



Universidad de las Ciencias Informáticas

Facultad 1

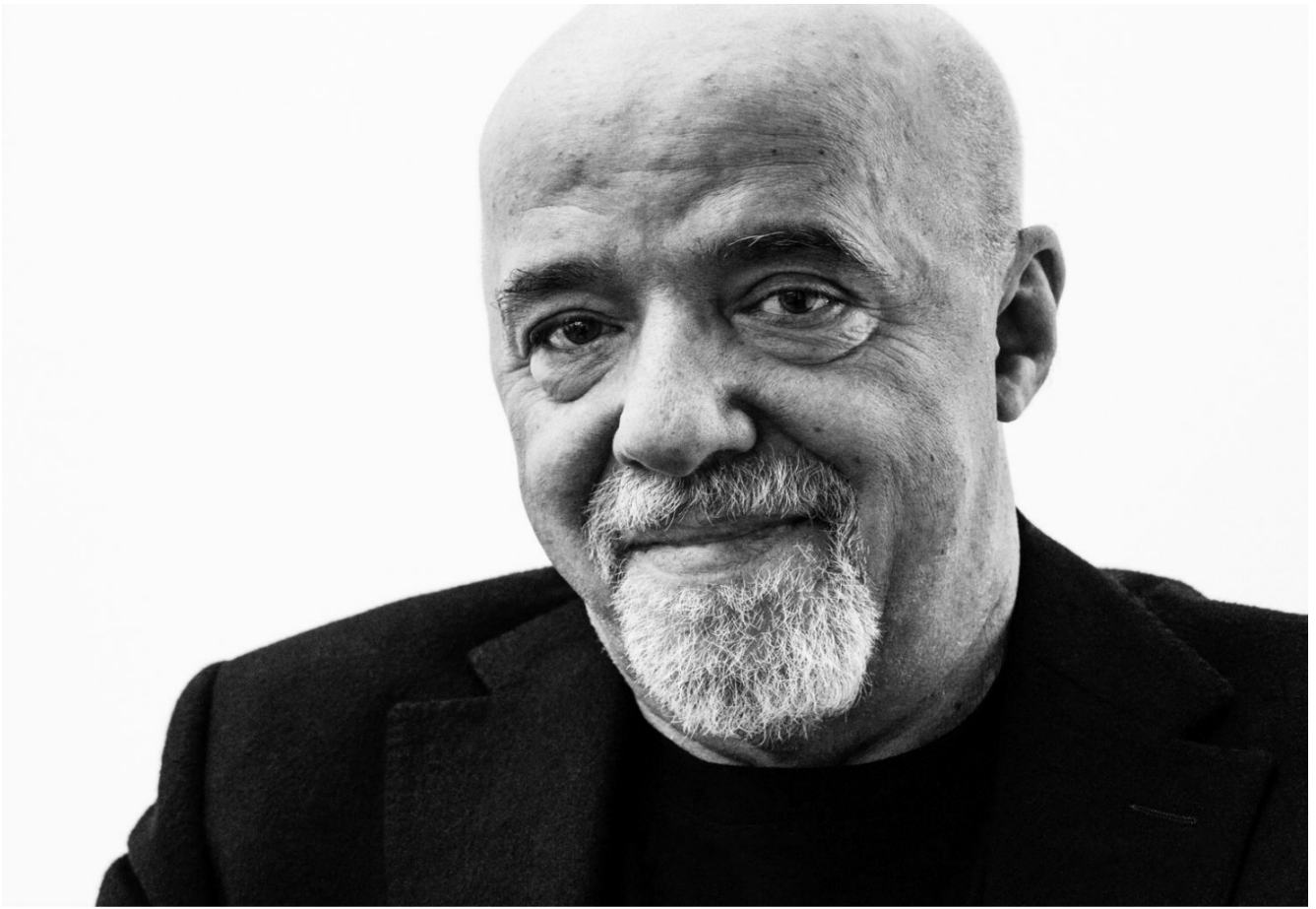
**“Módulo de Cotización para el Sistema de Control  
de la Militancia del Partido Comunista de Cuba.”**

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

Autor: Duriet Aguilera Alvarez

Tutor: Ing. Osay González Fuentes

**La Habana, 2019**



*"Es la posibilidad de que un sueño se haga realidad lo que hace la vida interesante."*

*Paulo Coelho*

## Agradecimientos

Bueno primero q todo quiero empezar agradeciendo a mi madre , gracias a ella soy lo que soy , gracias por ser la mejor mamá del mundo , por complacerme siempre y por luchar toda una vida por darle a tu hija lo mejor, por convertirte en mi compañera , amiga , confidente, gracias por brindarme las fuerzas cuando más las he necesitado , este logro te lo dedico a ti porque fuiste la persona que más confió en mi cuando al principio pensamos que todo iba a terminal mal , muchas gracias mami por ser como eres , te amo mucho. Quiero agradecerles a los mejores abuelos que la vida me pudo dar, recuerdo mi primer año cuando llegué a un punto de rechazo a la carrera que suspendía todos los exámenes y mi abuela cuando aprobé mi primera prueba de cálculo me llamo al apto llorando por teléfono dándome fuerzas y mostrándome que todo en la vida sucede por algún motivo simplemente hay que esforzarse y luchar por lo que uno quiere. Quiero agradecerle a Pà, mi abuelo, mi viejo que desde pequeña me crío inculcándome muchos valores que hoy en día y ya siendo una mujer hecha y derecha se lo agradezco infinitamente y gracias por complacer todos los caprichos de nieta mimada en todos estos años.

Quiero agradecerle a Perales, por brindarme todos los consejos de padre que tantas veces he necesitado, gracias por brindarme las lecciones de vida que en muchas ocasiones hemos platicado y gracias por ser también una de las personas que más me ayudaron y confiaron en mi a que yo realizara este logro tan importante en mi vida. Le agradezco a Sheila por estar presente aquí conmigo en este momento tan importante de mi vida, te quiero mucho y gracias por ser la hermana mayor que en varias ocasiones he necesitado.

Quiero agradecerle a mi tío querido que siempre estuvo presente en mi vida, ayudándome y apoyándome siempre en cualquier situación, gracias por siempre sentirte orgulloso de tu sobrina. A Anabel por formar parte de esta experiencia tan importante en mi vida.

Quiero agradecerle a mi papá, aunque no hemos tenido una relación muy cercana, espero que te sientas orgulloso de tu hija, y gracias por brindarme, aunque hayan sido pocas ocasiones tanta sabiduría que tienes y los consejos de vida que en muchas ocasiones me han servido y enseñado mucho.

Quiero agradecerle a mi novio, mi TITO, que hace 9 meses y 8 días comenzamos esta relación tan linda que llevamos, revolucionándome completamente y me has hecho en todo este tiempo la mujer más feliz el mundo. Gracias mi amor por la paciencia que has tenido en estos últimos meses que el estrés y la ansiedad

me pusieron en muchas ocasiones de malos genios y siempre tu haciéndome reír y contando sin resumen nuestras historias locas a todo el que apareciera. Te amo mucho mi amor.

Quiero agradecerle a una persona que en estos momentos no está aquí , porque expone ahora después que yo pero no importa claramente tenía que decirla en mis agradecimientos y quiero agradecerle por convertirse en la hermana que siempre quise tener , gracias por ser mi mejor amiga en todos estos años , gracias por todos los consejos aunque muchos no les hice caso por terca ,que me diste , gracias por ser mi confidente en muchas ocasiones , gracias por todo lo que hemos compartido en todo este tiempo te quiero muchísimo, igual quiero agradecerle al mejor cuñi Deniel , gracias por el mejor café que he tomado en la vida, gracias igual por aconsejarme cuando lo he necesitado , y gracias por el apoyo y la ayuda q me diste en el transcurso de esta etapa.

También quiero agradecerle, a mi hermano mayor como le digo, a mi mejor amigo Medero que lo extrañé mucho este año que no estuvo aquí, gracias por todos esos consejos de hermano mayor que en muchas ocasiones necesité y siempre estuviste para dárme los, gracias por confiar en mí siempre y por formar parte de mi vida, te quiero mucho.

Quiero agradecerle a la Yuli y Armando, los quiero mucho y me hubiese gustado muchísimo conocerlos desde mucho tiempo atrás, son magníficas personas, espero siempre poder contar con ustedes y seguir con esta bonita amistad.

A Rafi, por hacernos reír demasiado en todos estos años, por todas sus ocurrencias, por ser tan buen amigo.

A la pelui, Melissa por sus locuras y despistes, por ser tan única como solo ella sabe ser, gracias por esta bonita amistad que hemos formado, espero que, aunque nos distanciamos siempre te acuerdes de mí.

Asimismo quiero agradecer a todo el súper piquete que aunque los conocí este año, se han convertido en súper buenos amigos y los quiero un montón y gracias por las mejores fiestas que he cogido en mi vida, Lio, David, Enier, Leo, Lauri, el loki de Emilio , el surti , Adita.

A mis compañeros de aula, ale, que lo quiero muchísimo y siempre lo voy a mortificar cada vez que pueda, Bryan, Reimer, las lokis de Liana y Mayelin.

Quiero agradecerle a mi tutor Osay, gracias por ayudarnos tanto y por soportarme cuando te molestaba tanto por jabber y correo hasta tan tarde en la noche.

Al tribunal muchas gracias, al oponente osmel por esas maravillosas preguntas, muchas gracias.

Todos en general gracias por formar parte de mi vida y por acompañarme en esta experiencia única y tan importante en mi vida.

## **Dedicatoria**

Le dedico este triunfo a mi madre y mis abuelos.

## Declaración de autoría

Declaro por este medio que yo **Duriet Aguilera Alvarez**, con carné de identidad **96070406777** soy la autora principal del trabajo titulado “**Módulo de Cotización para el Sistema de Control de la Militancia del PCC**” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_\_ días del mes de \_\_\_junio\_\_\_\_\_ de \_\_\_\_\_ 2019\_

Autor: \_\_\_\_\_

Tutor: \_\_\_\_\_

## **Resumen**

El objetivo principal de esta investigación es desarrollar un módulo para el Sistema de Control de la Militancia (SICOM) del Partido Comunista de Cuba que permita gestionar y controlar la información referente a la cotización, a solicitud del Comité Central del Partido Comunista de Cuba. Surge ante la necesidad de contribuir a la gestión y control de los procesos gestionar cotización, gestionar cronogramas, gestionar Escalas PCC y UJC. Actualmente para ejecutar estos procesos, no cuentan con un módulo informático que trate la información descrita anteriormente, esto contribuye a que se vuelva muy engorroso elaborar y registrar informes referentes a la cotización a nivel municipal y provoca, además, que estos informes no se ajusten en su totalidad a la realidad objetiva, de ahí el objetivo de la presente investigación. Para la búsqueda y recopilación de información se emplearon los métodos de investigación: analítico sintético, análisis documental, modelación. Se llevó a cabo un estudio de diferentes sistemas que gestionan la información referente a la cotización en las organizaciones para adquirir una mayor visión a la hora de realizar la propuesta de solución. A partir de la realización de este trabajo se obtuvo como resultado un software que resuelve la problemática antes planteada y constituye a la vez un aporte tecnológico al Comité Central del Partido Comunista de Cuba.

**Palabras claves:** cotización, gestión de información, módulo web, Partido Comunista de Cuba.



## **ÍNDICE DE CONTENIDO**

INTRODUCCIÓN .....	1
CAPÍTULO 1. Fundamentación Teórica del Módulo de Cotización para el Sistema de Control de la Militancia del Partido Comunista de Cuba .....	5
<b>1.1 Conceptos asociados a la investigación</b> .....	5
<b>1.2 Estado del arte</b> .....	7
1.2.1 Principales sistemas de gestión de la información asociada a la cotización para organizaciones .....	7
<b>1.3 Metodología de desarrollo, Tecnologías y Lenguajes</b> .....	9
1.3.1 Metodología de desarrollo .....	10
1.3.2 Herramientas y Lenguaje .....	12
1.3.3 Servidor web .....	15
1.3.4 Entorno de desarrollo.....	15
1.3.5 Sistema Gestor de Bases de Datos .....	16
1.3.6 Marcos de Trabajo .....	16
<b>Conclusiones del capítulo</b> .....	17
CAPÍTULO 2. Características y Diseño del Módulo de Cotización para el Sistema de Control de la Militancia del Partido Comunista de Cuba .....	18
<b>2.1 Modelo de dominio</b> .....	18
2.1.1 Glosario de Términos.....	19
<b>2.2 Especificación de Requisitos</b> .....	19
2.2.1 Requisitos Funcionales.....	20
2.2.2 Requisitos No Funcionales .....	21
<b>2.3 Descripción de la solución propuesta</b> .....	22
2.3.1 Definición de los actores del sistema.....	22
2.3.2 Diagrama de Caso de Uso del Sistema .....	24

2.3.3 Descripción de los Casos de Uso del Sistema .....	24
<b>2.4 Arquitectura del Software .....</b>	<b>28</b>
2.4.1 Diagrama de Clase del Diseño .....	31
2.4.2 Patrones del Diseño.....	31
<b>Conclusiones del capítulo .....</b>	<b>35</b>
 CAPÍTULO 3. Implementación y prueba del Módulo de Cotización para el Sistema de Control de la Militancia del Partido Comunista de Cuba .....	 36
<b>3.1 Diagrama de Componentes .....</b>	<b>36</b>
<b>3.2 Diagrama de Despliegue .....</b>	<b>37</b>
<b>3.3 Estándares de Codificación.....</b>	<b>37</b>
<b>3.4 Pruebas del Software .....</b>	<b>38</b>
3.4.1 Estrategias de Pruebas.....	38
3.4.2 Pruebas de Aceptación.....	39
3.4.3 Pruebas de Unidad .....	40
3.4.4 Prueba de Integración.....	43
3.4.5 Prueba de Validación.....	44
<b>Conclusiones del capítulo .....</b>	<b>48</b>
 CONCLUSIONES GENERALES .....	 49
RECOMENDACIONES .....	50
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>51</b>
 ANEXOS .....	 54
Anexo 1 .....	54
Anexo 2 .....	68
Anexo 3.....	69
Anexo 4.....	70

## **Índice de Tablas**

---

<i>Tabla 1 Escala Salario. Cotización</i> .....	6
<i>Tabla 3 Características sistemas homólogos (Elaboración Propia)</i> .....	8
<i>Tabla 4 Requisitos Funcionales del Módulo de Cotización (Elaboración Propia)</i> .....	20
<i>Tabla 5 Actores del Sistema (Elaboración Propia)</i> .....	23
<i>Tabla 6 Descripción CU Gestionar Tipo de Cotización (Elaboración Propia)</i> .....	25
<i>Tabla 7 Diseño de casos de prueba del caso de uso Gestionar Tipo de cotización(Elaboración Propia)</i>	45
<i>Tabla 8 Descripción CU Gestionar Escala PCC(Elaboración Propia)</i> .....	54
<i>Tabla 9 Descripción CU. Gestionar cotización (Elaboración Propia)</i> .....	57
<i>Tabla 10 Descripción CU. Gestionar Cronograma (Elaboración Propia)</i> .....	60
<i>Tabla 11 Descripción CU. Gestionar Plan de Ingreso (Elaboración Propia)</i> .....	64

## Índice de Figuras

---

<i>Ilustración 1 Modelo de Dominio</i> .....	19
<i>Ilustración 2 Diagrama de Caso de Uso del Sistema</i> .....	24
<i>Ilustración 3 Funcionamiento de MTV Django (Rivera, 2010)</i> .....	29
<i>Ilustración 4 Estructura de directorios del módulo de cotización (Elaboración propia)</i> .....	30
<i>Ilustración 5 Diagrama de Clase del Diseño (Elaboración Propia)</i> .....	31
<i>Ilustración 6 Ejemplo Patrón Decorador</i> .....	32
<i>Ilustración 7 Ejemplo Patrón Decorador (Funcionalidad Plan de Ingreso)</i> .....	32
<i>Ilustración 8 Ejemplo Patrón Controlador (Funcionalidad Cotización)</i> .....	34
<i>ilustración 9 Ejemplo Patrón Experto</i> .....	34
<i>Ilustración 10 Diagrama de Componente (Elaboración Propia)</i> .....	36
<i>Ilustración 11 Diagrama de Despliegue (Elaboración Propia)</i> .....	37
<i>ilustración 12 Resultados Prueba de Aceptación</i> .....	40
<i>ilustración 13 Grafo de flujo de datos del método CotizacionCreate</i> .....	41
<i>ilustración 14 Diagrama Prueba de Integración</i> .....	44
<i>Ilustración 15 Resultado de las pruebas de validación (Elaboración Propia)</i> .....	47
<i>ilustración 16 Método Caja Blanca CotizacionCreate (Elaboración Propia)</i> .....	68
<i>ilustración 17 Pruebas de Unidad. PyUnit</i> .....	69

# INTRODUCCIÓN

---

## INTRODUCCIÓN

Como una de las organizaciones de mayor prestigio en Cuba se encuentra el Partido Comunista de Cuba (PCC). En su condición de Partido único de la nación cubana se estructura orgánicamente y desarrolla su vida interna sobre la base de la observación más rigurosa del principio leninista del centralismo democrático que conjuga una disciplina estricta y consciente con la más amplia democracia interna. El ejercicio de la dirección colectiva, la responsabilidad individual y la práctica de la crítica y la autocrítica ante los propios errores, garantiza la pureza y la cohesión de sus filas. (Granma, 2016)

Los organismos y organizaciones de base se guían en su actividad por los Estatutos y Reglamentos y por los acuerdos, resoluciones, orientaciones e indicaciones de sus órganos y organismos superiores, así como por las orientaciones y sugerencias del primer secretario del Comité Central, compañero Raúl Castro Ruz. Como organización, el PCC, se encuentra inmerso en el proceso de informatización de la sociedad cubana, y a raíz de esto ha decidido informatizar los procesos que se llevan a cabo. Para ello se ha solicitado a la Universidad de las Ciencias Informáticas (UCI) el desarrollo de un sistema que gestione diversos procesos que se realizan en la organización entre los que se puede mencionar: Registro de sanciones, Registros de Traslados, balances de los núcleos y de reporte, la gestión y control de la cotización, así como la gestión de las reuniones.

Actualmente el Centro de Identificación y Seguridad Digital (CISED) de la UCI, se encuentra desarrollando un Sistema para el Control de los Militantes del PCC (SICOM), a solicitud del Departamento de Funcionamiento del Comité Central del Partido Comunista de Cuba (CCPCC). El sistema SICOM tiene como objetivo llevar la gestión y control de los militantes del PCC.

Este sistema se encuentra en explotación desde hace dos años, y dentro de las funcionalidades que se han implementado se encuentran: el registro de militantes de la UJC y PCC, el registro de núcleos del PCC, Comités Primarios, Municipales y Provinciales, el registro de sanciones y traslados. También se han implementado módulos para la gestión de balances de los núcleos, y de reporte.

En estos momentos cuando un cotizador de una organización de base acude a la sede municipal del PCC a realizar la entrega del dinero recogido, la persona encargada de recibirlo debe chequear el listado de cotizantes y llevar un registro de los militantes que pagaron o no. En caso afirmativo registrar el salario recibido en ese mes y cuanto cotizó cada militante. Todo este proceso se realiza indistintamente en todos los municipios de forma manual, es decir, la búsqueda de la información es mediante hojas Excel, libros

# INTRODUCCIÓN

contables, plantillas lo que resulta lento y trabajoso, además existe dependencia del área de funcionamiento para saber si hubo alteración en las plantillas de los núcleos, por ejemplo: fue disuelto un núcleo, si hubo nuevas incorporaciones, traslados o bajas de un militante hacia otro núcleo. Además, al tratar con información en formato duro en ocasiones provoca duplicidad en los documentos, errores y en algunos casos pérdida de información debido a que se extravían documentos.

Un militante puede dejar de pagar su cuota uno o varios meses y pagar en un solo mes todas las cuotas que tiene atrasadas. Regularmente este proceso consiste en buscar las cuotas que el militante ha dejado de pagar y calcular el monto total de la deuda, lo cual puede generar errores de cálculo que puedan conllevar a problemas financieros, asimismo la búsqueda demanda abundante tiempo y trabajo. En los comités municipales al no existir un registro real de cotizantes provoca que el control de todo el proceso resulte difícil ya que no permite comprobar la realización correcta de todas las acciones referentes a la cotización como, por ejemplo: verificar que el militante que le correspondía cotizar, efectuó el pago de la cuota, confirmar la entrega del slip bancario al banco y chequear que no ocurran ilegalidades en la recogida de la cotización.

A partir de la situación problemática descrita anteriormente se propone como **problema a resolver**: ¿Cómo contribuir a la gestión y control de la información asociada a la cotización de los militantes del Partido Comunista de Cuba?

Se plantea como **objeto de estudio** de la presente investigación: proceso de gestión y control de la información en organizaciones políticas.

Dentro del objeto de estudio se enmarca como **campo de acción**: proceso de gestión y control de la información asociada a la cotización haciendo uso de sistemas informáticos.

El **objetivo general** de la investigación es: desarrollar el módulo de cotización para el Sistema de Control de la Militancia del Partido Comunista de Cuba que permita contribuir a la gestión y control de la información asociada a la cotización de los militantes utilizando tecnologías libres.

Para darle cumplimiento al objetivo planteado se responderán las siguientes **preguntas científicas**:

- ¿Cuáles son los supuestos teóricos que sustentan el desarrollo y la utilización del módulo de cotización para el Sistema de Control de la Militancia del PCC (SICOM)?
- ¿Qué aspectos deben tenerse en cuenta para realizar el análisis y diseño del módulo de cotización para el Sistema de Control de la Militancia del PCC (SICOM)?

# INTRODUCCIÓN

- ¿Cómo implementar, a partir del análisis y diseño realizado, el módulo de cotización para el Sistema de Control de la Militancia del PCC (SICOM)?
- ¿Qué resultados se obtendrán al validar el módulo de cotización para el Sistema de Control de la Militancia del PCC (SICOM)?

Acorde a las preguntas científicas ya expuestas, se trazan los siguientes **objetivos específicos**:

- Elaborar el marco teórico-metodológico de la investigación referente al desarrollo y utilización del módulo de cotización para el Sistema de Control de la Militancia del PCC (SICOM).
- Diseñar el módulo de cotización para el Sistema de Control de la Militancia del PCC (SICOM).
- Implementar el módulo de cotización para el Sistema de Control de la Militancia del PCC (SICOM).
- Validar las funcionalidades del módulo de cotización para el Sistema de Control de la Militancia del PCC (SICOM).

Las **tareas de investigación** a desarrollar para dar respuesta a los objetivos específicos son:

- Sistematización de los principales referentes teóricos que permiten la elaboración del marco teórico de la investigación.
- Análisis valorativo de sistemas internacionales existentes que gestionan la información referente a la cotización en las organizaciones.
- Selección de las herramientas y tecnologías utilizadas en la implementación del Sistema de Control de la Militancia del PCC (SICOM), así como la metodología implementada.
- Identificación de los requerimientos funcionales y no funcionales del módulo de cotización para el Sistema de Control de la Militancia del PCC (SICOM).
- Diseño del módulo de cotización para el Sistema de Control de la Militancia del PCC (SICOM).
- Implementación del módulo de cotización para el Sistema de la Militancia del PCC (SICOM).
- Validación del módulo de cotización para el Sistema de Control de la Militancia del PCC (SICOM).

Para el desarrollo de la investigación se emplearon los siguientes **métodos de la investigación científica**:

## **Métodos Teóricos:**

- Analítico-Sintético: se evidencia en el análisis de los elementos de la situación problemática relacionándolos entre sí y vinculándolos como un todo para descubrir relaciones y características generales entre los elementos de la realidad. Se emplea para analizar la información encontrada

# INTRODUCCIÓN

sobre los elementos que intervienen en la gestión de la información, específicamente del Partido Comunista de Cuba, y extraer ejemplos generales y específicos, tratando de establecer relación entre los diferentes criterios que puedan incidir y que tributen a la correcta gestión de la información.

- Modelación: Permite la representación estructural, relaciones internas y características de la solución propuesta, haciendo uso de diagramas.

## **Métodos empíricos:**

Análisis documental: se evidencia en el estudio de los documentos que establecen procedimientos para la gestión de los procesos de la vida interna del Partido Comunista de Cuba como son los reglamentos y estatutos de esta organización.

## **Estructura del documento**

El trabajo está estructurado en tres capítulos que abarcan todo el proceso para el desarrollo de la solución informática.

**Capítulo 1: Fundamentación teórica:** se expone el estado del arte del objeto de estudio de la investigación y se definen conceptos fundamentales asociados al dominio del problema y se definen las metodologías, tecnologías y herramientas a utilizar durante el desarrollo de la solución.

**Capítulo 2: Descripción de la propuesta de solución:** se describe la propuesta de solución, los requisitos funcionales y no funcionales, así como los procesos y actores que intervienen en el proceso de control de la cotización de los militantes del Partido. También, se describen los patrones de diseño y el patrón de arquitectura utilizados por el marco de trabajo, así como el modelo de datos definido para el registro de la información a gestionar.

**Capítulo 3: Implementación y validación de la solución:** se abordan las fases de implementación y despliegue de la propuesta de solución teniendo en cuenta las técnicas de codificación empleadas, así como la realización de pruebas para evaluar y verificar la calidad del sistema, tales como las pruebas de unidad, integración, validación y de aceptación para satisfacer las necesidades del cliente.

## **Posibles resultados:**

Con la realización de la presente investigación, se pretende obtener un módulo de cotización para el SICOM que gestione y controle la información referente al pago de la cotización de los militantes del PCC.



# CAPÍTULO 1

## **CAPÍTULO 1. Fundamentación Teórica del Módulo de Cotización para el Sistema de Control de la Militancia del Partido Comunista de Cuba**

En este capítulo se abordan los principales conceptos relacionados con la gestión de información asociada a la cotización, para lograr una mejor comprensión. Se presentan las tecnologías, metodologías y herramientas a emplear en el proceso de desarrollo.

### **1.1 Conceptos asociados a la investigación.**

En el presente epígrafe se describen un conjunto de conceptos asociados a la gestión de información y al proceso de cotización del Partido Comunista de Cuba con el objetivo de facilitar la comprensión del objeto de estudio de la investigación.

#### **La gestión de la información en las organizaciones**

La Gestión de Información es un proceso estratégico que tiene lugar en una organización de cualquier tipo (incluidas las comunidades y otras entidades de carácter social). Es un proceso que abarca todos los procesos y actividades de esa organización y sus componentes por lo que tiene una estrecha relación con el sistema que lo rige y participan en él diferentes componentes. (Duarte, 2016)

La gestión de información se define como el proceso mediante el cual se obtienen, despliegan o utilizan recursos básicos (económicos, físicos, humanos, materiales) para manejar información dentro y 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. (Rodríguez, 2015)

La Gestión de Información se concibe como un proceso estratégico de planificación, organización, dirección y control, de forma eficaz y eficiente, de las estrategias, recursos, procesos, sistemas, productos, servicios, y demás capacidades informacionales existentes en una organización o en la sociedad, con el objetivo de mejorar el desempeño, la toma de decisiones, la adaptación al cambio, y la creación de fortalezas y ventajas competitivas por parte de los individuos, las organizaciones y la administración pública. (Cruz, 2015)

A partir del análisis de los conceptos referentes a la Gestión de la Información expuestos anteriormente se decide utilizar el brindado por el Dr. Yunier Rodríguez Cruz , pues define todos los procesos que forman parte de la gestión de la información ya que se puede vincular de forma eficiente a la gestión de la información vinculada a la cotización en el módulo a desarrollar para añadir al Sistema para el Control de la

# CAPÍTULO 1

Militancia del PCC (SICOM) y expone los objetivos que se pretenden lograr con la solución del problema planteado, como la optimización de tiempo y esfuerzo que se realiza en el momento de realizar el proceso.

## **Partido Comunista de Cuba**

El artículo No. 5 de la Constitución de la República define al Partido Comunista de Cuba como la fuerza dirigente superior de la sociedad y del Estado que organiza y orienta los esfuerzos comunes hacia la construcción del socialismo. (Cuba, 2019)

## **Cotización**

La cotización es el aporte económico individual de los miembros de una organización para contribuir a sufragar los gastos de la misma en el cumplimiento de su misión, lo cual incide en su formación, responsabilidad, disciplina, colectivismo y sentido de pertenencia. (PCC, 1999)

La cotización de los militantes es un principio organizativo del partido. La cuota a pagar cada mes se determina sobre la base de los ingresos percibidos por los militantes en ese período, incluidas las diversas fuentes o vías de que provengan. (PCC, 1999)

El monto de los ingresos percibidos se obtendrá por parte de la organización de base mediante información que, personalmente, brinde cada uno de los miembros del partido mensualmente. Los militantes pagarán la cuota atendiendo a la siguiente escala (PCC, 1999):

*TABLA 1 ESCALA SALARIO. COTIZACIÓN*

Escala(pesos)	Escala(pesos)	Importe
0	0	0.20 centavos
1	125	0.50 centavos
126	199	2 pesos
200	299	4 pesos
300	399	8 pesos
400	499	15 pesos
500	2999	3%
3000	4999	90
5000	6999	120
7000	8999	140

# CAPÍTULO 1

9000	10 999	150
11 000	12 999	200
13 000	15 999	220
16 000 +	-	250

## 1.2 Estado del arte

### 1.2.1 Principales sistemas de gestión de la información asociada a la cotización para organizaciones

En la actualidad existen diversas soluciones informáticas que se dedican a gestionar la información vinculada a la cotización en las organizaciones entre ellas se encuentran:

#### ➤ Sistema Red Internet

El sistema Red desarrollado por el Ministerio de empleo y Seguridad Social de España, es el medio con el cual las empresas y la Tesorería General de la Seguridad Social (TGSS<sup>1</sup>) interactúan para el intercambio de información y documentos a través de internet. Entre los principales trámites que se realizan se encuentra: Cotización, plataforma en el cual se gestiona el proceso recaudatorio de las cotizaciones de la Seguridad Social, así como el trámite de los pagos y el control de los Seguros Sociales.

Por medio del Sistema RED se puede entrar en contacto directo con la TGSS que, gracias a los medios tecnológicos y elementos de seguridad necesarios, les permite el acceso a datos de empresa y trabajadores, así como la remisión de documentos de cotización y afiliación y partes médicos.

Para acceder al Sistema RED, es a través de un certificado digital y se debe disponer de un autorizado RED. El sistema permite realizar las comunicaciones de afiliación y cotización de trabajadores a la Seguridad Social de manera telemática.

#### ➤ Siltra (Sistema de Liquidación Directa)

**Siltra** desarrollada por el Ministerio de empleo y Seguridad Social de España, es una aplicación de escritorio multiplataforma desarrollada en JAVA que permite el intercambio de ficheros de cotización (formato XML),

---

<sup>1</sup> TGSS: es un Servicio Común de la Seguridad Social, tutelado por el Ministerio de Empleo y Seguridad Social del Gobierno de España.

# CAPÍTULO 1

afiliación e INSS<sup>2</sup> entre el usuario y la TGSS mediante certificado Digital, en el Sistema de Liquidación Directa. (Subdirección General de Afiliación, 2017)

Este sistema mejora la impresión de documentos de cálculo, recibos de liquidación de cotizaciones, se minimizan los errores relativos a la aplicación de las reglas de cotización, contrastando los datos con carácter previo a la liquidación y se trabaja desde un sistema completamente informatizado, y por ello, se elimina la necesidad de acudir físicamente a las oficinas de la TGSS y se evita el soporte papel.

Ambos sistemas presentan aspectos positivos que establecen relación con el módulo a desarrollar para el Sistema de Control de la Militancia del PCC (SICOM), entre los que se puede mencionar:

*Tabla 2 Características sistemas homólogos (Elaboración Propia)*

<b>Sistema Red Internet</b>	<b>Sistema SILTRA</b>
El cálculo de la liquidación se hace en base a la información suministrada por el usuario.	La Tesorería General de la Seguridad Social es la que calcula la liquidación a nivel de trabajador con la información de sus bases de datos.
El cálculo de la cotización se realiza a cada trabajador.	Simplifica la forma de cotizar la Seguridad Social y reduce las cargas administrativas.
Las liquidaciones que no eran calculadas por el sistema se ingresaban en formato papel.	Mediante este sistema la transmisión es exclusivamente a través de medios electrónicos.

Atendiendo a las particularidades que presentan ambos sistemas:

## **Sistema Red Internet**

- No se puede acceder al código fuente de la aplicación ya que es privativo.

---

<sup>2</sup> INSS: Instituto Nacional de la Seguridad Social en España.

# CAPÍTULO 1

- Para acceder al sistema es imprescindible disponer de un certificado electrónico válido que garantice la seguridad y confidencialidad de todas las actuaciones y se debe disponer de un autorizado RED.

## **Sistema SILTRA**

- No se puede acceder al código fuente ya que es privativo
- El cálculo de la cotización se hace a nivel de Código de Cuenta Cotización (CCC).<sup>3</sup>

Luego de realizado el análisis de los sistemas informáticos internacionales que se utilizan para gestionar la cotización en las organizaciones se concluye que ambos presentan características que no permiten la integración a SICOM, para ello se ratifica la necesidad de implementar un nuevo módulo que cumpla todos los requerimientos que demanda el cliente.

## **Módulo de Cotización para el Sistema de Control de la Militancia del PCC (SICOM):**

- Se puede acceder al código fuente ya que es libre.
- El cálculo de la cotización es a través del propio sistema.
- Como el módulo debe integrarse al Sistema para el Control de la Militancia del PCC (SICOM), se seleccionaron, la misma metodología, tecnologías y lenguajes que se emplearon para desarrollar el sistema informático.

### **1.3 Metodología de desarrollo, Tecnologías y Lenguajes**

Para el desarrollo del Módulo de Cotización se utilizarán las tecnologías, metodología y lenguajes que se emplearon en el desarrollo del sistema informático SICOM (Sistema de Control de la Militancia) ya que esta solución se añade al sistema creado para facilitar la realización de los procesos. Las mismas se definen como:

---

<sup>3</sup> Código de Cuenta de Cotización(CCC): es un código numérico único de once dígitos, sin intercalar guiones barras o puntos, cuyos dos primeros dígitos corresponden a la provincia donde se encuentren y los últimos dos serían un numero de control del mismo. Este es número asignado por el Seguro Social y obligatorio para cada empresa y/o responsables de facilitar el pago de cuotas para el Seguro Social. Con dicho código, el Seguro Social podrá identificar a la empresa y encargarse de llevar un control y hacer que se ejecuten sus obligaciones.

# CAPÍTULO 1

## 1.3.1 Metodología de desarrollo

Una metodología de desarrollo de software es un proceso o conjunto de procedimientos, técnicas y documentación que permiten a los desarrolladores guiar y ejecutar el proyecto con el objetivo de crear nuevas aplicaciones de calidad que satisfagan las expectativas del cliente. (Ruíz, y otros, 2017). Pressman define la metodología como “un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo de sistemas computacionales”.

En el campo de desarrollo de software existen dos tipos de metodologías, las denominadas tradicionales (formales) y las ágiles.

Las primeras son un tanto rígidas, exigen una documentación exhaustiva y se centran en cumplir con el plan del proyecto definido totalmente en la fase inicial del desarrollo del mismo; mientras que la segunda enfatiza el esfuerzo en la capacidad de respuesta a los cambios, las habilidades del equipo y mantener una buena relación con el usuario. (Ruíz, y otros, 2017)

Para la implementación del módulo y según lo expuesto anteriormente, se decide utilizar una metodología ágil ya que brinda facilidades en cuanto a la documentación y el entorno cambiante que tiene el proyecto. A continuación, se describe la metodología seleccionada para guiar el proceso de desarrollo de software de la presente investigación:

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (**AUP**) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo:

- Desarrollo Dirigido por Pruebas (test driven development - TDD en inglés)
- Modelado ágil
- Gestión de Cambios ágil
- Refactorización de Base de Datos para mejorar la productividad.

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 decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI.

**AUP-UCI** es una variación de la metodología AUP. En aras de aumentar la calidad del software que se produce, esta metodología se apoya en el Modelo CMMI-DEV v1.3. (por sus siglas en inglés *Capability*

# CAPÍTULO 1

*Maturity Model Integration Development*, Integración de Modelos de Madurez de Capacidades para Desarrollo). El mismo constituye una guía para aplicar las mejores prácticas y obtener un producto o servicio con calidad en una entidad desarrolladora. (Sánchez, 2014)

## Fases de la metodología:

1. **Inicio:** El objetivo de esta fase es llevar a cabo las actividades relacionadas con la planeación del proyecto. Se realiza un estudio inicial de la organización que actúa como cliente y se obtiene información acerca del alcance del proyecto, se realizan estimaciones de tiempo y esfuerzo, y finalmente se decide si se ejecuta o no.
2. **Ejecución:** En esta etapa se ejecutan las actividades requeridas para desarrollar el software. Durante el desarrollo se modela el negocio, se obtienen los requisitos, se elabora la arquitectura y el diseño, se implementa y se libera el producto.
3. **Cierre:** En esta fase se analizan tanto los resultados del proyecto como su ejecución y se llevan a cabo las actividades formales de cierre de proyecto.

La metodología **AUP-UCI** se ajusta al ambiente que presenta el negocio que se desea informatizar y da la posibilidad al cliente de siempre acompañar al equipo de desarrollo para convenir los requisitos y así poder implementarlos. También, es una de las que mayor prestigio presenta hoy día entre los desarrolladores de la UCI, pues fue aprobada cuando la institución se sometió al proceso de certificación CMMI nivel 2 para el desarrollo de software.

La metodología **AUP-UCI** propone 7 disciplinas:

1. **Modelado de negocio:** es la disciplina destinada a comprender los procesos de negocio de una organización. Se comprende cómo funciona el negocio que se desea informatizar para tener garantías de que el software desarrollado va a cumplir su propósito. Para modelar el negocio se proponen las siguientes variantes
  - Casos de Uso del Negocio (CUN).
  - Descripción de Proceso de Negocio (DPN).
  - Modelo Conceptual (MC).

A partir de las variantes anteriores se condicionan cuatro escenarios para modelar el sistema en la disciplina Requisitos

2. **Requisitos:** El esfuerzo principal en la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos

# CAPÍTULO 1

funcionales y no funcionales del producto. Existen tres formas de encapsular los requisitos [Casos de Uso del Sistema (CUS), Historias de usuario (HU) y Descripción de requisitos por proceso (DRP)], agrupados en cuatro escenarios condicionados por el Modelado de negocio.

Para la realización del proyecto se definió en la disciplina Requisitos utilizar el escenario No 2, que expone modelar CUS (Caso de Uso del Sistema) ya que el negocio fue modelado a través de modelo conceptual.

3. **Análisis y diseño:** En esta disciplina, si se considera necesario, los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos, y una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo su arquitectura). Además, en esta disciplina se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Los modelos desarrollados son más formales y específicos que el de análisis.
4. **Implementación:** En la implementación, a partir de los resultados del análisis y diseño se construye el sistema.
5. **Pruebas internas:** En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. Se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posible componentes de prueba ejecutables para automatizar las pruebas.
6. **Pruebas de liberación:** Pruebas diseñadas y ejecutadas por una entidad certificadora de la calidad externa, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.
7. **Pruebas de Aceptación:** Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido. (Sánchez, 2014)

## 1.3.2 Herramientas y Lenguaje

Las herramientas son aplicaciones, programas o meramente instrucciones que ofrecen la posibilidad de realizar determinadas funcionalidades con disímiles propósitos. De otro modo, las tecnologías son el conjunto de conocimientos técnicos, ordenados científicamente que permiten diseñar y crear bienes y servicios (Hernández, 2006). A continuación, se muestran todas las herramientas, tecnologías y lenguajes



# CAPÍTULO 1

utilizadas en el desarrollo del Módulo para la gestión de la cotización de los Militantes del Partido Comunista de Cuba.

## **Lenguaje de modelado**

Un lenguaje de modelado es un lenguaje artificial para expresar modelos, habitualmente estos se muestran en forma de diagramas por comodidad. Los lenguajes de modelado, al igual que los lenguajes naturales, poseen un conjunto de palabras que existen en el lenguaje (léxico); y un conjunto de reglas que nos dicen cómo se pueden combinar dichas palabras para componer "frases" que tengan sentido (sintaxis). Estas "palabras" de los lenguajes de modelado no se transmiten mediante texto o sonido como en el caso de los lenguajes naturales, sino que, habitualmente, lo hacen en forma de iconos o dibujos para facilitar la visualización formal de los conceptos, que son abstractos. (Patricia, y otros, 2015)

## **UML 2.0**

Se define como un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema. Proporciona una forma estándar de diagramar planos de un sistema, abarcando las partes conceptuales (funciones del sistema, y en principio también procesos industriales), y los objetos concretos (clases escritas en lenguajes de programación específico, esquemas de bases de datos, componentes de software reutilizables). (Patricia, y otros, 2015)

## **Herramienta de Modelado**

Dentro de las principales herramientas para el modelado se encuentran herramientas de Ingeniería de Software Asistida por Computadora (*CASE* por sus siglas en inglés) definidas como "Herramientas individuales para ayudar al desarrollador de software o administrador de proyecto durante una o más fases del desarrollo de software (o mantenimiento)." ( Universidad Politécnica de Valencia, 2016)

## **Visual Paradigm**

Es una herramienta CASE que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, implementación y pruebas. Ayuda a una rápida construcción de aplicaciones con mejor calidad y a un menor costo. Permite construir diagramas de diversos tipos, código inverso, generar código desde diagramas y generar documentación. (Cabrera González, y otros, 2012)

Para la presente investigación se utiliza esta herramienta en su versión 8.0 por ser una herramienta de software libre que permite realizar ingeniería tanto inversa como directa.

# CAPÍTULO 1

## Herramienta para el control de versiones

Un sistema de control de versiones es una herramienta capaz de registrar todos los cambios que se realizan en uno o más proyectos, guardando a su vez versiones anteriores del proyecto, a las que podemos acudir en caso de haber cometido un error o al no funcionar de la forma correcta. (acensTechnologies, 2015).

Para garantizar el control de versiones en el sistema se emplea el repositorio *GitLab*, específicamente el cliente **Git** en su versión 2.8.

## Lenguajes de programación

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo. (Rolando, y otros, 2016)

A continuación, se muestra el lenguaje de programación utilizado para el desarrollo del módulo de cotización de los militantes del PCC:

### Python

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para *scripting* y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas. (Rolando, y otros, 2016)

Las ventajas de usar Python son:

- **Facilidad de uso.** Una persona puede empezar a hacer programas sencillos en Python en muy poco tiempo. A esto contribuye la gestión automática de memoria o las operaciones sencillas de lectura y escritura, a diferencia de otros lenguajes como C.
- **Legibilidad del código.** La estructura del código es bastante natural y promueve una forma de escribir que facilita su lectura. Esta es una ventaja importante frente a lenguajes dirigidos al mismo sector.
- **Facilidad de uso en dispositivos.** Algunas plataformas como Raspberry Pi están basadas en Python.
- **Facilidad de escritura de código asíncrono.** Los lenguajes diseñados antes de que las plataformas multiprocesador (o multinúcleo) estuvieran tan generalizadas suelen tener estructuras bastante

# CAPÍTULO 1

complicadas para mantener distintos hilos de ejecución; en Python el código asíncrono es bastante sencillo de gestionar.

- **Abundancia de bibliotecas.** Hay muchas bibliotecas disponibles para extender la funcionalidad básica de Python a cualquier campo.
- **Gran base de usuarios.** Esto hace que exista mucho código disponible en internet y que los foros de usuarios sean bastante activos, por lo que es fácil encontrar ayuda cuando se necesita.

Para el desarrollo de este módulo web se utilizó Python en su versión 3.5.1.

### 1.3.3 Servidor web

Un servidor Web es un programa que utiliza el protocolo de transferencia de hipertexto, HTTP (*Hypertext Transfer Protocol*), para servir los archivos que forman páginas Web a los usuarios, en respuesta a sus solicitudes, que son reenviados por los clientes HTTP de sus computadoras. Las computadoras y los dispositivos dedicados también, pueden denominarse servidores Web.(Rouse, 2015)

### 1.3.4 Entorno de desarrollo

Un Entorno de Desarrollo Integrado, o bien conocido como IDE (por sus siglas en inglés Integrated Development Environment), es una aplicación de software, que proporciona servicios integrales para facilitarle al programador de computadora el desarrollo de software. Normalmente, un IDE consiste de un editor de código fuente, herramientas de construcción automáticas y un depurador. La mayoría de los IDEs tienen autocompletado inteligente de código. (Suárez, 2016)

#### PyCharm IDE

*PyCharm* es uno de los entornos de desarrollo más completos para Python. Es parte de la suite de herramientas de programación ofrecidas por JetBrains, que cuenta con entornos para construir código en distintos idiomas como PHP y Ruby. (Python, 2019)

#### Ventajas de PyCharm

Trabajar con *PyCharm* tiene ventajas básicas (similares a las ofrecidas por otros IDE) pero también algunas específicas a las cuales debe su popularidad. Es así que *PyCharm* tiene un editor inteligente, que permite completar código con algunos atajos de teclado.

Una de las características notables de *PyCharm* es la posibilidad que tiene de refactorizar el código, que en términos generales, significa modificar el código sin comprometer la ejecución del mismo.

Para llevar a cabo la implementación del módulo web se hará uso de PyCharm en su versión 2017.2.2

# CAPÍTULO 1

## 1.3.5 Sistema Gestor de Bases de Datos

Es una herramienta de propósito general útil para estructurar, almacenar y controlar los datos ofreciendo interfaces de acceso a la base de datos. Tareas fundamentales que desempeñan estos sistemas hacen referencia a la seguridad de acceso a los datos, al mantenimiento de la integridad de los datos, a mecanismos de recuperación debidos a fallos físicos y lógicos, al control de concurrencia en el momento de acceder a los datos y a la eficiencia del sistema evaluada, generalmente, en términos del tiempo de respuesta a las consultas de los usuarios. (Rolando, y otros, 2017)

### PostgreSQL

“PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado”.(PostgreSQL, 2017)

“PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando”. (Rolando, y otros, 2017)

Para desarrollar el Sistema web para la gestión de la cotización de los militantes del PCC se va a utilizar como Sistema Gestor de Base de Dato PostgreSQL en su versión 9.4.1.

## 1.3.6 Marcos de Trabajo

Un marco de trabajo o *framework* en inglés, está diseñado para apoyar el desarrollo de aplicaciones web. Es un software o conjunto de librerías que tratan de facilitar aquellas actividades comunes que son realizadas durante el desarrollo de una aplicación web, como, por ejemplo: acceso a la base de datos, uso de plantillas, manejo de sesiones; además, promueve la reutilización de código.

### Django

Django es un *framework* de código libre que permite el desarrollo web ágil y utilizando el menor número de código posible; requiere también de un nivel de conocimientos básicos del lenguaje de programación Python. (Cabrera, 2016)

Para el desarrollo del sistema web se decidió usar Django en su versión 1.11.8.

# CAPÍTULO 1

## Conclusiones del capítulo

- El análisis de los conceptos asociados al objeto y campo de la investigación permitió obtener un conjunto de elementos fundamentales para desarrollar una propuesta de solución dirigida a la informatización del proceso gestión y control de la cotización de los militantes del PCC.
- Luego de realizado un estudio de los sistemas homólogos existentes, se observaron algunas particularidades que no permitían la integración al Sistema de Control de la Militancia del PCC (SICOM); por tal motivo se ratifica la necesidad de implementar un nuevo módulo para el sistema informático que gestione y controle la información asociada a la cotización en la organización.
- Las herramientas seleccionadas facilitan el desarrollo en cuanto al modelado y la implementación del módulo, de forma que se logre una correcta integración con el Sistema de Control de la Militancia del PCC (SICOM), guiada por la metodología de desarrollo de software *AUP* versión UCI.

## CAPÍTULO 2

### **CAPÍTULO 2. Características y Diseño del Módulo de Cotización para el Sistema de Control de la Militancia del Partido Comunista de Cuba**

En el presente capítulo se realiza la descripción de la propuesta de solución. Además, se exponen las principales características del módulo, y se definen los requisitos funcionales y no funcionales necesarios para lograr que el módulo de cotización para el Sistema de Control de la Militancia del PCC funcione correctamente. Se identifican las actividades que se informatizarán y se construyen los artefactos correspondientes al análisis y diseño. Se caracteriza la arquitectura y patrones de diseño implementados en el marco de trabajo.

#### **2.1 Modelo de dominio**

Un modelo de dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. Este modelo se describe mediante diagramas de clase, que se van refinando con los detalles y sirven de base para el diseño de la capa lógica o de negocio del sistema y, posteriormente, para la construcción de los programas orientados a objetos. Estos diagramas muestran las clases del dominio y las relaciones entre ellas a través de asociaciones. (Casadiego, 2017)

El proyecto en el cual se encuentra enmarcada la gestión y control de la cotización de los militantes del PCC constituye un producto personalizable y que se encuentra sujeto a cambios constantes, para ello se determinó que realizar un modelo de dominio constituiría una forma correcta de manejar los cambios que se producen en el mismo. Para una mejor comprensión de los conceptos asociados al funcionamiento del módulo se mostrará a continuación el glosario de términos y el modelo de dominio.

## CAPÍTULO 2

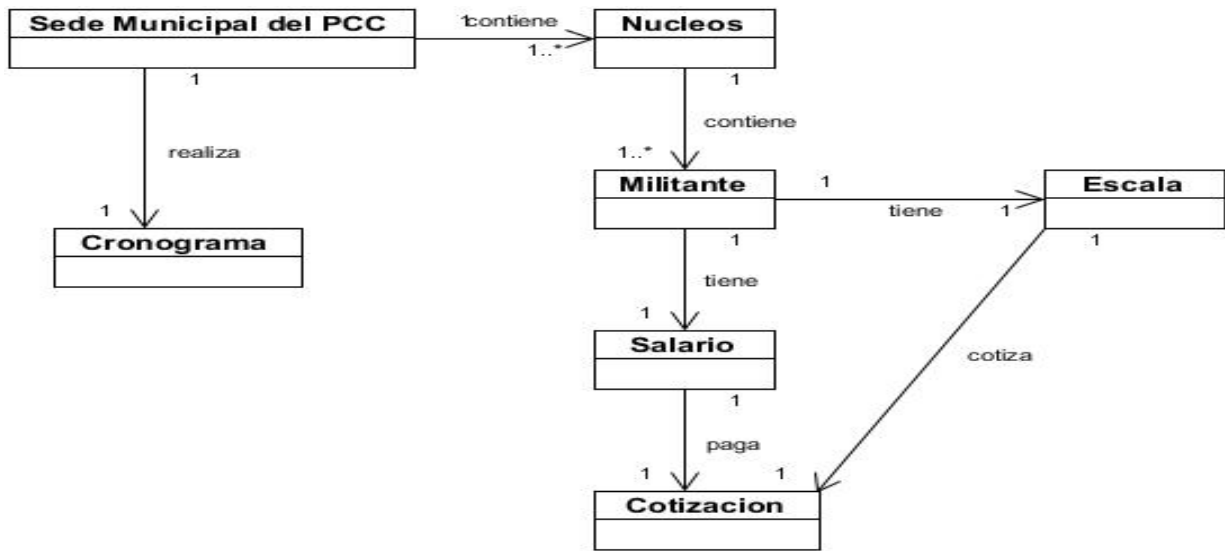


ILUSTRACIÓN 1 MODELO DE DOMINIO

### 2.1.1 Glosario de Términos

**Sede Municipal del PCC:** entidad que representa al Comité Central del PCC a nivel municipal, es el encargado de manejar toda la información de los militantes del PCC proveniente de cada núcleo.

**Núcleo o Comité de Base:** entidad que representa a las personas que son miembros de PCC y lleva a la práctica la política del partido en el lugar donde actúa.

**Militante:** entidad que representa a las personas vinculadas al PCC.

**Salario:** entidad que representa el dinero que recibe una persona de la empresa o entidad para la que trabaja en concepto de paga, generalmente de manera periódica.

**Escala:** entidad que representa la escala que presenta una persona dentro de la organización.

**Cotización:** entidad que representa el aporte económico individual que realiza cada militante para sufragar los gastos de la organización.

**Cronograma:** entidad que representa el conjunto de funciones y tareas que se llevan a cabo en un tiempo estipulado y bajo unas condiciones que garanticen la optimización del tiempo.

### 2.2 Especificación de Requisitos

En el proceso de desarrollo de software, los requisitos se utilizan como datos de entrada en la etapa de diseño del producto que se desea lograr. Estos establecen con detalle los servicios y restricciones del

## CAPÍTULO 2

sistema, y representan qué debe hacer el sistema, pero no cómo hacerlo, además incorporan las cualidades o propiedades que el mismo debe poseer. (Sommerville, 2015)

### 2.2.1 Requisitos Funcionales

Los requisitos funcionales son declaraciones de los servicios que proveerá el sistema, de manera en que éste reaccionará en situaciones particulares. Estos muestran las funcionalidades que deben satisfacerse para cumplir con las especificaciones de software. (Sommerville, 2015)

*TABLA 3 REQUISITOS FUNCIONALES DEL MÓDULO DE COTIZACIÓN (ELABORACIÓN PROPIA)*

<b>RF_1:</b> Insertar tipos de cotización	<b>RF_15:</b> Eliminar cotización
<b>RF_2:</b> Editar tipos de cotización	<b>RF_16:</b> Listar cotización
<b>RF_3:</b> Eliminar tipos de cotización	<b>RF_17:</b> Insertar cronograma
<b>RF_4:</b> Listar tipos de cotización	<b>RF_18:</b> Editar cronograma
<b>RF_5:</b> Insertar escala PCC	<b>RF_19:</b> Eliminar cronograma
<b>RF_6:</b> Editar escala PCC	<b>RF_20:</b> Listar cronograma
<b>RF_7:</b> Eliminar escala PCC	<b>RF_21:</b> Insertar plan de ingreso
<b>RF_8:</b> Listar escala PCC	<b>RF_22:</b> Editar plan de ingreso
<b>RF_9:</b> Insertar escala UJC	<b>RF_23:</b> Eliminar plan de ingreso
<b>RF_10:</b> Editar escala UJC	<b>RF_24:</b> Listar plan de ingresos
<b>RF_11:</b> Eliminar escala UJC	<b>RF_25:</b> Insertar liquidación de la cotización
<b>RF_12:</b> Listar escala UJC	<b>RF_26:</b> Editar liquidación de la cotización
<b>RF_13:</b> Insertar cotización	<b>RF_27:</b> Eliminar liquidación de la cotización
<b>RF_14:</b> Editar cotización	<b>RF_28:</b> Listar liquidación de la cotización



## CAPÍTULO 2

<b>RF_29:</b> Insertar slip bancario	<b>RF_30:</b> Actualizar slip bancario
<b>RF_31:</b> Eliminar slip bancario	<b>RF_32:</b> Listar slip bancario
<b>RF_33:</b> Visualizar cronograma provincial	
<b>RF_34:</b> Visualizar cronograma nacional	

### 2.2.2 Requisitos No Funcionales

Los requisitos no funcionales son aquellos requisitos que no describen información a guardar, ni funciones a realizar, sino que son propiedades que hacen al producto atractivo, usable, rápido o confiable. Además, se conocen como un conjunto de características de calidad, que es necesario tener en cuenta al diseñar e implementar el software. (Sommerville, 2015)

#### ❖ Usabilidad

**RnF\_1:** El módulo debe ser una aplicación web.

**RnF\_2:** La aplicación debe mostrar una interfaz con elementos identificativos al contexto donde se aplica

#### ❖ Seguridad

**RnF\_3** El módulo debe ser tolerante a fallos, y mostrar solo la información necesaria para orientar al usuario.

#### ❖ Eficiencia:

**RnF\_4:** El módulo debe permitir que 420 usuarios interactúen con él de manera concurrente.

#### ❖ Rendimiento

**RnF\_5** Los tiempos de respuesta del módulo deben ser rápidos, no mayores de 5 segundos para cualquier operación realizada por el usuario.

#### ❖ Implementación

**RnF\_6:** El módulo debe ser implementado utilizando las herramientas, lenguajes y tecnologías utilizadas en el desarrollo del Sistema para el control de la Militancia del PCC (SICOM).

## CAPÍTULO 2

### 2.3 Descripción de la solución propuesta

El módulo web para la gestión y control de la cotización constituye un subsistema del Sistema para el Control de la Militancia del PCC (SICOM). Para acceder al módulo es necesario que el usuario se autentique con rol de estadístico municipal, provincial o nacional.

El estadístico municipal es el encargado de gestionar y administrar toda la información necesaria de los militantes en cada núcleo, mientras que los estadísticos provinciales y nacionales se encargan de visualizar el cronograma realizado para la entrega de la cotización en todos los municipios y provincias.

El subsistema estará compuesto por 3 secciones: Gestionar cotización, Contabilidad y Gestionar escalas.

En la sección **Gestionar Cotización** el Estadístico Municipal accede a las funcionalidades Cotización y Cronograma. La funcionalidad cotización muestra el listado de todos los núcleos que cotizan, compuesto por los militantes que pertenecen a un núcleo determinado, a los cuales se les gestiona la cotización. En caso de que un militante no aporte la cuota a pagar, se le acumula la deuda en pago pendiente, actualizándose por cada mes que deje de pagar, hasta que finalmente realice la liquidación correspondiente a las cuotas atrasadas. La funcionalidad cronograma muestra la lista de cronogramas realizados donde se especifican las fechas a las que cada núcleo le corresponde entregar la cotización.

En la sección **Contabilidad** el Estadístico Municipal accede a las funcionalidades Plan de Ingreso, Modelo de Liquidación y Slip Bancario. La funcionalidad Plan de Ingreso, se encarga de gestionar la cantidad anual que debe ingresar cada núcleo al PCC; mientras que Slip Bancario permite gestionar el importe de la cantidad de dinero que es entregado al banco mensualmente, el mismo identificado mediante un código. La funcionalidad Modelo de Liquidación permite gestionar la cantidad de dinero que liquidó cada núcleo teniendo en cuenta la cantidad de militantes que cotizaron.

En la sección **Gestionar escalas** el Estadístico Municipal accede a las funcionalidades Tipo de Cotización y escalas PCC y UJC. La funcionalidad Tipo de Cotización permite gestionar la información asociada a tipo de cotización; mientras que para realizar el cálculo de la cotización se tiene en cuenta una escala, en dependencia del salario recibido de cada militante.

#### 2.3.1 Definición de los actores del sistema

Los actores son las distintas personas que usan el sistema o producto en el contexto de la función y comportamiento que va a describirse. Estos representan los papeles que desempeñan las personas cuando

## CAPÍTULO 2

opera el sistema. (Pressman, 2010) En el caso específico de la solución que se propone los actores serán estadístico municipal, estadístico provincial y estadístico nacional. A continuación, se muestra la descripción del actor del sistema.

*TABLA 4 ACTORES DEL SISTEMA (ELABORACIÓN PROPIA)*

Actores	Descripción
<b>Estadístico Municipal</b>	Encargado de gestionar y administrar toda la información necesaria de los militantes en cada núcleo.

Actores	Descripción
<b>Estadístico Provincial</b>	Encargado de visualizar el cronograma realizado para el cobro de la cotización en todos los municipios.

Actores	Descripción
<b>Estadístico Nacional</b>	Encargado de visualizar el cronograma realizado para el cobro de la cotización en todas las provincias.

# CAPÍTULO 2

## 2.3.2 Diagrama de Caso de Uso del Sistema

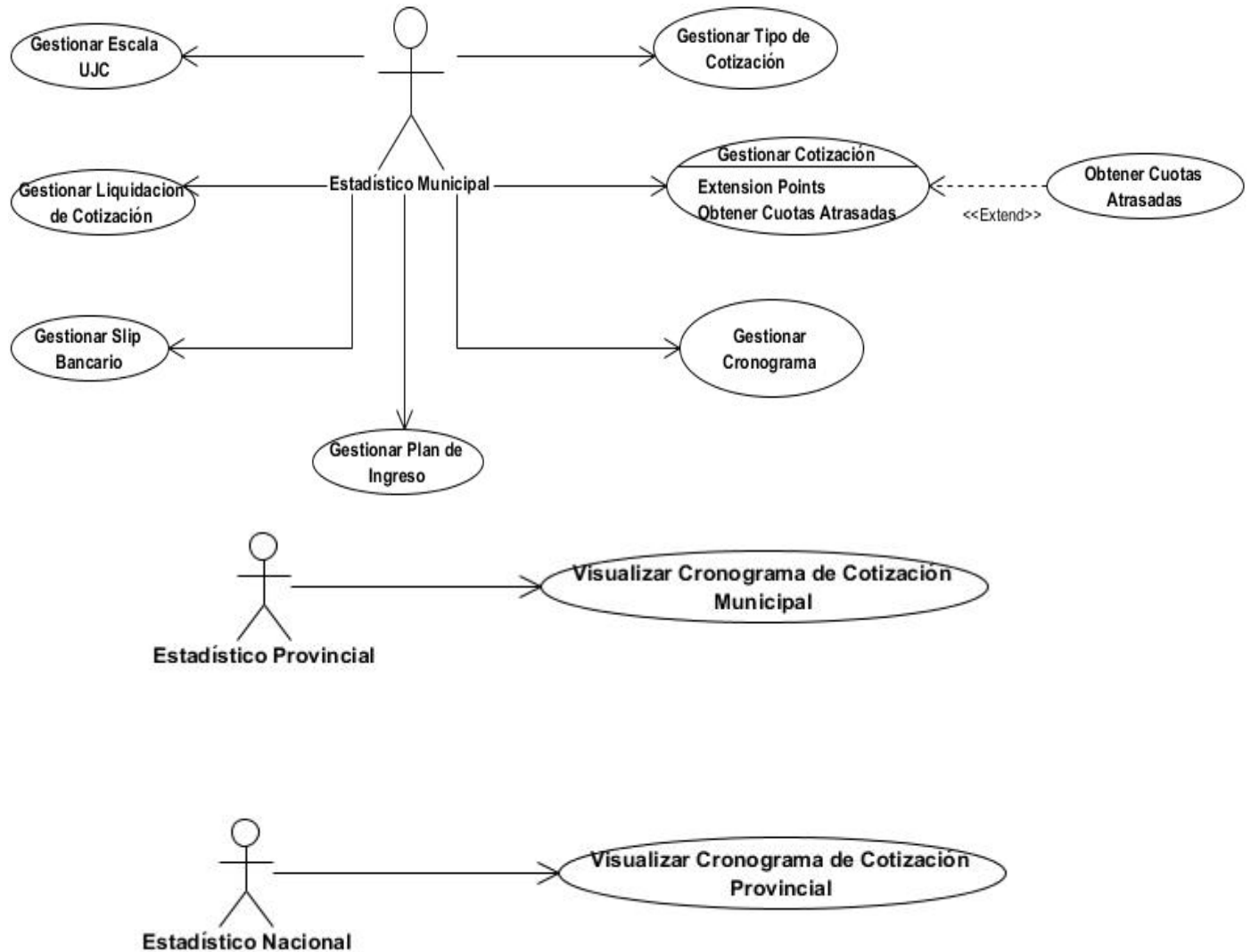


ILUSTRACIÓN 2 DIAGRAMA DE CASO DE USO DEL SISTEMA

## 2.3.3 Descripción de los Casos de Uso del Sistema

La descripción de los casos de uso (CUS) del sistema recoge el objetivo del CUS así como un resumen del proceso que alberga el mismo. Las siguientes tablas hacen una descripción textual de los casos de uso Gestionar Tipo de Cotización.

## CAPÍTULO 2

TABLA 5 DESCRIPCIÓN CU GESTIONAR TIPO DE COTIZACIÓN (ELABORACIÓN PROPIA)

<b>Nombre del Caso de Uso</b>	Gestionar Tipo de Cotización
<b>Objetivos</b>	Administrar el Tipo de Cotización.
<b>Actores</b>	Estadístico Municipal
<b>Resumen</b>	<p>El Caso de Uso inicia cuando el Estadístico Municipal va a realizar alguna de las siguientes operaciones:</p> <ul style="list-style-type: none"><li>• <b>Insertar tipos de cotización:</b> Cuando se va a insertar un tipo de cotización existente, el Estadístico Municipal introduce los datos, así finaliza el Caso de Uso.</li><li>• <b>Editar tipos de cotización:</b> Cuando el Estadístico Municipal necesita modificar los datos, modifica los datos del tipo de cotización y actualiza la Base de Datos, así finaliza el Caso de Uso.</li><li>• <b>Eliminar tipo de cotización:</b> Cuando un Estadístico Municipal necesita eliminar un tipo de cotización, debe tener en cuenta que si hay militantes con ese tipo de cotización asignado no podrá eliminarse, así finaliza el Caso de Uso.</li><li>• <b>Listar tipos de cotización:</b> Cuando un Estadístico Municipal necesita listar tipos de cotización, debe listar las causas de no ingreso, y finaliza el Caso de Uso.</li></ul>
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítico
<b>Referencias</b>	RF1, RF2, RF3, RF4
<b>Precondiciones</b>	El Estadístico Municipal debe tener permisos de administrador.

## CAPÍTULO 2

<b>Postcondiciones</b>	<p>En dependencia de la acción que el Estadístico Municipal desee realizar:</p> <ul style="list-style-type: none"> <li>• Se Inserta un Tipo de Cotización.</li> <li>• Se edita Tipo de Cotización.</li> <li>• Se elimina Tipo de Cotización.</li> <li>• Se listan los Tipos de Cotización.</li> </ul>
------------------------	---

### Curso Normal de los Eventos

<b>Acciones del Actor</b>	<b>Propuesta del Sistema</b>
<p>1-El Estadístico Municipal selecciona cual acción desea realizar:</p> <ol style="list-style-type: none"> <li>1. Insertar Tipo de Cotización.</li> <li>2. Modificar Tipo de Cotización.</li> <li>3. Eliminar Tipo de Cotización.</li> <li>4. Listar Tipo de Cotización.</li> </ol>	<p>2- En dependencia de la acción seleccionada por el Estadístico Municipal el Sistema muestra la interfaz correspondiente:</p> <p>Primeramente, el sistema muestra la opción Gestionar Tipo de Cotización, donde se despliega una tabla que muestra las opciones que puede elegir el Estadístico Municipal. Estas son:</p> <ol style="list-style-type: none"> <li>1. Insertar un nuevo tipo de cotización: Ir a la sección <b>“insertar tipo de cotización”</b>.</li> <li>2. Modificar datos de tipo de cotización: Ir a la sección <b>“editar tipo de cotización”</b>.</li> <li>3. Eliminar un Tipo de Cotización: Ir a la sección <b>“eliminar tipo de cotización”</b>.</li> </ol> <p>Listar Tipos de Cotización: Ir a la sección <b>“listar tipo de cotización”</b>.</p>

### Sección “Insertar Tipo Cotización”: Flujo Normal de Eventos

<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1- El Estadístico Municipal selecciona la opción Insertar.	2- El sistema muestra la interfaz para insertar un tipo de cotización.
3- El Estadístico Municipal inserta los datos de un tipo de cotización: nombre	4- El sistema verifica que no exista ningún campo vacío.

## CAPÍTULO 2

e indica al sistema almacenar el tipo de cotización.	
	5- El sistema valida los datos insertados
	6- El sistema registra el nuevo tipo de cotización, así finaliza el Caso de Uso.
<b>Sección “Insertar Tipo Cotización”: Paso 4. Flujo Alternativo de Eventos: Existe campos vacíos</b>	
	4.1- El sistema muestra un mensaje: “Existen campos vacíos”.
<b>Sección “Insertar Tipo Cotización”: Paso 5. Flujo Alternativo de Eventos: Valida datos insertados</b>	
	5.1- El sistema muestra un mensaje: “Los datos insertados son incorrectos”.
<b>Sección “Editar Tipo de Cotización”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del sistema</b>
1- El Estadístico Municipal selecciona la opción Editar.	2- El sistema muestra el listado de tipos de cotización que se desea editar
3- El estadístico Municipal selecciona el tipo de cotización que desea editar	4- El Estadístico Municipal edita los datos en el tipo de cotización que desea actualizar e indica guardar los cambios realizados.
	5- El sistema valida completitud de los datos.
	6- El sistema actualiza el tipo de cotización.
<b>Sección “Editar tipo de cotización”: Paso 3. Flujo Alternativo de Eventos: Cancelar Modificación</b>	
4.1- Si el Estadístico Municipal no indica guardar los cambios, cancela la operación y finaliza así el caso de uso.	
<b>Sección “Editar tipo de cotización”: Paso 4. Flujo Alternativo de Eventos: Existen Campos Obligatorios vacíos</b>	

## CAPÍTULO 2

	4.1- El sistema muestra un mensaje “Existen campos vacíos” y va a la acción 3 de esta sección.
<b>Sección “Eliminar tipo de cotización”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del sistema</b>
1- El Estadístico Municipal selecciona el tipo de cotización que desee eliminar.	2- El sistema muestra la lista de tipos de cotización que se desea eliminar
<b>Sección “Listar tipo de cotización”: Flujo Normal de Eventos</b>	
<b>Acciones del autor</b>	<b>Respuesta del sistema</b>
1- El Estadístico Municipal selecciona la opción listar tipo de cotización.	2- El sistema muestra la acción “ <b>listar tipo de cotización</b> ”.
	3- El sistema muestra la lista de tipos de cotización existente en la Base de Datos.

### 2.4 Arquitectura del Software

La Arquitectura de Software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones. (Rivera, 2010)

Para la solución propuesta se decidió usar el *framework* de desarrollo Django, lo que implica una serie de decisiones de diseño. Django sigue una arquitectura Modelo-Vista-Controlador solo que hace una adaptación de esta a Modelo-Vista-Plantilla (a partir de ahora MTV por sus siglas en inglés, *Model Template View*). Por tanto, el sistema propuesto hereda una arquitectura MTV.



## CAPÍTULO 2

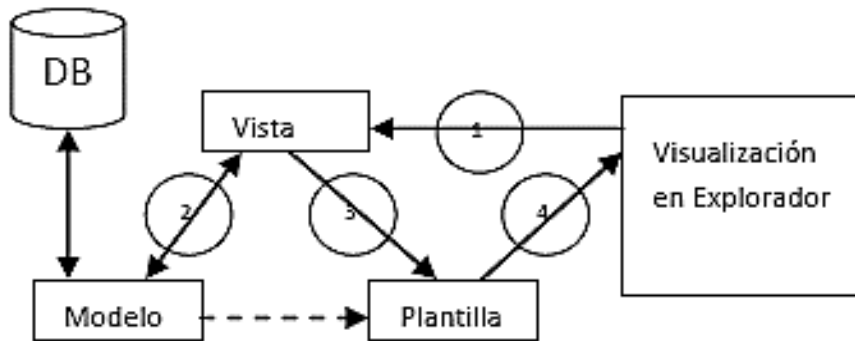


ILUSTRACIÓN 3 FUNCIONAMIENTO DE MTV DJANGO (RIVERA, 2010)

1. El Navegador manda una solicitud
2. La vista interactúa con el modelo para obtener datos
3. La vista llama a la plantilla
4. La plantilla renderiza la respuesta a la solicitud del navegador

**Modelo:** Contiene toda la información sobre los datos. Cada una de las entidades de la base de datos se encuentra en el modelo en forma de clases de Python, y sus atributos se almacenan en variables con ciertos parámetros. También estos archivos poseen métodos, lo que permite indicar y controlar el comportamiento de los datos.

**Vista:** Es la capa de la lógica de negocios, contiene la lógica que accede al modelo y la delega a la plantilla apropiada. Esta capa sirve de “puente” entre el modelo y la plantilla, se presenta en forma de funciones de Python y su función principal es determinar qué datos serán visualizados en las plantillas.

**Plantilla:** Define la interfaz de las páginas web, o sea, cómo se van a mostrar los datos al usuario.

## CAPÍTULO 2

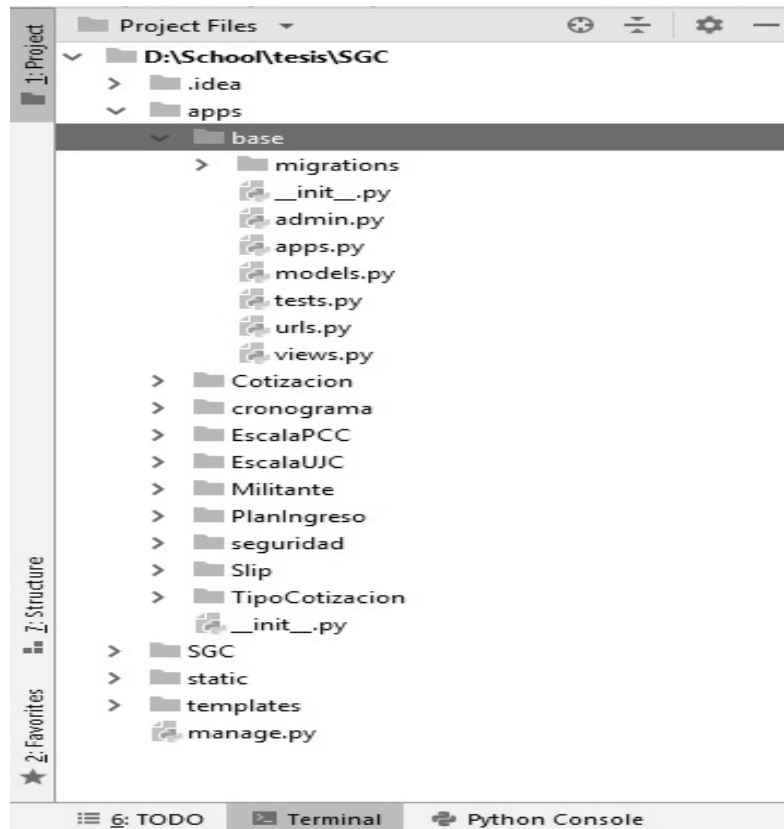


ILUSTRACIÓN 4 ESTRUCTURA DE DIRECTORIOS DEL MÓDULO DE COTIZACIÓN (ELABORACIÓN PROPIA)

# CAPÍTULO 2

## 2.4.1 Diagrama de Clase del Diseño

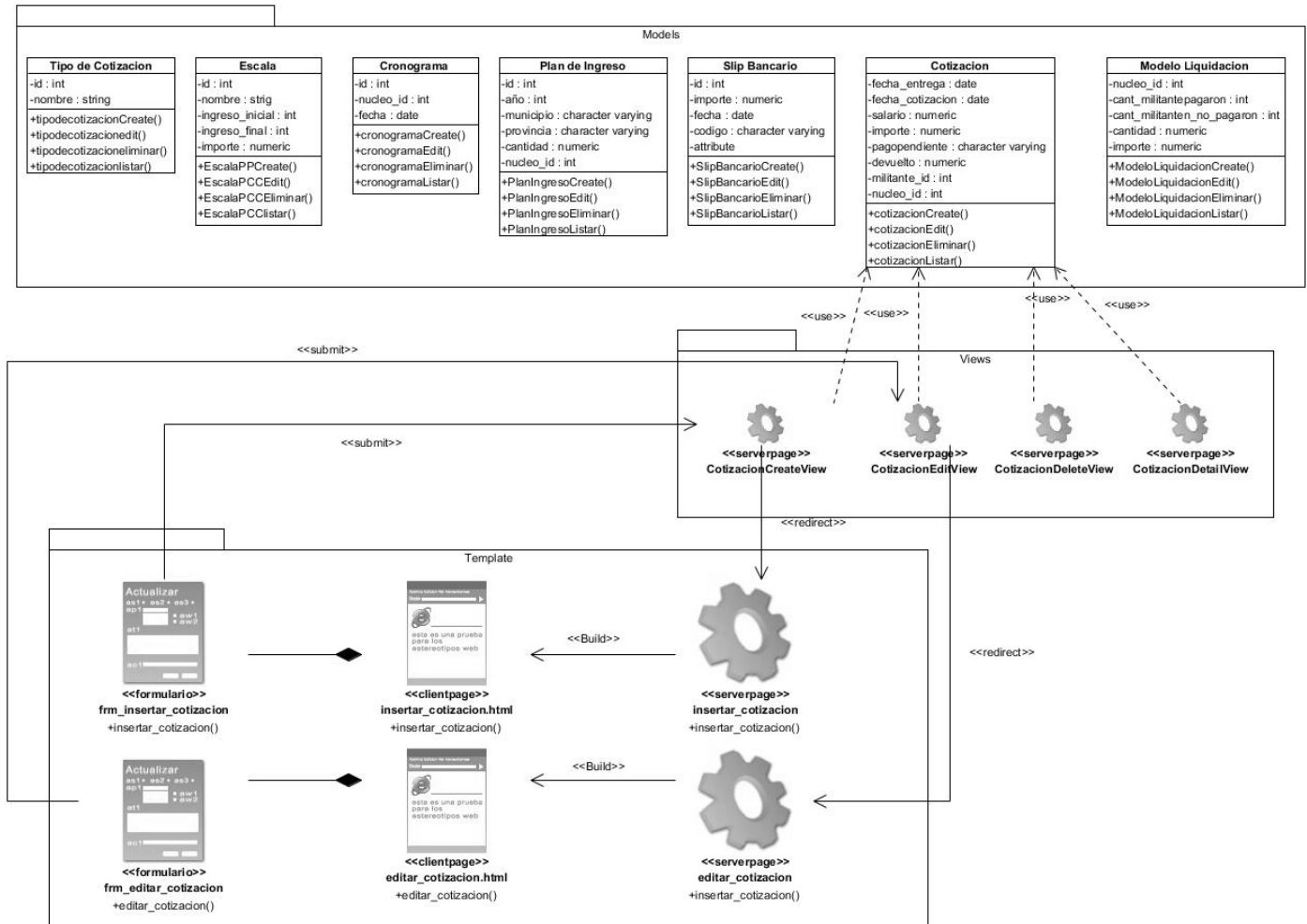


ILUSTRACIÓN 5 DIAGRAMA DE CLASE DEL DISEÑO (ELABORACIÓN PROPIA)

## 2.4.2 Patrones del Diseño

Los patrones de diseño son descripciones de clases y objetos relacionados que están particularizados para resolver un problema de diseño general en un determinado contexto. Un patrón de diseño nomina, abstrae e identifica los aspectos clave de una estructura de diseño común, lo que los hace útiles para crear un diseño orientado a objetos reutilizable. El patrón de diseño identifica las clases e instancias participantes, sus roles y colaboraciones, y la distribución de responsabilidades. (Escalante, 2015)

## CAPÍTULO 2

### Patrones Gof

Los Patrones Gof (por sus siglas en inglés, *Gang of Four*) describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos y agrupan los patrones de acuerdo a su propósito: creación, estructura y comportamiento.

(Castelao, 2008)

A continuación, se describe el patrón Gof utilizado en la solución propuesta:

**Decorador:** Patrón estructural que extiende la funcionalidad de un objeto dinámicamente de manera tal que es transparente a sus clientes, utiliza una instancia de una subclase de la clase original que delega las operaciones al objeto original (Gamma et al., 1994). Este patrón se evidencia en la clase `login_require` al utilizar el decorador de Django `@login_require` para garantizar que el usuario este autenticado y tenga permiso para acceder a esa funcionalidad.



ILUSTRACIÓN 6 EJEMPLO PATRÓN DECORADOR

```
@login_required(login_url='login')
@permission_required(perm='PlanIngreso.add_planingreso', reverse_lazy('error_403'))
def dispatch(self, request, *args, **kwargs):
    super(PlanIngresoCreate, self).dispatch(*args, **kwargs)
```

ILUSTRACIÓN 7 EJEMPLO PATRÓN DECORADOR (FUNCIONALIDAD PLAN DE INGRESO)

## CAPÍTULO 2

### Patrones GRASP

Los patrones GRASP son parejas de problema solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. Describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. (Castelao, 2008)

Los patrones GRASP utilizados en la solución propuesta fueron:

**Bajo acoplamiento:** Plantea que se debe poder reutilizar las funcionalidades de las distintas clases, con un nivel de dependencia mínima. 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. Por ejemplo, se puede cambiar la URL de cierta parte de la aplicación sin tener que afectar la implementación subyacente o se puede modificar el HTML de una página sin tener que tocar el código Python que la renderiza.

**Alta cohesión:** Asigna responsabilidades de manera tal que la cohesión siga siendo alta, o sea que las funcionalidades de las clases estén altamente relacionadas de forma tal que exista una colaboración entre ellas para compartir el esfuerzo y no caiga todo el peso sobre una única clase. Usar este patrón simplifica el mantenimiento y favorece el bajo acoplamiento. Este patrón se utiliza en todas las clases. Un ejemplo se evidencia en las funcionalidades correspondientes a **Gestionar Cotización**, manifiesto en el diagrama de Clase del Diseño expuesto anteriormente, mostrando la relación que existe entre las funcionalidades de las clases.

**Controlador:** Permite manejar todos los eventos del sistema, al servir de intermediario entre las interfaces y el algoritmo que las implementa. Este patrón se evidencia en la clase *CotizacionCreate*. Dicha función recibe la solicitud enviada por el Estadístico Municipal, crea una cotización en la base de datos y renderiza la respuesta en la plantilla correspondiente.

# CAPÍTULO 2

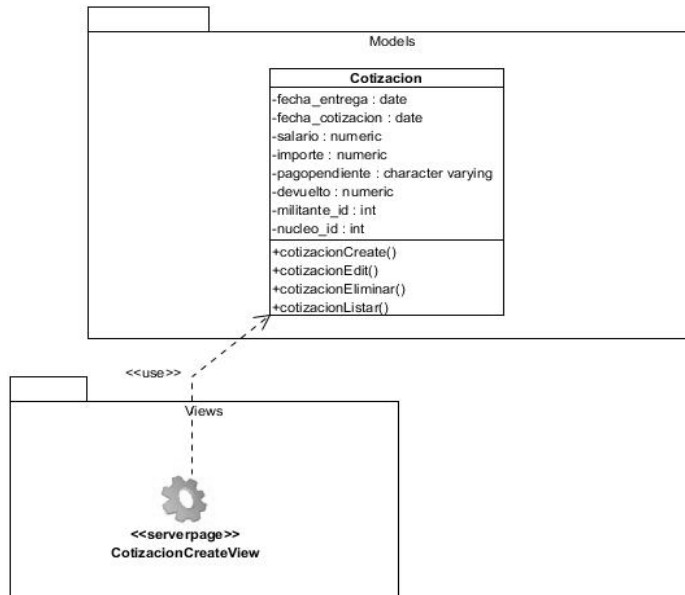


ILUSTRACIÓN 8 EJEMPLO PATRÓN CONTROLADOR (FUNCIONALIDAD COTIZACIÓN)

**Experto en Información:** las responsabilidades deben ser asignadas a las clases que poseen la información para realizar dicha responsabilidad. Un ejemplo se evidencia en la funcionalidad *CotizacionListar()*.

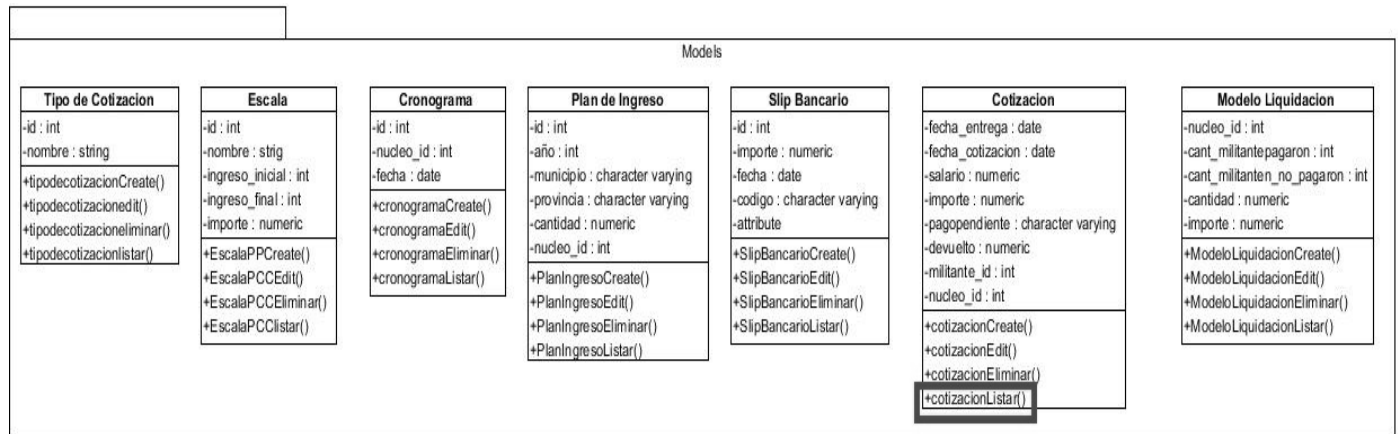


ILUSTRACIÓN 9 EJEMPLO PATRÓN EXPERTO

## CAPÍTULO 2

### Conclusiones del capítulo

- El desarrollo del presente capítulo permitió sentar las bases para la construcción del sistema propuesto a partir del modelado de la propuesta de solución, del análisis y descripción de los procesos de negocio que propició la obtención de las principales funcionalidades que debe cumplir la aplicación y quedaron manifestadas en la descripción de requisitos de software.
- La utilización de los patrones de diseño permitió identificar aspectos importantes de la estructura del diseño del módulo web propuesto, garantizando una mayor organización e hizo el código más eficiente.
- Adoptar la arquitectura de software Modelo-Vista-Plantilla propuesta por el framework de desarrollo utilizado, abonó una propicia organización del módulo a implementar.

# CAPÍTULO 3

## CAPÍTULO 3. Implementación y prueba del Módulo de Cotización para el Sistema de Control de la Militancia del Partido Comunista de Cuba

En el presente capítulo se muestran los artefactos correspondientes a las etapas de implementación y pruebas del sistema. 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. Además, se brindan los aspectos referentes a la implementación y se recogen los resultados obtenidos luego de realizar las pruebas necesarias al Módulo de Cotización para el Sistema de Control de la Militancia del PCC.

### 3.1 Diagrama de Componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, bibliotecas cargadas dinámicamente, etc. (Torres, 2002)

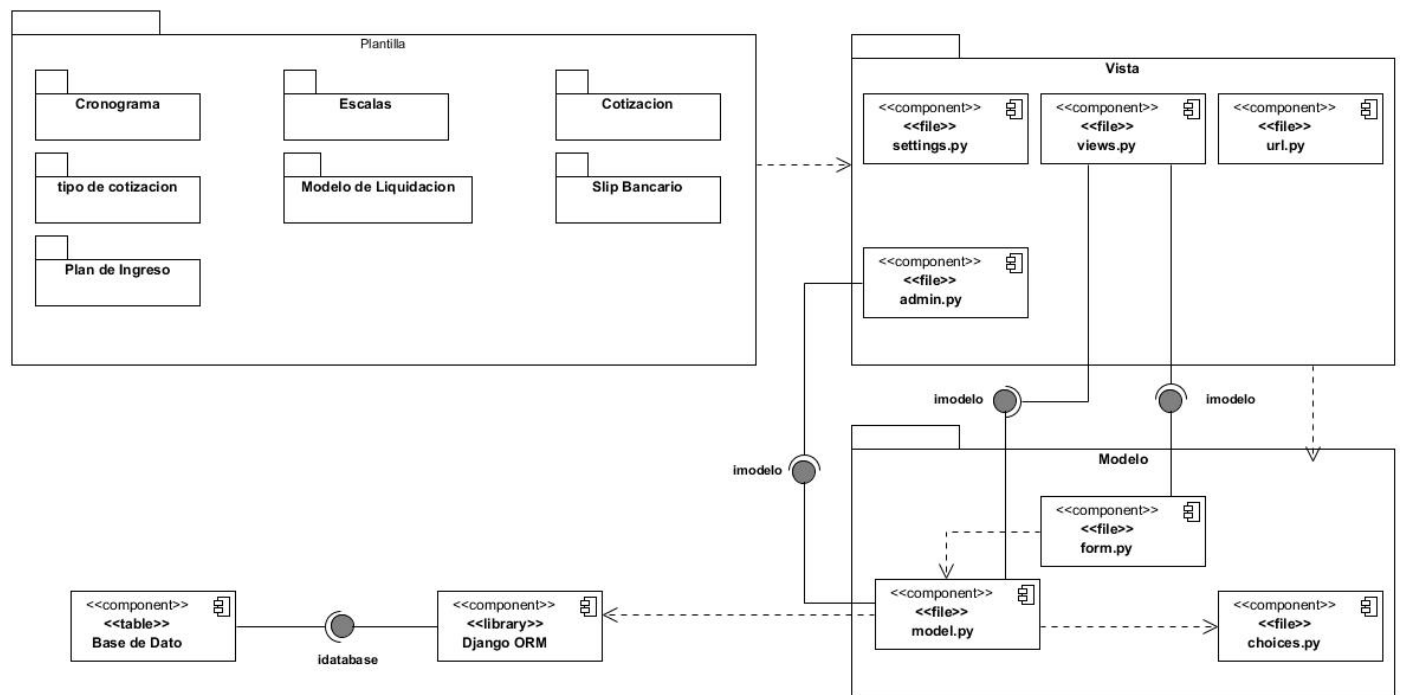


ILUSTRACIÓN 10 DIAGRAMA DE COMPONENTE (ELABORACIÓN PROPIA)



## CAPÍTULO 3

### 3.2 Diagrama de Despliegue

Los diagramas de despliegue muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Modelan la distribución en tiempo de ejecución de los elementos de procesamiento y componentes de software, junto a los procesos y objetos asociados. (Torres, 2002)

A continuación, se muestra el diagrama de despliegue definido para la propuesta de solución, que cuenta con 3 nodos principales. El nodo PC Cliente que requiere de un navegador web, el nodo Servidor Web donde debe estar instalado el módulo web para la gestión y control de la cotización del PCC y el nodo SGBD en el cual debe estar instalado el Sistema Gestor de Bases de Datos PostgreSQL con la base de datos de SICOM. El Nodo PC Cliente representa las estaciones de trabajo de los usuarios que se conectan al Módulo, como: Estadístico Municipal, Provincial y Nacional, las mismas realizan peticiones al Servidor Web mediante protocolo HTTP (*Hypertext Transfer Protocol*) a través del puerto 443. El nodo Servidor Web, a su vez, estará conectado al nodo SGBD mediante el protocolo TCP/IP a través del puerto 5432.



ILUSTRACIÓN 11 DIAGRAMA DE DESPLIEGUE (ELABORACIÓN PROPIA)

### 3.3 Estándares de Codificación

Un estándar de codificación comprende todos los aspectos de la generación del código fuente de un software. Es preciso definir una serie de pautas que lleven por objetivo uniformar la estructura del código de manera tal que este sea lo más legible posible. Cuando el proyecto de software incorpore código fuente previo, o bien cuando realice el mantenimiento de un sistema de software creado anteriormente, el estándar de codificación debería establecer cómo operar con la base de código existente. (Calleja, 2015)

Para la definición de los estándares de codificación que se utilizaron en el presente trabajo de diploma se tuvieron en cuenta las características que rigen a Python. A continuación, se listan los estándares por los que se rige la autora:

## CAPÍTULO 3

1. Los nombres de los atributos, variables y parámetros tendrán todas las letras en minúsculas y usarán el guion bajo como delimitador entre palabras.
2. Declarar las variables en líneas separadas.
3. Los métodos deben comenzar con letra mayúsculas.
4. La mayoría de los elementos se deben nombrar usando sustantivos.

### 3.4 Pruebas del Software

Las pruebas del software es la verificación dinámica del comportamiento de un programa contra el comportamiento esperado, usando un conjunto finito de casos de prueba, seleccionados de manera adecuada. (Paz, 2015)

El objetivo de las pruebas del software es descubrir errores. Este objetivo se logra mediante una serie de pasos. Las de unidad e integración se concentran en la verificación funcional de un componente y en la incorporación de componentes en una arquitectura de software. Las pruebas de validación demuestran la conformidad con los requerimientos del software y las del sistema validan el software una vez que se incorporó en un sistema más grande. Cada paso de la prueba se logra a través de una serie de técnicas de prueba sistemáticas que auxilian en el diseño de casos de prueba. Con cada paso de prueba, se amplía el nivel de abstracción con la que se considera el software. (Pressman, 2010)

Para garantizar la calidad requerida de la solución propuesta, así como el cumplimiento de los requisitos especificados, se desarrollaron pruebas que garantizan el cumplimiento de dichas acciones.

#### 3.4.1 Estrategias de Pruebas

Una estrategia de prueba de software proporciona una guía que describe los pasos que deben realizarse como parte de la prueba, cuándo se planean y se llevan a cabo dichos pasos, y cuánto esfuerzo, tiempo y recursos se requerirán. Por tanto, cualquier estrategia de prueba debe incorporar la planificación, el diseño de casos de prueba, la ejecución y la recolección y evaluación de los resultados. Una estrategia de prueba de software debe ser suficientemente flexible para promover un uso personalizado de la prueba. Al mismo tiempo, debe ser suficientemente rígida para alentar la planificación razonable y el seguimiento de la gestión conforme avanza el proyecto. (Pressman, 2010)

## CAPÍTULO 3

### 3.4.2 Pruebas de Aceptación

Cuando se construye software para un cliente, se llevan a cabo las pruebas de aceptación para permitir que el cliente valide todos los requisitos. Son ejecutadas antes de que la aplicación sea instalada dentro de un ambiente de producción. La prueba de aceptación es conducida a determinar como el sistema satisface sus criterios de aceptación validando los requisitos que han sido levantados para el desarrollo, incluyendo la documentación y procesos de negocio. (Pressman, 2010)

#### Prueba alfa

La prueba alfa se lleva a cabo, por un cliente, en el lugar de desarrollo. Se usa el software de forma natural con el desarrollador como observador del usuario y registrando los errores y los problemas de uso. Las pruebas alfa se llevan a cabo en un entorno controlado. (Pressman, 2010)

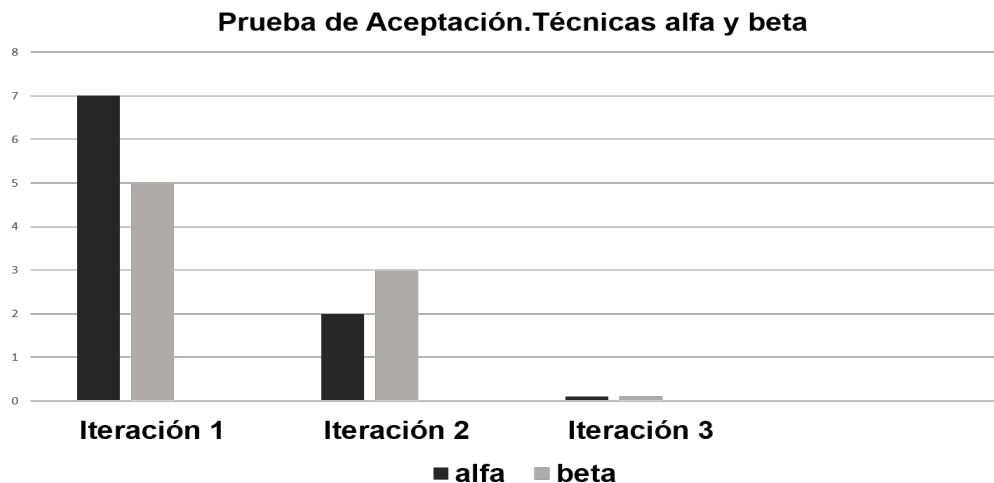
#### Prueba beta

La prueba beta se lleva a cabo por los usuarios finales del software en los lugares de trabajo de los clientes. El cliente registra todos los problemas que encuentra durante la prueba beta e informa a intervalos regulares al desarrollador. Como resultado de los problemas informados durante la prueba beta, el desarrollador del software lleva a cabo modificaciones y así prepara una versión del producto de software para toda la clase de clientes (Pressman, 2010)

#### Resultados de las pruebas de aceptación

Para la ejecución de las pruebas de aceptación se realizaron varias reuniones con el cliente, en este caso, el director del centro CISED (Centro de Identificación y Seguridad Digital) de la UCI, Ing. Ciencias Informáticas Osay Gonzales Fuentes donde se le explicó el funcionamiento del módulo y se observó su interacción con los componentes del mismo identificando un conjunto de no conformidades agrupadas en errores de validación e interfaz. Luego se realizaron pruebas por el cliente en su entorno de trabajo y sin observadores, permitiéndole iniciar el trabajo con el módulo en una situación real donde identificó un conjunto de no conformidades relacionadas con funciones incorrectas, las mismas fueron erradicándose a medida que eran informadas. La realización de las pruebas de aceptación a la solución utilizando las técnicas alfa y beta, permitió demostrar que el sistema cumple satisfactoriamente los requisitos del cliente, emitiéndose el acta de aceptación, garantizando evaluar el grado de calidad del módulo (**Ver acta de aceptación Anexo 4**).

## CAPÍTULO 3



*ILUSTRACIÓN 12 RESULTADOS PRUEBA DE ACEPTACIÓN*

### 3.4.3 Pruebas de Unidad

La prueba de unidad centra el proceso de verificación en la menor unidad del diseño: el módulo. Usando la descripción del diseño detallado como guía, se prueban los caminos de control importantes, con el fin de descubrir errores dentro del módulo. Se prueba la interfaz para asegurar que la información fluye de forma adecuada hacia y desde la unidad del programa que está siendo probada. Se examinan las estructuras de datos locales para asegurar que los datos que se mantienen temporalmente conservan su integridad durante la ejecución del algoritmo. El objetivo de estas pruebas es ejecutar un código fuente al llamar directamente a los métodos de una clase pasándole a estos los parámetros adecuados. (Paz, 2015)

Para la realización de las pruebas unitarias se ejecutaron dos técnicas, primeramente, la técnica de caja blanca utilizando el método de camino básico o ruta básica y como segunda técnica haciendo uso del *framework* de prueba *PyUnit*, donde los resultados obtenidos se pueden observar en el **Anexo 3**.

La técnica de **caja blanca**, a veces definida como prueba de “caja de cristal” o “caja transparente”, es una técnica de diseño de casos de prueba que usa la estructura de control para obtener los casos de prueba. Dentro de esta estructura de control se pueden encontrar la estructura de un componente de software como puede ser sentencias de decisiones, caminos distintos del código, la estructura de una página web, etc. (Peño, 2015). Los métodos de prueba de caja blanca aportan los siguientes puntos:

- Garantizan que todas las rutas del código se revisan al menos una vez.
- Revisan las condiciones lógicas.

## CAPÍTULO 3

- Revisan estructuras de datos.

La prueba de ruta básica es un método de prueba de caja blanca, que se basa en diseñar un caso de prueba por cada camino independiente del programa. Con esta técnica se intenta garantizar que se prueben todos los caminos de ejecución del programa, al menos una vez. (Peño, 2015)

Para aplicar las pruebas de ruta básica hay que conocer la representación que se conoce como grafo de flujo de datos y complejidad ciclomática.

1. Generar grafo de flujos de datos.
2. Cálculo de complejidad ciclomática,  $V(G)$ .

$V(G) = NA$  (Número de Aristas) -  $NN$  (Número de Nodos) + 2.

$V(G) = P$  (Nodos predicados) + 1.

$V(G) = R$  (Número de regiones).

3. Determinar los caminos independientes o básicos.
4. Generar un caso de prueba para cada camino de ejecución.

Con la utilización de este método de camino básico es posible la detección de errores no detectados anteriormente en el código. De igual manera el mantenimiento del sistema sería de forma más efectiva y certera. Este tipo de prueba se le aplico a todos los métodos del sistema desarrollado, a continuación, se representa los resultados del método *CotizacionCreate* (**Ver Anexo 2**).

Paso1: Generar el grafo.

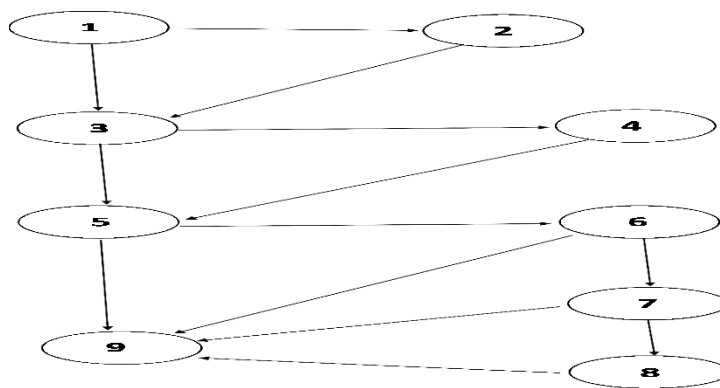


ILUSTRACIÓN 13 GRAFO DE FLUJO DE DATOS DEL MÉTODO COTIZACIONCREATE

## CAPÍTULO 3

Paso2: Calcular la complejidad.

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

$$V(G) = 13 - 9 + 2$$

$$V(G) = 6$$

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

$$V(G) = 5 + 1$$

$$V(G) = 6$$

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

$$V(G) = 6$$

Paso3: Determinar los caminos básicos.

CB1: 1-2-3-4-5-6-7-8-9

CB2: 1-2-3-5-6-7-8-9

CB3: 1-3-4-5-6-7-8-9

CB4: 1-2-3-5-6-9

CB5: 1-3-5-6-7-9

CB6: 1-3-5-9

Paso4: Caso de prueba para el camino básico.

Caso de prueba para camino básico 1.

Entrada: \$formulario->isValid() == true

Resultado esperado: Crear Cotización

Resultado de la prueba: Satisfactorio.

Caso de prueba para camino básico 2.

Entrada: \$formulario->isValid() == false

Resultado esperado: Mostrar página Crear.

Resultado de la prueba: Satisfactorio.

## CAPÍTULO 3

### Caso de prueba para camino básico 3.

Entrada: \$element->type == "application/rss+xml"

Resultado esperado: Se obtiene el href.

Resultado de la prueba: Satisfactorio.

### Caso de prueba para camino básico 4.

Entrada: \$data

Resultado esperado: No encuentra la página.

Resultado de la prueba: Satisfactorio.

### Caso de prueba para camino básico 5.

Entrada: \$formulario->isValid() == true

Resultado esperado: lista debe

Resultado de la prueba: Satisfactorio.

### Caso de prueba para camino básico 6.

Entrada: \$data

Resultado esperado: No encuentra la página.

Resultado de la prueba: Satisfactorio.

#### **3.4.4 Prueba de Integración**

Estas pruebas se ocupan de probar las interfaces entre los componentes, las interacciones con distintas partes de un mismo sistema, como el sistema operativo, el sistema de archivos, el hardware y las interfaces entre varios sistemas (Peño, 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

## CAPÍTULO 3

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.

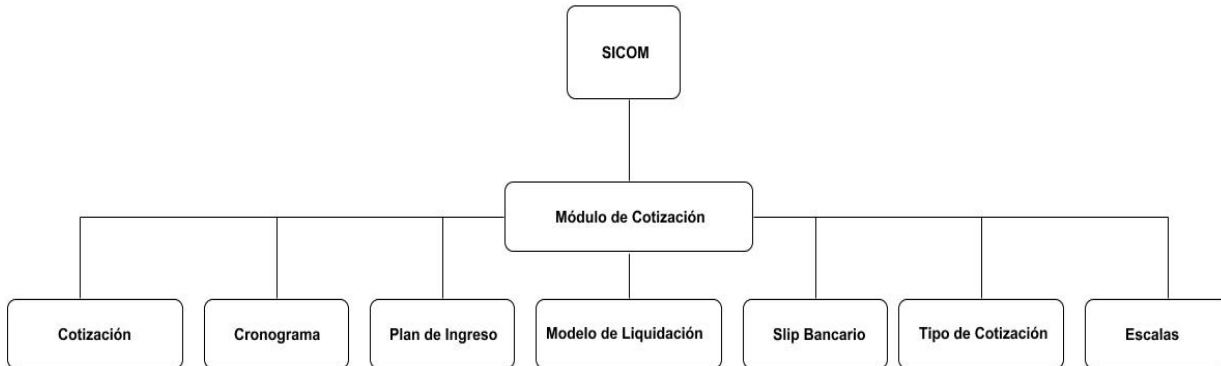


ILUSTRACIÓN 14 DIAGRAMA PRUEBA DE INTEGRACIÓN

Los componentes Cotización, Cronograma, Plan de Ingreso, Modelo de Liquidación, Slip Bancario, Tipo de Cotización y Escalas se integrarían primero al módulo de cotización este a su vez al siguiente nivel de control, Sistema para el Control de la Militancia del PCC (SICOM).

### 3.4.5 Prueba de Validación

Las pruebas de validación comienzan en la culminación de las pruebas de integración, cuando se ejercitaron componentes individuales, el software está completamente ensamblado como un paquete y los errores de interfaz se descubrieron y corrigieron. En el nivel de validación o de sistema, desaparece la distinción entre software convencional, software orientado a objetos y *webapps*. Las pruebas se enfocan en las acciones visibles para el usuario y las salidas del sistema reconocibles por el usuario. (Pressman, 2010)

La validación del software se logra a través de una serie de pruebas que demuestran conformidad con los requerimientos. Un plan de prueba subraya las clases de pruebas que se van a realizar y un procedimiento de prueba. Este procedimiento define casos de prueba específicos que se diseñan para garantizar que: se satisfacen todos los requerimientos de funcionamiento, se logran todas las características de comportamiento, todo el contenido es preciso y se presenta de manera adecuada. Además, se logran todos los requerimientos de rendimiento, documentación correcta y se satisfacen la facilidad de uso. (Pressman, 2010)



## CAPÍTULO 3

La validación del software se consigue mediante una serie de pruebas de **caja negra** que se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra intentan encontrar errores en las categorías siguientes (Pressman, 2010):

1. Funciones incorrectas o faltantes.
2. Errores de interfaz.
3. Errores en las estructuras de datos o en el acceso a bases de datos externas.
4. Errores de comportamiento o rendimiento.
5. Errores de inicialización y terminación

La prueba de caja negra aplicada al módulo se denomina Partición de Equivalencia que consiste en los valores de entrada del programa o del sistema, se dividen en grupos que vayan a tener comportamiento similar, de manera que puedan ser procesados de la misma forma. Las particiones de equivalencia o clases son aplicables a datos válidos y datos no válidos. También, pueden aplicarse a los valores de salida, valores internos, valores relativos al tiempo o a los parámetros de interfaz. (Pressman, 2010)

A continuación, se presentan los casos de prueba del CU Gestionar Tipo de cotización.

*TABLA 6 DISEÑO DE CASOS DE PRUEBA DEL CASO DE USO GESTIONAR TIPO DE COTIZACIÓN (ELABORACIÓN PROPIA)*

<b>Nombre de la sección</b>	<b>Escenario</b>	<b>Acción realizada</b>	<b>Respuesta del sistema</b>	<b>Resultado de la prueba</b>
SC 1: Insertar Tipo de cotización	EC 1.1: Introducir los datos tipo de cotización.	El estadístico municipal introduce los datos del tipo de cotización tales como: nombre.	El sistema inserta el nuevo tipo de cotización.	Resultado Satisfactorio.
	EC 1.2: El tipo de cotización insertado está incompleto.	El estadístico municipal selecciona el botón aceptar	El sistema muestra un mensaje de alerta que existen campos vacíos.	Resultado Satisfactorio.

## CAPÍTULO 3

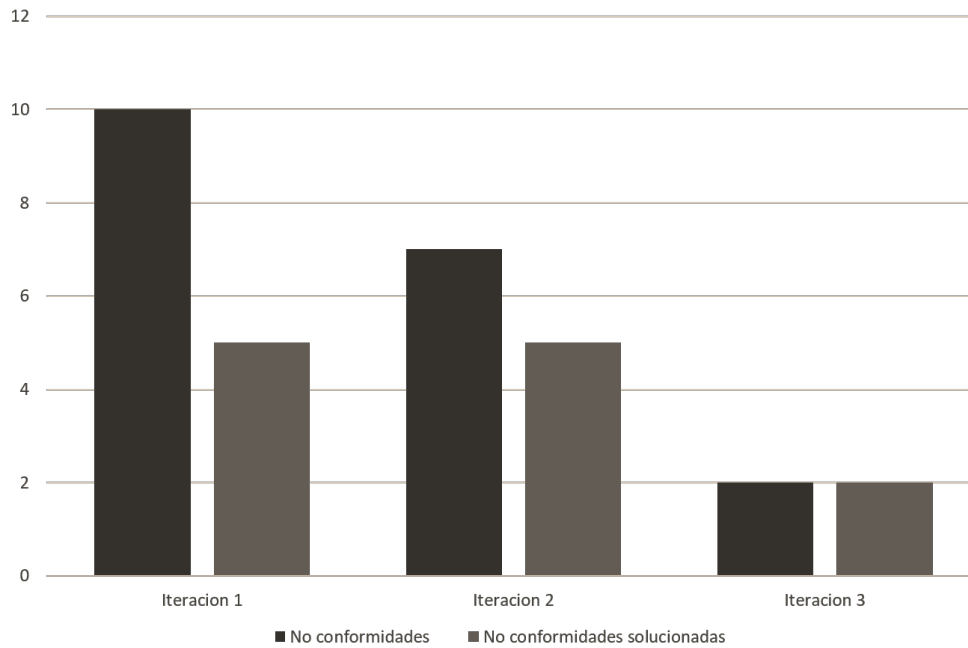
		para insertar tipo de cotización.		
SC 2: Editar tipo de cotización	EC 2.1: Editar tipo de cotización	El estadístico municipal introduce los datos que desea editar.	El sistema edita tipo de cotización.	Resultado Satisfactorio.
	EC 2.1: el tipo de cotización a editar está incompleto.	El estadístico municipal selecciona el botón aceptar para editar el tipo de cotización.	El sistema muestra un mensaje de alerta que existen campos vacíos.	Resultado Satisfactorio.
SC 3: Eliminar Tipo de cotización.	EC 2.1: Eliminar tipo de cotización.	El estadístico municipal selecciona el tipo de cotización que desea eliminar	El sistema elimina tipo de cotización.	Resultado Satisfactorio.
SC 4: Listar tipo de cotización.	EC 3.1: Listar tipos de cotización.	El estadístico municipal selecciona la opción listar tipo de cotización.	El sistema lista los tipos de cotizaciones.	Resultado Satisfactorio.

### Resultados de las pruebas de validación

Las pruebas de validación se ejecutaron tras concluir cada una de las tres iteraciones de implementación, de manera general en la primera iteración se detectaron 10 no conformidades, agrupadas en no conformidades de código y ortografía, de ellas fueron solucionadas 5. En la segunda iteración, se detectaron 7 no conformidades, incluyendo las 5 no resueltas en la iteración anterior, aquí se resolvieron 5 de ellas. En

## CAPÍTULO 3

la tercera iteración solo se encontró las 2 no conformidades que quedaron pendientes en la iteración anterior, las mismas fueron solucionadas satisfactoriamente.



*ILUSTRACIÓN 15 RESULTADO DE LAS PRUEBAS DE VALIDACIÓN (ELABORACIÓN PROPIA)*

# CAPÍTULO 3

## Conclusiones del capítulo

- Con el transcurso del proceso de implementación el uso de los estándares de codificación contribuyó a tener una adecuada uniformidad y legibilidad en el código fuente.
- El diagrama de componentes permitió la posibilidad de visualizar la dependencia entre los componentes del sistema, además, reflejó de manera más detallada la ubicación física de los componentes de acuerdo a la arquitectura empleada.
- La aplicación de las pruebas de software a la solución desarrollada permitió encontrar errores que afectaban el funcionamiento del módulo, posibilitando corregirlos a tiempo para que el mismo cumpliera totalmente con los requisitos funcionales definidos en la etapa de análisis.

# CONCLUSIONES GENERALES

## CONCLUSIONES GENERALES

La presente investigación tuvo como base el desarrollo de un módulo de cotización para el Sistema de Control de la Militancia del PCC (SICOM), para ello se dio cumplimiento a una serie de objetivos específicos, los cuales fueron cumplidos satisfactoriamente, por lo que se puede arribar a las siguientes conclusiones:

- La elaboración del marco teórico-metodológico que sustenta el desarrollo y utilización de un módulo de cotización para el Sistema de Control de la Militancia del PCC (SICOM) ratificó la necesidad de informatizar los procesos que se llevan a cabo en la organización.
- La obtención, descripción y validación de los requisitos definieron las características técnicas y funcionales de la propuesta de solución.
- A partir del análisis y diseño del módulo web se obtuvieron los artefactos necesarios para guiar el proceso de desarrollo del mismo.
- Las pruebas internas realizadas permitieron probar el correcto funcionamiento del módulo y los resultados satisfactorios de las mismas otorgaron validez a la presente investigación, siendo esto la base para que el módulo pueda favorecer los procesos que tienen lugar en el Comité Central del Partido Comunista de Cuba.

# RECOMENDACIONES

## RECOMENDACIONES

La autora de la presente investigación propone las siguientes recomendaciones:

- Agregarle al módulo la funcionalidad que permita importar nóminas.
- Agregarle al módulo funcionalidades que permitan realizar auditorías a la cotización.

# REFERENCIAS BIBLIOGRÁFICAS

## REFERENCIAS BIBLIOGRÁFICAS

**Universidad Politécnica de Valencia. 2016.** *Introducción a Herramientas CASE y System Architect.* 2016.

**acensTechnologies. 2015.** *Control de versiones: Git y GitHub.* 2015.

**Cabrera González, Lianet y Pompa Torres, Enrique Roberto. 2012.** *Extension de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información.* 2012.

**Cabrera, Robinson Danilo Chávez. 2016.** "ANÁLISIS DEL FRAMEWORK DJANGO PARA IMPLEMENTAR APLICACIONES WEB CON BASE DE DATOS MARIADB Y METODOLOGÍA DE DESARROLLO SCRUM. 2016.

**Cabrera, Robinson Danilo Chávez. 2016.** ANÁLISIS DEL FRAMEWORK DJANGO PARA IMPLEMENTAR APLICACIONES WEB CON BASE DE DATOS MARIADB Y METODOLOGÍA DE DESARROLLO SCRUM. . Ecuador : s.n., 2016.

**Calleja, Manuel Arias. 2015.** *Estándares de codificación.* 2015.

—. 2016. *Estándares de Codificación.* 2016.

**Casadiego, Jorge Manuel Pacheco. 2017.** *Metodologia para elaborar el Modelo Conceptual de Datos .* Colombia : s.n., 2017.

**Castelao, Paula Montoto. 2008.** *Introducción al Diseño con Patrones.* Universidad de Coruña : s.n., 2008.

**Cruz, Yunier Rodríguez. 2015.** *Gestión de la Información y del Conocimiento para la toma de desiciones organizacionales.* 2015.

**Cuba, Constitución de la República de. 2019.** *PCC.* 2019.

**Duarte, Gloria Ponjuán. 2016.** *Gestión de la Información: Presiciones Conceptuales a partir de sus orígenes.* 2016.

**Erick Nùñez Navarrete,Darío Rìos Navarro. 2017.** *Desarrollo de un marco de trabajo (framework) para el desarrollo de aplicaciones web en la Universidad Nacionalde CostaRica.* Costa Rica : s.n., 2017.

**Escalante, Lain Càrdenas. 2014.** *Aplicación de Patrones de Diseño para Garantizar Alta Flexibilidad en el Software.* 2014.

## REFERENCIAS BIBLIOGRÁFICAS

**Escalante, Mg.Lain Cárdenas. 2015.** *Aplicación de Patrones de Diseño para Garantizar Alta Flexibilidad en el Software.* . 2015.

Escuela de Python . <https://www.escuelapython.com/pycharm-uno-de-los-mejores-ide-para-python/>. [En línea] <https://www.escuelapython.com>.

**Española, Real Academia.** Cotización.

**Francisco José García Peñalvo,Alicia García Holgado . 2017.** *Fundamentos de la Vista de Casos de Uso* . Salamanca,España : s.n., 2017.

**Granma, Periódico. 2016.** Informe Central del 7mo. Congreso del PCC. *Periódico Granma.* 2016.

**Hernández, Coromoto León. 2006.** *Herramientas y Lenguajes de Programación.* 2006.

**Informáticas, Universidad de la Ciencias UCI. 2015.** *Metodología de desarrollo para la actividad productiva de la UCI.* 2015.

Jetbrains. [En línea] <https://www.jetbrains.com/pycharm/>.

**Larman, Craig.** UML y Patrones .

**Millàn, Martha Elena. 2016.** *Fundamentos de Bases de Datos* . Madrid : s.n., 2016.

**Patricia, López y Francisco, Ruiz. 2015.** *Lenguaje Unificado de Modelado - UML.* 2015.

**Paz, Juan Andres Mera. 2015.** *Análisis del proceso de pruebas.* 2015.

**Paz, Julián Andrés Mera. 2015.** *Análisis del Proceso de Pruebas de Calidad de Software.* 2015.

**PCC. 1999.** *Reglamentos del PCC.* 1999.

**PCC, 7mo Congreso del.** *Actualización de los Lineamientos de la política económica y social del Partido y la Revolución para el periodo 2016-2021.* 2016 : s.n.

**Peño, Jose Manuel Sanchez. 2015.** *Pruebas de Software.Fundamentos y Tecnicas* . 2015.

**Pressman, Roger S. 2010.** *Ingeniería del Software.* 2010.

**Python, Escuela de. 2019.** jetbrains. *Welcome to PyCharm help.* [En línea] 2019.  
<https://www.jetbrains.com/help/pycharm/2017.1/pycharm-2017.1-help.pdf>.

**RAE.** Cotización.



## REFERENCIAS BIBLIOGRÁFICAS

**Revista de Información, Tecnología y Sociedad . 2015.** *Phython – DjangoFramework de desarrollo web para perfeccionistas* Basado en el Modelo MTV. España : s.n., 2015.

**Rivera, Alcides Neptalí. 2010.** *Estudio de la Arquitectura de Software.* 2010.

**Rodríguez, Miguel Ángel Palacios. 2015.** *Gestión de Información como eje transversal para el éxito de las organizaciones.* 2015.

**Rolando, Molina Ríos Jimmy, Nancy, Loja Mora y Paola, Zea Ordóñez. 2016.** *Evaluación en los Frameworks en el desarrollo de Aplicaciones Web con Python.* 2016.

**Rolando, Molina Ríos Jimmy, Paola, Zea Ordóñez y Fabían, Castillo Fausto. 2017.** *Administración de Bases de Datos con PostgreSQL.* 2017.

**Rossum, Guido Van.** *Tutorial de Python.* Argentina : s.n.

**Ruíz, José Luis Enríquez, Flores, Eder Flores y Solano, Cesar Honores. 2017.** *Metodología de Desarrollo de Software.* 2017.

**Sánchez, Tamara Rodríguez. 2014.** *Metodología de desarrollo para la Actividad productiva de la UCI.* 2014.

**2019.** Seguridad Social.Sede Electrónica . *Especificaciones técnicas Sistema Red Internet .* [En línea] 2019. <http://www.seg-social.es/wps/portal/wss/internet/Inicio>.

**Sommerville, Ian. 2015.** *Requerimientos del software.* 2015.

—. **2011.** *Software Engineering.* 2011. 9.

**Suárez, Yuniel. 2016.** Doofy's Blog. *Qué es un IDE – DeProgramación.* [En línea] 2016.

**Subdirección General de Afiliación ,Cotización y Gestión del Sistema RED. 2018.** *Sistema Siltra.* s.l. : España, 2018.

**Subdirección General de Afiliación, Cotización y Gestión del Sistema RED. 2017.** *SILTRA Manual de Usuario .* España : s.n., 2017.

**Torres, Patricio Letelier. 2002.** *Desarrollo de Software Orientado a Objeto usando UML.* Departamento Sistemas Informáticos y Computación (DSIC) Universidad Politécnica de Valencia (UPV) - España : s.n., 2002.

# ANEXOS

## ANEXOS

### Anexo 1

TABLA 7 DESCRIPCIÓN CU GESTIONAR ESCALA PCC (ELABORACIÓN PROPIA)

<b>Nombre del Caso de Uso</b>	Gestionar Escala PCC
<b>Objetivos</b>	Administrar Escala PCC.
<b>Actores</b>	Estadístico Municipal
<b>Resumen</b>	<p>El Caso de Uso inicia cuando el Estadístico Municipal va a realizar alguna de las siguientes operaciones:</p> <ul style="list-style-type: none"><li>• <b>Insertar Escala PCC:</b> Cuando se va a insertar una Escala PCC existente, el Estadístico Municipal introduce los datos, así finaliza el Caso de Uso.</li><li>• <b>Editar Escala PCC:</b> Cuando el Estadístico Municipal necesita modificar los datos, modifica los datos de Escala PCC y actualiza la Base de Datos, así finaliza el Caso de Uso.</li><li>• <b>Eliminar Escala PCC:</b> Cuando un Estadístico Municipal necesita eliminar una Escala PCC, debe tener en cuenta que si hay militantes con esa escala asignado no podrá eliminarse, así finaliza el Caso de Uso.</li><li>• <b>Listar Escala PCC:</b> Cuando un Estadístico Municipal necesita listar la Escala PCC, y finaliza el Caso de Uso.</li></ul>
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítico

## ANEXOS

<b>Referencias</b>	RF5, RF6, RF7, RF8
<b>Precondiciones</b>	El Estadístico Municipal debe tener permisos de administrador.
<b>Postcondiciones</b>	En dependencia de la acción que el Estadístico Municipal desee realizar: <ul style="list-style-type: none"> <li>• Se Inserta un Escala PCC.</li> <li>• Se edita Escala PCC.</li> <li>• Se elimina Escala PCC.</li> <li>• Se listan Escala PCC.</li> </ul>

### Curso Normal de los Eventos

Acciones del Actor	Propuesta del Sistema
<p>1-El Estadístico Municipal selecciona cual acción desea realizar:</p> <ol style="list-style-type: none"> <li>5. Insertar Escala PCC.</li> <li>6. Modificar Escala PCC.</li> <li>7. Eliminar Escala PCC</li> <li>8. Listar Escala PCC.</li> </ol>	<p>2- En dependencia de la acción seleccionada por el Estadístico Municipal el Sistema muestra la interfaz correspondiente:</p> <p>Primeramente, el sistema muestra la opción Gestionar Escala PCC, donde se despliega una tabla que muestra las opciones que puede elegir el Estadístico Municipal. Estas son:</p> <ol style="list-style-type: none"> <li>4. Insertar una nueva Escala PCC: Ir a la sección <b>“insertar Escala PCC”</b>.</li> <li>5. Modificar datos de Escala PCC: Ir a la sección <b>“editar Escala PCC”</b>.</li> <li>6. Eliminar una Escala PCC: Ir a la sección <b>“eliminar Escala PCC”</b>.</li> </ol> <p>Listar Escala PCC: Ir a la sección <b>“listar Escala PCC”</b>.</p>

### Sección “Insertar Escala PCC”: Flujo Normal de Eventos

Acciones del Actor	Respuesta del Sistema
--------------------	-----------------------

## ANEXOS

7- El Estadístico Municipal selecciona la opción Insertar.	8- El sistema muestra la interfaz para insertar una Escala PCC.
9- El Estadístico Municipal inserta los datos de Escala PCC: nombre, ingreso inicial, ingreso final e importe. Indica al sistema almacenar la Escala PCC.	10- El sistema verifica que no exista ningún campo vacío.
	11- El sistema valida los datos insertados
	12- El sistema registra la nueva Escala PCC, así finaliza el Caso de Uso.
<b>Sección “Insertar Escala PCC”: Paso 4. Flujo Alternativo de Eventos: Existe campos vacíos</b>	
	4.1- El sistema muestra un mensaje: “Existen campos vacíos”.
<b>Sección “Editar Escala PCC”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del sistema</b>
7- El Estadístico Municipal selecciona la opción Editar.	8- El sistema ejecuta la acción “ <b>editar Escala PCC</b> ”.
9- El Estadístico Municipal edita los datos en Escala PCC que desea actualizar e indica guardar los cambios realizados.	10- El sistema valida completitud de los datos.
	11- El sistema actualiza la Escala PCC
<b>Sección “Editar Escala PCC”: Paso 3. Flujo Alternativo de Eventos: Cancelar Modificación</b>	
4.1- Si el Estadístico Municipal no indica guardar los cambios, cancela la operación y finaliza así el caso de uso.	
<b>Sección “Editar Escala PCC”: Paso 4. Flujo Alternativo de Eventos: Existen Campos Obligatorios vacíos</b>	
	4.1- El sistema muestra un mensaje “Existen campos vacíos” y va a la acción 3 de esta sección.

## ANEXOS

<b>Sección “Eliminar Escala PCC”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del sistema</b>
3- El Estadístico Municipal selecciona la Escala PCC que desee eliminar.	4- El sistema muestra la lista de escalas que se desea eliminar

*Tabla 8 Descripción CU. Gestionar cotización (Elaboración Propia)*

<b>Nombre del Caso de Uso</b>	Gestionar Cotización.
<b>Objetivos</b>	Administrar Cotización.
<b>Actores</b>	Estadístico Municipal
<b>Resumen</b>	<p>El Caso de Uso inicia cuando el Estadístico Municipal va a realizar alguna de las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>• <b>Insertar Cotización:</b> Cuando se va a insertar una Cotización, el Estadístico Municipal introduce o importa el salario de los militantes y calcula, de acuerdo a la escala establecida, la cuota que corresponde, así como inserta las causas por las cuales no cotizó. También debe alertar si el militante al que se le está insertando la cotización tiene cuotas atrasadas, es decir dejó de cotizar meses anteriores al actual, así finaliza el Caso de Uso.</li> <li>• <b>Editar Cotización:</b> Cuando el Estadístico Municipal necesita modificar los datos, modifica los datos de Cotización y actualiza la Base de Datos, así finaliza el Caso de Uso.</li> <li>• <b>Eliminar Cotización:</b> Cuando un Estadístico Municipal necesita eliminar una Cotización,</li> </ul>

## ANEXOS

	<p>debe tener en cuenta que solo podrá cancelar aquellas que no se les haya realizado el cierre, así finaliza el Caso de Uso.</p> <ul style="list-style-type: none"> <li>• <b>Listar Cotización:</b> Cuando un Estadístico Municipal necesita listar una Cotización, y finaliza el Caso de Uso.</li> </ul>
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítico
<b>Referencias</b>	RF13, RF14, RF15, RF16
<b>Precondiciones</b>	El Estadístico Municipal debe tener permisos de administrador.
<b>Postcondiciones</b>	<p>En dependencia de la acción que el Estadístico Municipal desee realizar:</p> <ul style="list-style-type: none"> <li>• Se Inserta una Cotización.</li> <li>• Se edita Cotización.</li> <li>• Se elimina Cotización.</li> <li>• Se listan Cotización.</li> </ul>
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Propuesta del Sistema</b>
<p>1-El Estadístico Municipal selecciona cual acción desea realizar:</p> <ul style="list-style-type: none"> <li>9. Insertar Cotización.</li> <li>10. Modificar Cotización.</li> <li>11. Eliminar Cotización.</li> <li>12. Listar Cotización.</li> </ul>	<p>2- En dependencia de la acción seleccionada por el Estadístico Municipal el Sistema muestra la interfaz correspondiente:</p> <p>Primeramente, el sistema muestra la opción Gestionar Cotización, donde se despliega una tabla que muestra las opciones que puede elegir el Estadístico Municipal. Estas son:</p> <ul style="list-style-type: none"> <li>7. Insertar una nueva Cotización: Ir a la sección <b>“insertar Cotización”</b>.</li> </ul>

## ANEXOS

	<p>8. Modificar datos de Cotización: Ir a la sección <b>“editar Cotización”</b>.</p> <p>9. Eliminar una Cotización: Ir a la sección <b>“eliminar Cotización”</b>.</p> <p>Listar Cotización: Ir a la sección <b>“listar Cotización”</b>.</p>
<b>Sección “Insertar Cotización”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
13- El Estadístico Municipal selecciona la opción Insertar.	14- El sistema muestra la interfaz para insertar una Cotización.
15- El Estadístico Municipal inserta los datos de Cotización: fecha de entrega, fecha de cotización, salario, importe, vuelto, no cotiza, pago pendiente, devuelto y retenido e indica al sistema almacenar Cotización.	16- El sistema verifica que no exista ningún campo vacío.
	17- El sistema valida los datos insertados
	18- El sistema registra la nueva Cotización, así finaliza el Caso de Uso.
<b>Sección “Insertar Cotización”: Paso 4. Flujo Alternativo de Eventos: Existe campos vacíos</b>	
	4.1- El sistema muestra un mensaje: “Existen campos vacíos”.
<b>Sección “Editar Cotización”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del sistema</b>
12- El Estadístico Municipal selecciona la opción Editar.	13- El sistema ejecuta la acción <b>“editar Cotización”</b> .
14- El Estadístico Municipal edita los datos en la Cotización que desea	15- El sistema valida completitud de los datos.

## ANEXOS

actualizar e indica guardar los cambios realizados.	
	16- El sistema actualiza la Cotización.
<b>Sección “Editar Cotización”: Paso 3. Flujo Alternativo de Eventos: Cancelar Modificación</b>	
4.1- Si el Estadístico Municipal no indica guardar los cambios, se cancela la operación y finaliza así el caso de uso.	
<b>Sección “Editar Cotización”: Paso 4. Flujo Alternativo de Eventos: Existen Campos Obligatorios vacíos</b>	
	4.1- El sistema muestra un mensaje “Existen campos vacíos” y va a la acción 3 de esta sección.
<b>Sección “Eliminar Cotización”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del sistema</b>
5- El Estadístico Municipal selecciona la Cotización que desee eliminar.	6- El sistema muestra el listado de cotizaciones que desea eliminar
<b>Sección “Listar Cotización”: Flujo Normal de Eventos</b>	
<b>Acciones del autor</b>	<b>Respuesta del sistema</b>
4- El Estadístico Municipal selecciona la opción <b>listar Cotización</b> .	5- El sistema ejecuta la acción “ <b>listar Cotización</b> ”.
	6- El sistema muestra la lista de las Cotizaciones existentes en la Base de Datos.

*TABLA 9 DESCRIPCIÓN CU. GESTIONAR CRONOGRAMA (ELABORACIÓN PROPIA)*

<b>Nombre del Caso de Uso</b>	Gestionar Cronograma.
<b>Objetivos</b>	Administrar Cronograma.
<b>Actores</b>	Estadístico Municipal
<b>Resumen</b>	El Caso de Uso inicia cuando el Estadístico Municipal va a realizar alguna de las siguientes operaciones: <ul style="list-style-type: none"> <li>• <b>Insertar Cronograma:</b> Cuando se va a</li> </ul>



## ANEXOS

	<p>insertar un Cronograma, el Estadístico Municipal introduce las fechas en las que les corresponde cotizar a los núcleos por meses., así finaliza el Caso de Uso.</p> <ul style="list-style-type: none"> <li>• <b>Editar Cronograma:</b> Cuando el Estadístico Municipal necesita modificar los datos, modifica las fechas en las que los núcleos deben cotizar y actualiza la Base de Datos, así finaliza el Caso de Uso.</li> <li>• <b>Eliminar Cronograma:</b> Cuando un Estadístico Municipal necesita eliminar un Cronograma de un núcleo seleccionado, debe tener en cuenta que no se haya cerrado el mes, así finaliza el Caso de Uso.</li> <li>• <b>Listar Cronograma:</b> Cuando un Estadístico Municipal necesita listar los Cronogramas, y finaliza el Caso de Uso.</li> </ul>
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítico
<b>Referencias</b>	RF17, RF18, RF19, RF20
<b>Precondiciones</b>	El Estadístico Municipal debe tener permisos de administrador.
<b>Postcondiciones</b>	<p>En dependencia de la acción que el Estadístico Municipal desee realizar:</p> <ul style="list-style-type: none"> <li>• Se Inserta un Cronograma.</li> <li>• Se edita Cronograma.</li> <li>• Se elimina Cronograma.</li> <li>• Se listan Cronogramas.</li> </ul>
<b>Curso Normal de los Eventos</b>	

## ANEXOS

Acciones del Actor	Propuesta del Sistema
<p>1-El Estadístico Municipal selecciona cual acción desea realizar:</p> <p>13. Insertar Cronograma. 14. Modificar Cronograma. 15. Eliminar Cronograma. 16. Listar Cronograma.</p>	<p>2- En dependencia de la acción seleccionada por el Estadístico Municipal el Sistema muestra la interfaz correspondiente:</p> <p>Primeramente, el sistema muestra la opción Gestionar Cronograma, donde se despliega una tabla que muestra las opciones que puede elegir el Estadístico Municipal. Estas son:</p> <p>10. Insertar un nuevo Cronograma: Ir a la sección <b>“insertar Cronograma”</b>. 11. Modificar datos de Cronograma: Ir a la sección <b>“editar Cronograma”</b>. 12. Eliminar un Cronograma: Ir a la sección <b>“eliminar Cronograma”</b>. 13. Listar Cotización: Ir a la sección <b>“listar Cronograma”</b>.</p>
<b>Sección “Insertar Cronograma”: Flujo Normal de Eventos</b>	
Acciones del Actor	Respuesta del Sistema
19- El Estadístico Municipal selecciona la opción Insertar.	20- El sistema muestra la interfaz para insertar un Cronograma.
21- El Estadístico Municipal inserta los datos de Cronograma: núcleo y fecha e indica al sistema almacenar Cronograma.	22- El sistema verifica que no exista ningún campo vacío.
	23- El sistema valida los datos insertados.
	24- El sistema registra el nuevo Cronograma, así finaliza el Caso de Uso.
<b>Sección “Insertar Cronograma”: Paso 4. Flujo Alternativo de Eventos: Existe campos vacíos</b>	
	4.1- El sistema muestra un mensaje: “Existen campos vacíos”.

## ANEXOS

<b>Sección “Editar Cronograma”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del sistema</b>
17- El Estadístico Municipal selecciona la opción Editar.	18- El sistema ejecuta la acción <b>“editar Cronograma”</b> .
19- El Estadístico Municipal edita los datos en el Cronograma que desea actualizar e indica guardar los cambios realizados.	20- El sistema valida completitud de los datos.
	21- El sistema actualiza el Cronograma.
<b>Sección “Editar Cronograma”: Paso 3. Flujo Alternativo de Eventos: Cancelar Modificación</b>	
4.1- Si el Estadístico Municipal no indica guardar los cambios, se cancela la operación y finaliza así el caso de uso.	
<b>Sección “Editar Cronograma”: Paso 4. Flujo Alternativo de Eventos: Existen Campos Obligatorios vacíos</b>	
	4.1- El sistema muestra un mensaje “Existen campos vacíos” y va a la acción 3 de esta sección.
<b>Sección “Eliminar Cronograma”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del sistema</b>
7- El Estadístico Municipal selecciona el Cronograma que desee eliminar.	8- El sistema la lista de cronogramas que desea eliminar.
<b>Sección “Listar Cronograma”: Flujo Normal de Eventos</b>	
<b>Acciones del autor</b>	<b>Respuesta del sistema</b>
7- El Estadístico Municipal selecciona la opción <b>listar Cronograma</b> .	8- El sistema ejecuta la acción <b>“listar Cronograma”</b> .
	9- El sistema muestra la lista de los Cronogramas existentes en la Base de Datos.

## ANEXOS

TABLA 10 DESCRIPCIÓN CU. GESTIONAR PLAN DE INGRESO (ELABORACIÓN PROPIA)

<b>Nombre del Caso de Uso</b>	Gestionar Plan de Ingreso.
<b>Objetivos</b>	Administrar Plan de Ingreso.
<b>Actores</b>	Estadístico Municipal
<b>Resumen</b>	<p>El Caso de Uso inicia cuando el Estadístico Municipal va a realizar alguna de las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>• <b>Insertar Plan de Ingreso:</b> Cuando se va a insertar un Cronograma, el Estadístico Municipal introduce los datos correspondientes a Plan de Ingreso, así finaliza el Caso de Uso.</li> <li>• <b>Editar Plan de Ingreso:</b> Cuando el Estadístico Municipal necesita modificar los datos, modifica los datos correspondientes a un Plan de Ingreso y actualiza la Base de Datos, así finaliza el Caso de Uso.</li> <li>• <b>Eliminar Plan de Ingreso:</b> Cuando un Estadístico Municipal necesita eliminar un Plan de Ingreso seleccionado luego de previa confirmación, así finaliza el Caso de Uso.</li> <li>• <b>Listar Plan de Ingreso:</b> Cuando un Estadístico Municipal necesita listar los Plan de Ingreso, y finaliza el Caso de Uso.</li> </ul>
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítico
<b>Referencias</b>	RF21, RF22, RF23, RF24
<b>Precondiciones</b>	El Estadístico Municipal debe tener permisos de administrador.

## ANEXOS

<b>Postcondiciones</b>	<p>En dependencia de la acción que el Estadístico Municipal desee realizar:</p> <ul style="list-style-type: none"> <li>• Se Inserta un Plan de Ingreso.</li> <li>• Se edita Plan de Ingreso.</li> <li>• Se elimina Plan de Ingreso.</li> <li>• Se listan Plan de Ingreso.</li> </ul>
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Propuesta del Sistema</b>
<p>1-El Estadístico Municipal selecciona cual acción desea realizar:</p> <ul style="list-style-type: none"> <li>17. Insertar Plan de Ingreso.</li> <li>18. Modificar Plan de Ingreso.</li> <li>19. Eliminar Plan de Ingreso.</li> <li>20. Listar Plan de Ingreso.</li> </ul>	<p>2- En dependencia de la acción seleccionada por el Estadístico Municipal el Sistema muestra la interfaz correspondiente:</p> <p>Primeramente, el sistema muestra la opción Gestionar Cronograma, donde se despliega una tabla que muestra las opciones que puede elegir el Estadístico Municipal. Estas son:</p> <ul style="list-style-type: none"> <li>14. Insertar un nuevo Plan de Ingreso: Ir a la sección <b>“insertar Plan de Ingreso”</b>.</li> <li>15. Modificar datos de Plan de Ingreso: Ir a la sección <b>“editar Plan de Ingreso”</b>.</li> <li>16. Eliminar un Plan de Ingreso: Ir a la sección <b>“eliminar Plan de Ingreso”</b>.</li> <li>17. Listar Plan de Ingreso: Ir a la sección <b>“listar Plan de Ingreso”</b>.</li> </ul>
<b>Sección “Insertar Plan de Ingreso”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
<p>25- El Estadístico Municipal selecciona la opción Insertar.</p>	<p>26- El sistema muestra la interfaz para insertar un Plan de Ingreso.</p>

## ANEXOS

27- El Estadístico Municipal inserta los datos de Plan de Ingreso: e indica al sistema almacenar Plan de Ingreso.	28- El sistema verifica que no exista ningún campo vacío.
	29- El sistema valida los datos insertados.
	30- El sistema registra el nuevo Plan de Ingreso, así finaliza el Caso de Uso.
<b>Sección “Insertar Plan de Ingreso”: Paso 4. Flujo Alternativo de Eventos: Existe campos vacíos</b>	
	4.1- El sistema muestra un mensaje: “Existen campos vacíos”.
<b>Sección “Editar Plan de Ingreso”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del sistema</b>
22- El Estadístico Municipal selecciona la opción Editar.	23- El sistema ejecuta la acción <b>“editar Plan de Ingreso”</b> .
24- El Estadístico Municipal edita los datos en el Plan de Ingreso que desea actualizar e indica guardar los cambios realizados.	25- El sistema valida completitud de los datos.
	26- El sistema actualiza el Plan de Ingreso.
<b>Sección “Editar Plan de Ingreso”: Paso 3. Flujo Alternativo de Eventos: Cancelar Modificación</b>	
4.1- Si el Estadístico Municipal no indica guardar los cambios, se cancela la operación y finaliza así el caso de uso.	
<b>Sección “Editar Plan de Ingreso”: Paso 4. Flujo Alternativo de Eventos: Existen Campos Obligatorios vacíos</b>	
	4.1- El sistema muestra un mensaje “Existen campos vacíos” y va a la acción 3 de esta sección.
<b>Sección “Eliminar Plan de Ingreso”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del sistema</b>

## ANEXOS

9- El Estadístico Municipal selecciona el Plan de Ingreso que desee eliminar.	10- El sistema muestra el listado de plan de ingreso que desea eliminar.
<b>Sección “Listar Plan de Ingreso”: Flujo Normal de Eventos</b>	
<b>Acciones del autor</b>	<b>Respuesta del sistema</b>
10-El Estadístico Municipal selecciona la opción <b>listar Plan de Ingreso</b> .	11-El sistema ejecuta la acción “ <b>listar Plan de Ingreso</b> ”.
	12- El sistema muestra la lista de los Plan de Ingreso existentes en la Base de Datos.

# ANEXOS

## Anexo 2

```
def CotizacionCreate(request, pk): 1
    militante = get_object_or_404(Militante, pk=pk) 1
    listaDebe = MilitanteDebe.objects.filter(militantId=pk) 1
    debeTotal = 0 1
    if len(listaDebe) > 0: 1
        for control in listaDebe: 2
            debeTotal += control.importe 2
        escalas = EscalaPCC.objects.all() 3
        nombreEscala = " 3
        nucleo = militante.nucleo 3
        salario = militante.salario 3
        for var in escalas: 3
            if salario >= var.ingreso_inicial and salario <= var.ingreso_final: 3
                nombreEscala = var.nombre 4
            escalaAplicar = get_object_or_404(EscalaPCC, nombre=nombreEscala) 5
            importeCalculado = salario - (salario - escalaAplicar.importe) 5
            importeCalculado += debeTotal 5
            form = CotizacionForm(
                initial={'militante': militante, 'nucleo': nucleo, 'salario': salario, 'importe': importeCalculado,
                    'pagoPendiente': debeTotal}) 5

    if request.method == 'POST': 6
        form = CotizacionForm(request.POST) 6
        if form.is_valid(): 6
            militanteCotizo = Cotizacion.objects.filter(militante__id=pk) 7
            fecha = form.instance.fechaEntrega 7
            fechaCotizacion = form.instance.fechaCotizacion 7
            bandera1 = False
            for var in militanteCotizo: 8
                if var.fechaEntrega.month == fecha.month: 8
                    bandera1 = True
            if bandera1:
                messages.error(request, 'El militante ya se le registro una cotización para este mes. Verifique!!!')
            elif fecha < fechaCotizacion: 8
                messages.error(request, 'La fecha de entrega no puede ser menor que la fecha de cotización. Verifique!!!')
            else:
                #Metodo para eliminar las deudas del militante si paga.
                mil = MilitanteDebe.objects.filter(militantId=pk) 8
                for var in mil: 8
                    var.delete() 8
                .. 8
```

ILUSTRACIÓN 16 MÉTODO CAJA BLANCA COTIZACIONCREATE (ELABORACIÓN PROPIA)



# ANEXOS

## Anexo 3

The image shows a screenshot of an IDE with two tabs: 'settings.py' and 'tests.py'. The 'tests.py' tab is active, displaying the following Python code:

```
1 from django.test import TestCase
2
3 # Create your tests here.
4 from django.core.urlresolvers import reverse
5 from django.contrib.auth.models import User
6
7
8 class CotizacionTest(TestCase):
9     def setUp(self):
10         test_user1 = User.objects.create_user(username='duriet', password='12345')
11         test_user1.save()
12
13     def test_listar_escalapcc_cotizacion_page(self):
14         login = self.client.login(username='duriet', password='12345')
15         resp = self.client.get(reverse('URL_EscalaPCC:listar_escalapcc'))
16         self.assertEqual(str(resp.context['user']), 'duriet')
17         self.assertEqual(resp.status_code, 200)
18         self.assertTemplateUsed(resp, 'EscalaPCC/EscalaPCCListar.html')
19
20     def test_listar_nucleo_page(self):
21         login = self.client.login(username='duriet', password='12345')
22         resp = self.client.get(reverse('URL_Cotizacion:listar_nucleo'))
23         self.assertEqual(str(resp.context['user']), 'duriet')
24         self.assertEqual(resp.status_code, 200)
25         self.assertTemplateUsed(resp, 'Cotizacion/NucleoListar.html')
```

Below the code editor, the IDE shows the execution of the tests. The output window displays the following information:

- Run: SGC x Test: apps.Cotizacion.tests.CotizacionTest x
- Test Results: 329 ms
- Tests passed: 2 of 2 tests - 329 ms
- Testing started at 7:57 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

ILUSTRACIÓN 17 PRUEBAS DE UNIDAD. PYUNIT

# ANEXOS

## Anexo 4

**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 CCPCC y en función de la ejecución del proyecto **SICOM**, se hace entrega de los productos que se relacionan a continuación:

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

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

Comentarios:

Observaciones:

<b>Entrega</b>	<b>Recibe</b>
Nombre y apellidos: Duriet Aguilera Alvarez.	Nombre y apellidos: Osay González Fuentes
Cargo: Programador	Cargo: Director del Centro de Identificación y Seguridad Digital, UCI

Fecha: 21/05/19