordReference Corporation. 2014. WordReference. *WordReference*. [En línea] Online Language Dictionaries, 2014. [Citado el: 20 de Octubre de 2014.] www.wordreference.com/definicion/adicional.

WordReference. 2015. WordReference. *WordReference Corporation*. [En línea] WordReference, 2015. [Citado el: 24 de Febrero de 2015.] <http://www.wordreference.com/definicion/componente>.

Zayas, Carlos Alvarez de. 1995. *Metodología de la investigación científica*. Santiago de Cuba : Centro de estudio de Educación Superior, 1995.

Anexos

Anexo 1: Entrevista realizada al cliente

- ✚ ¿Qué otra funcionalidad tiene el SICOM?
- ✚ ¿Cómo funciona actualmente el PCC?
- ✚ ¿Cómo funciona la Gestión de la Información en las reuniones?
- ✚ ¿Los militantes interactúan en la toma de acuerdos?
- ✚ ¿Cuáles son los reportes que se generan de las reuniones?
- ✚ ¿Quiénes interactúan en la creación del acta de la reunión?
- ✚ ¿Qué requisitos se debe tener para desplegar el módulo de funcionamiento?
- ✚ ¿A qué sector está dirigido el módulo?

Anexo 2: No conformidades de las pruebas de caja negra

Las no conformidades están agrupadas en errores ortográficos y fallos en el código.

Ejemplo: Faltas de ortografía

Errores en las funcionalidades

Anexo 3: Capturar información adicional.

```

if request.method == 'POST':
form = ReunionForm(request.POST)
if form.is_valid():
tipo = request.POST['tipoReunion']
fecha = request.POST['fecha']
if tipo == 'Ordinaria':
subFechas = fecha.split('-')
resultado=Reunion.objects.filter(fecha__year=subFechas[0],
fecha__month=subFechas[1],
tipoReunion='Ordinaria')
if(len(resultado) == 0):
form.save()

```

```

messages.success(request, 'Se ha registrado la reunión correctamente.')
return redirect('URL_modulo:listarReunion_modulo')
else:
messages.error(request, 'Ya existe una reunión registrada para esta fecha. ¡Verifique!!!')
elif tipo == 'Extraordinaria': form.save()
messages.success(request, 'Se ha registrado la reunión correctamente.')
return redirect('URL_modulo:listarReunion_modulo')
    else:
        form = ReunionForm()
        messages.error(request, 'Se ha introducido un campo incorrecto. ¡Inténtelo de nuevo!!!')
        return render(request, 'modulo/reunion_form.html', {'form':form})

return render(request, 'modulo/reunion_form.html', {'form':ReunionForm})

```

Anexo 4: Campo Incorrecto

```

if request.method == 'POST':
form = ReunionForm(request.POST)
if form.is_valid():
form = ReunionForm
return render(request, 'Se ha introducido un campo incorrecto. ¡Inténtelo de nuevo!!!')

```

Anexo 5: Descripción del caso de uso gestionar acuerdo.

Caso de Uso	Gestionar Acuerdo
Actor	Estadístico Municipal
Resumen	<p>El Proceso comienza cuando el estadístico municipal desea realizar alguna de las siguientes acciones:</p> <ul style="list-style-type: none"> • Insertar un acuerdo

	<ul style="list-style-type: none"> • Editar un acuerdo • Eliminar un acuerdo • Visualizar un acuerdo
Precondiciones	<p>En dependencia de la opción que el estadístico municipal desea realizar:</p> <ul style="list-style-type: none"> • Se Inserta un acuerdo • Se edita un acuerdo • Se elimina un acuerdo • Se visualiza un acuerdo.
Referencia	RF14, RF15, RF16, RF17
Prioridad	Crítico

Flujo Básico de Eventos

Acción del actor	Respuesta del Sistema
<p>1. El estadístico municipal selecciona cuál es la acción que desea realizar:</p> <p>1. Se Inserta un acuerdo.</p> <p>2. Se edita un acuerdo.</p> <p>3. Se elimina un acuerdo.</p> <p>4. Se visualiza un acuerdo.</p>	
	<p>2. 2 En dependencia de la acción seleccionada por el estadístico municipal el Sistema muestra la interfaz correspondiente:</p> <p>3. Insertar un acuerdo: Ir a la sección “acuerdo”.</p>

	<p>4. Modificar datos de un acuerdo: Ir a la sección “editar acuerdo”.</p> <p>3. Eliminar un acuerdo: Ir a la sección “eliminar acuerdo”.</p> <p>4. Visualizar acuerdo: Ir a la sección “acuerdo”.</p>
--	---

Sección 1: “Eliminar Acuerdo”

Acción del actor	Respuesta del Sistema
1. El estadístico municipal selecciona la opción eliminar acuerdo.	
	2. El Sistema muestra un mensaje de confirmación para eliminar el acuerdo
3. El estadístico municipal selecciona la opción eliminar.	
	4. El sistema elimina el acuerdo

Sección 2: “Visualizar Acuerdo”

Acción del actor	Respuesta del Sistema
1. El estadístico municipal selecciona la opción acuerdo.	
	2. El sistema ejecuta la opción acuerdo.
	3. El sistema muestra un listado con todos los acuerdos que existen en la base de datos.

Sección 3: “Insertar acuerdo”

Acción del actor	Respuesta del Sistema
1. El estadístico municipal selecciona la opción insertar acuerdo.	
	2. El sistema muestra la interfaz para insertar un acuerdo
3. El estadístico municipal escribe el acuerdo que desea insertar	
	4. El sistema verifica que no exista ningún campo vacío.
	5. El sistema inserta satisfactoriamente el acuerdo
Flujos Alternos al paso 4: “existe campos vacíos”	
Acción del actor	Respuesta del Sistema
	1. El sistema muestra un mensaje “Existen campos vacíos”
Sección 4: “Editar Acuerdo”	
Acción del actor	Respuesta del Sistema
1. El estadístico municipal selecciona la opción editar	
	2. El sistema muestra un listado con todos los acuerdos
3. El estadístico municipal selecciona el acuerdo que desea editar	

4. El estadístico municipal edita el acuerdo	
	5. El sistema valida que los datos estén correctos
	6. El sistema actualiza el acuerdo
Flujos Alternos al paso 3: “Cancelar Modificación”	
Acción del actor	Respuesta del Sistema
1. Si el estadístico municipal no indica guardar los cambios, se cancela la operación y finaliza así el caso de uso.	
Flujos alternados al paso 5: “Existen campos incorrectos”	
	1. El sistema muestra un mensaje “Existen campos incorrectos”
Poscondiciones	Gestionar acuerdo ha sido evaluado y corregido todos los errores en el proceso de modelado

Anexo 6: Gestionar Intervención

Caso de Uso	Gestionar Intervención
Actor	Estadístico Municipal
Resumen	<p>El Proceso comienza cuando el estadístico municipal desea realizar alguna de las siguientes acciones:</p> <ul style="list-style-type: none"> • Insertar una intervención • Editar una intervención • Eliminar una intervención • Visualizar una intervención

Precondiciones	<p>En dependencia de la opción que el estadístico municipal desea realizar:</p> <ul style="list-style-type: none"> • Se inserta una intervención • Se edita una intervención • Se elimina una intervención • Se visualiza una intervención
Referencia	RF8, RF9, RF10, RF18
Prioridad	Crítico

Flujo Básico de Eventos

Acción del actor	Respuesta del Sistema
<p>1. El estadístico municipal selecciona cuál es la acción que desea realizar:</p> <ul style="list-style-type: none"> 1. Se Inserta una intervención 2. Se edita una intervención 3. Se elimina una intervención 4. Se visualiza una intervención 	
	<p>2. 2 En dependencia de la acción seleccionada por el estadístico municipal el Sistema muestra la interfaz correspondiente:</p> <ul style="list-style-type: none"> 5. Insertar una intervención: Ir a la sección “intervención”. 6. Modificar datos de una intervención: Ir a la sección “editar intervención”. 3. Eliminar una intervención: Ir a la sección

	<p>“eliminar intervención”.</p> <p>4. Visualizar una intervención: Ir a la sección “intervención”.</p>
Sección 1: “Eliminar intervención”	
Acción del actor	Respuesta del Sistema
1. El estadístico municipal selecciona la opción eliminar intervención.	
	2. El Sistema muestra un mensaje de confirmación para eliminar la intervención
3. El estadístico municipal selecciona la opción eliminar.	
	4. El sistema elimina la intervención
Sección 2: “Visualizar intervención”	
Acción del actor	Respuesta del Sistema
1. El estadístico municipal selecciona la opción intervención.	
	2. El sistema ejecuta la opción intervención.
	3. El sistema muestra un listado con todas las intervenciones que existen en la base de datos.
Sección 3: “Insertar intervención”	
Acción del actor	Respuesta del Sistema

1. El estadístico municipal selecciona la opción insertar intervención.	
	2. El sistema muestra la interfaz para insertar una intervención
3. El estadístico municipal escribe la intervención que desea insertar	
	4. El sistema verifica que no exista ningún campo vacío.
	5. El sistema inserta satisfactoriamente la intervención
Flujos Alternos al paso 4: “existe campos vacíos”	
Acción del actor	Respuesta del Sistema
	1. El sistema muestra un mensaje “Existen campos vacíos”
Sección 4: “Editar intervención”	
Acción del actor	Respuesta del Sistema
1. El estadístico municipal selecciona la opción editar	
	2. El sistema muestra un listado con todas las intervenciones
3. El estadístico municipal selecciona la intervención que desea editar	

4. El estadístico municipal edita la intervención	
	5. El sistema valida que los datos estén correctos
	6. El sistema actualiza la intervención
Flujos Alternos al paso 3: “Cancelar Modificación”	
Acción del actor	Respuesta del Sistema
1. Si el estadístico municipal no indica guardar los cambios, se cancela la operación y finaliza así el caso de uso.	
Flujos alternados al paso 5: “Existen campos incorrectos”	
	1. El sistema muestra un mensaje “Existen campos incorrectos”
Poscondiciones	Gestionar intervención ha sido evaluado y corregido todos los errores en el proceso de modelado

Anexo 7: Prueba de caja blanca ()

```
tests.py x
1 from django.contrib.auth.models import User
2   from django.test import TestCase
3
4   # Create your tests here.
5   from django.core.urlresolvers import reverse
6   from django.contrib.auth.models import User
7
8
9   class FuncionamientoTest(TestCase):
10      def setUp(self):
11          test_user1 = User.objects.create_user(username='yeni', password='12345')
12          test_user1.save()
13
14      def test_listar_reunion_page(self):
15          login = self.client.login(username='yeni', password='12345')
16          resp = self.client.get(reverse('URL_modulo:listarReunion_modulo'))
17          self.assertEqual(str(resp.context['user']), 'osay')
18          self.assertEqual(resp.status_code, 200)
19          self.assertTemplateUsed(resp, 'modulo/reunion_list.html')
20
21      def test_listar_acuerdo_page(self):
22          login = self.client.login(username='yeni', password='12345')
23          resp = self.client.get(reverse('URL_modulo:listarAcuerdo_modulo'))
24          self.assertEqual(str(resp.context['user']), 'osay')
25          self.assertEqual(resp.status_code, 200)
26          self.assertTemplateUsed(resp, 'modulo/acuerdo_list.html')
27
```

```
tests.py x
15 login = self.client.login(username='yeni', password='12345')
16 resp = self.client.get(reverse('URL_modulo:listarReunion_modulo'))
17 self.assertEqual(str(resp.context['user']), 'osay')
18 self.assertEqual(resp.status_code, 200)
19 self.assertTemplateUsed(resp, 'modulo/reunion_list.html')
20
21 def test_listar_acuerdo_page(self):
22     login = self.client.login(username='yeni', password='12345')
23     resp = self.client.get(reverse('URL_modulo:listarAcuerdo_modulo'))
24     self.assertEqual(str(resp.context['user']), 'osay')
25     self.assertEqual(resp.status_code, 200)
26     self.assertTemplateUsed(resp, 'modulo/acuerdo_list.html')
27
```

Run: SGRPCC x Test: apps.reporte.tests.FuncionamientoTest x

Test Results 284 ms

Tests passed: 2 of 2 tests - 284 ms

Testing started at 2:49 PM ...
E:\proyectos\newEnv\Scripts\python.exe "C:\Program Files\JetBrains\PyCharm 2019.1.1\he
Creating test database for alias 'default'...
Destroying old test database for alias 'default'...
Destroying test database for alias 'default'...

Process finished with exit code 0

Anexo 8: Carta de aceptación

UCI Universidad de las Ciencias Informáticas | **Acta de aceptación**

ACTA DE ACEPTACIÓN

En cumplimiento del Contrato de servicios para el desarrollo del Sistema para Control de Militantes del PCC con el CCPCIC y en función de la ejecución del proyecto SICCMI, se hace entrega de los productos que se relacionan a continuación:

- Aplicación web "Sistema para Control de Militantes del PCC. Módulo Funcionamiento".

La Parte Cliente, luego de haber revisado los productos de trabajo determina que se **Aceptan**.

Comentarios:

Entrega:  Recibe: 

Nombre y apellidos: **Yenil Peña Rodríguez** Nombre y apellidos: **Osay González Fuentes**

Cargo: Programador Cargo: Director del Centro de Identificación y Seguridad Digital UCI

Fecha: **21/05/19**