

Universidad de las Ciencias Informáticas

Facultad 6



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

Título:

“Sistema de Gestión de Ingresos por concepto de Cotización para la Unión de Jóvenes Comunistas (UJC)”

Autores: Indyra Fernández Pérez

Marisleidis Leonor Rendón Núñez

Tutor: Ing. Yunior Mesa Reyes

Co-tutor: Ing. Rolando Enrique Fuentes Orosco

La Habana, Julio 2016

“Año 58 de la Revolución”



"Cuando quieres realmente una cosa, todo el Universo conspira para ayudarte a conseguirla"

Paulo Coelho

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Indyra Fernández Pérez

Firma del autor

Marisleidis Leonor Rendón Núñez

Firma del autor

Ing. Yunior Mesa Reyes

Firma del tutor

Ing. Rolando Enrique Fuentes Orozco

Firma del Co-tutor

Datos de contacto

Tutor:

Ing. Yunior Mesa Reyes

Especialidad de graduación: Ingeniero en Ciencias Informáticas

Categoría docente: Instructor (*Aspirante a Asistente*)

Categoría científica: Ninguna (*Aspirante a Master*)

Años de experiencia en el tema: 6

Años de graduado: 6

Correo electrónico: ymreyes@uci.cu

Co-tutor:

Ing. Rolando Enrique Fuentes Orozco

Especialidad de graduación: Ingeniero en Ciencias Informáticas

Categoría docente: Recién graduado

Correo electrónico: refuentes@uci.cu

Agradecimientos

De Indyra:

De Marisleidis:

Dedicatoria

Indyra:

Marisleidis:

Resumen

La Unión de Jóvenes Comunistas (UJC) es una de las organizaciones cubanas que genera a diario gran cúmulo de información en el desarrollo de sus procesos. Uno de ellos es la gestión de ingresos por concepto de cotización, se realiza de forma mensual y consiste en el aporte económico individual de cada uno de sus miembros. La gestión de este proceso la lleva a cabo el Jefe del Departamento de Control (JDC) de forma manual mediante el uso de formatos convencionales. Esto trae como consecuencia que la información que se analiza a través de los reportes que se puedan elaborar no sea totalmente confiable y actualizada, imposibilitando consultar en el momento que se necesiten estos reportes. Por tal motivo la presente investigación aborda el desarrollo de un sistema que permita mostrar en tiempo real la información obtenida del proceso, posibilitando así una correcta y oportuna toma de decisiones por parte de los organismos superiores. Para el desarrollo de esta investigación se utilizaron los métodos empíricos entrevista y análisis de documentos mientras que se empleó la modelación y el analítico-sintético como métodos teóricos. La metodología de desarrollo de software OpenUP guió el desarrollo de la solución propuesta, utilizando como marcos de trabajo Symfony en su versión 2.8.4 y ExtJS 3.4, PostgreSQL 9.5 como gestor de base datos y NetBeans IDE 8.0 como entorno de desarrollo.

Palabras claves: cotización, proceso de gestión de ingresos, toma de decisiones.

Abstract

The UJC (Unión de Jóvenes Comunistas) is one of the Cuban organizations that generate a great amount of information daily on the development of processes and activities. One of these processes is managing income by quotation concept, which is done monthly and consist of a single economic contribution of each of its members. The management of this process is performed by the Head of Control Department manually and through conventional formats. This brings consequences as that analyzed information through reports that maybe developed empirically is not completely reliable and updated, being impossible the consult in the time that these reports are needed. Therefore, this research addresses essential aspects of the development of a system that allows to show, in real time, the information acquired from the process, thus enabling a correct and timely decision making by higher organisms. For the development of this proposal empirical methods like interview, analytical-synthetic, document analysis, and modeling like theoretical method were used. It was selected as software development methodology OpenUP, Symfony 2.8.4 as framework and ExtJS 3.4, PostgreSQL 9.5 as database manager and as an integrated development environment NetBeans IDE 8.0.

Keywords: *quotation, revenue management process, decision making.*

Índice

Introducción	1
Capítulo 1: Fundamentación teórica del Sistema de Gestión de Ingresos por concepto de Cotización para la UJC	6
1.1 Proceso de gestión de información	6
1.1.1. Los sistemas de gestión de información	6
1.1.2. Ventajas del uso de los sistemas de gestión de información	7
1.2 Proceso de gestión de ingresos por concepto de cotización	7
1.2.1 Procesos que se realizan en la UJC y se relacionan con la cotización	10
1.2.3 Reportes que se generan para el análisis de la cotización y sus características	12
1.3 Sistemas informáticos vinculados al campo de acción	13
Conclusiones de los sistemas estudiados:	14
1.4 Metodología de desarrollo del software	14
1.4.1 OpenUP	14
1.5 Lenguaje de Modelado	15
1.5.1 UML 2.0	15
1.6 Herramienta CASE	16
1.6.1 Visual Paradigm para UML 8.0	16
1.7 Lenguajes de programación	17
1.7.1 PHP 5.6	17
1.8 Entorno de desarrollo	17
1.8.1 NetBeans IDE 8.0	18
1.9 Gestor de bases de datos	18
1.9.1 PostgreSQL 9.3	18
1.9.2 PgAdmin III v1.22.0	19
1.10 Marcos de trabajo	19
1.10.1 Symfony 2.8.4	20
1.10.2 ExtJS 3.4	20
1.11 Servidor Web	21
1.11.1 Apache 2.4	21
Conclusiones Parciales	22
Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por concepto de Cotización para la UJC ...	23
2.1 Descripción de la solución propuesta	23
2.2 Modelo de dominio	23
2.2.1 Descripción de las clases del modelo de dominio	24
2.3 Especificación de los requisitos de software	25
2.3.1 Requisitos Funcionales	26
2.3.2 Requisitos no funcionales	27
2.4 Modelación del sistema	29
2.4.1 Descripción de los actores del sistema	30

2.4.2 Casos de Uso	30
2.4.3 Diagrama de Casos de Uso del sistema.....	30
2.4.4 Patrones de Casos de Uso utilizados.....	31
2.4.5 Relación entre los Requisitos Funcionales y los Casos de Uso.....	32
2.4.6 Descripción de los Casos de Uso del sistema.....	33
2.5 Arquitectura de software propuesta.....	40
2.5.3 Patrón arquitectónico utilizado	40
2.5.2 Modelo físico de la base de datos.....	41
2.5.3 Diagrama de clases del diseño	42
2.5.4 Patrones de diseño utilizados en la solución	43
2.5.5 Patrones GRASP	43
2.5.6 Patrones GOF	45
2.5.7 Diagrama de despliegue.....	46
Conclusiones Parciales	47
Capítulo 3. Implementación y prueba del Sistema de Gestión de Ingresos por concepto de Cotización para la UJC.....	48
3.1 Modelo de Implementación	48
3.1.1 Diagrama de componentes	48
3.2 Estándares de codificación	49
3.3 Pruebas del software	50
3.3.1 Prueba de caja blanca	50
3.3.2 Prueba de caja negra.....	52
3.3.3 Pruebas de aceptación.....	54
3.4 Resultados de las pruebas	54
Conclusiones Parciales	55
Conclusiones Generales.....	56
Recomendaciones	57
Referencias Bibliográficas.....	58
Bibliografía	62
Anexos	66
Anexo1:	66

Índice de Figuras

Fig. 1 Modelo de dominio (Fuente: Elaboración propia)	24
Fig. 2 Diagrama de Casos de Uso (Fuente: Elaboración propia)	32
Fig. 3 Representación del Patrón Arquitectónico Modelo Vista Controlador (Alvarez, 2014).....	42
Fig. 4 Modelo físico de la Base de Datos (Fuente: Elaboración propia)	43
Fig. 5 Diagrama de clases del diseño del CU Administrar información del militante (Fuente: Elaboración propia)	44
Fig. 6 Fragmento de código donde se evidencia la utilización del patrón GRASP: Creador (Fuente: Elaboración propia).....	45
Fig. 7 Fragmento de código donde se evidencia la utilización del patrón GOF: Iterador (Fuente: Elaboración propia).....	47
Fig. 8 Diagrama de despliegue (Fuente: Elaboración propia).....	48
Fig. 9 Diagrama de componentes (Fuente: Elaboración propia)	51
Fig. 10 Código del método <i>listarCotizanComiteBaseAction</i> (Fuente: Elaboración propia).....	53
Fig. 11 Grafo de flujo del método <i>listarCotizanComiteBaseAction</i> (Fuente: Elaboración propia).....	53
Fig. 12 Resultados de la prueba de caja negra (Fuente: Elaboración propia).....	57

Índice de Tablas

Tabla 1: Escala de pago de cotización vigente aprobada por el Buró Nacional (Buró Nacional, 2013)	8
Tabla 2. Descripción de los actores del sistema (Fuente: Elaboración propia).....	31
Tabla 3. Relación de los CU y los RF (Fuente: Elaboración propia)	33
Tabla 4. Descripción del CU Administrar información del militante (Fuente: Elaboración propia)	34
Tabla 5. Caso de prueba para el camino básico No.1 (Fuente:Elaboración propia).....	54
Tabla 6. Caso de prueba para el camino básico No.2 (Fuente:Elaboración propia).....	54
Tabla 7. Caso de prueba para el camino básico No.3 (Fuente:Elaboración propia).....	54
Tabla 8. Caso de prueba del CU5 Administrar información del militante (Fuente:Elaboración propia)	55

Introducción

Desde la antigüedad la gestión de la información ha sido de gran importancia para el conocimiento y administración de las diferentes empresas y organizaciones. Con el auge de las Tecnologías de la Informática y las Comunicaciones (TIC) surgen los sistemas para la gestión de la información utilizados en áreas, organismos y empresas. Estos sistemas permiten garantizar la confidencialidad, integridad y disponibilidad de la información, convirtiéndose así en un elemento fundamental para el control y toma de decisiones.

En Cuba existen empresas y organizaciones que producto a sus actividades generan un gran conjunto de información, la cual va representando su actuar diario e histórico. Una de estas organizaciones es la Unión de Jóvenes Comunistas (UJC), organización política de vanguardia de la juventud cubana. Es la organización juvenil del Partido Comunista de Cuba (PCC) y la principal cantera para el ingreso a sus filas, forjada en la concepción Marxista-Leninista, las ideas y la práctica del pensamiento de Maceo, Martí, Mella, Camilo, el Che y Fidel (UJC, 2014).

La UJC asume como misión: contribuir a la educación comunista de las nuevas generaciones (...), propiciando la participación creadora, consciente y entusiasta de todos los niños, adolescentes y jóvenes en la construcción del Socialismo, expresada en el estudio, el trabajo y la defensa de la Patria (UJC, 2013). Por la importancia en el cumplimiento de su misión es una de las organizaciones que se encuentra presupuestada por el gobierno cubano. No obstante, la UJC realiza acciones y procesos que le permiten generar ingresos.

Entre los procesos que realiza la UJC para generar ingresos se encuentran los ingresos por servicio de alojamiento, los obtenidos por la venta de periódicos, publicaciones y litografías de la Casa Editora Abril¹ y el proceso de gestión de ingresos por concepto de cotización. Este último se realiza de forma mensual y consiste en el aporte económico individual de sus miembros para contribuir a sufragar los gastos de la organización en el cumplimiento de su misión. Incide en la formación, responsabilidad, disciplina, colectivismo y sentido de pertenencia de cada militante. Con esta finalidad el Buró Nacional establece en los Estatutos de la UJC, el deber del militante de abonar mensualmente la cotización, en correspondencia a sus ingresos, teniendo en cuenta que el que infrinja los principios y normas de la UJC, deberá ser analizado y en consecuencia con la gravedad de sus actos sancionado, cumpliendo su sanción dentro o fuera de la Organización.

¹ Formada en 1980 con la unión de varias publicaciones juveniles que ya existían (Pionero, El Caimán Barbudo, Alma Mater, Somos Jóvenes y Juventud Técnica, entre otras revistas).

Por lo antes expuesto, las organizaciones de base y organismos de dirección de la UJC asegurarán, con su labor política, que se comprenda por sus integrantes la importancia de este aporte. De este modo, el militante asume la responsabilidad de cotizar con ingresos salariales o no, desde los estudiantes hasta los trabajadores, a los que se les tendrá en cuenta las diferentes situaciones que se les pueda presentar y el total de ingreso adquirido en el mes, asumiéndolo como un principio organizativo y de disciplina consiente (Comité Nacional, 2014).

Para realizar dicho proceso el Comité de Base (CB), como célula fundamental de la Organización, designa un activista de cotización que exige el pago de la cuota establecida, en el tiempo requerido y los montos correspondientes. En el desarrollo del proceso, él mismo actualiza la planilla de la cotización en la que refleja el listado de los militantes, los salarios percibidos mensualmente, las cuotas que se abonan y sus firmas.

Al despachar la planilla y el monto correspondiente se debe tener en cuenta las cuotas atrasadas, para elaborar con posterioridad un resumen mensual de cotización. A partir de este despacho realizado al organismo superior, se crean resúmenes y reportes en documentos de extensiones .xls y .doc. Para el completo análisis de la información que se ha de almacenar en estos resúmenes y reportes mensuales, el Jefe del Departamento de Control (JDC) debe considerar no solo el resultado del proceso de gestión de ingresos por cotización, sino también la relación del mismo con altas y bajas realizadas en el CB. Todo este ciclo para controlar el estado del pago de la cotización por parte de la militancia se lleva a cabo a través de formatos convencionales².

La información que se registra en los reportes se genera con el objetivo de perfeccionar el control sobre los ingresos, permitiendo realizar los análisis y valoraciones políticas correspondientes para el tratamiento a los principales indicadores y que se tomen acciones oportunas ante los incumplimientos por parte de los militantes. La consulta de los mismos se realiza de forma individual y manual, lo que obstaculiza que la información se utilice de forma simultánea. También se torna tedioso y lento consultar las fuentes para obtener cualquier información y generar otras, siendo los datos históricos los de más difícil acceso.

Lo anteriormente explicado trae como consecuencia que la información que se analiza a través de los reportes que se puedan elaborar empíricamente no sea totalmente confiable y actualizada, pudiendo tener algún margen de error producto al propio trabajo humano. Dificulta también la disponibilidad de la información al personal que la necesite en el momento y lugar necesarios. Esto implica que el JDC tenga un alto nivel de abstracción de cómo se lleva a cabo dicho proceso y no pueda determinar correctamente

² Formatos duros que pueden ser impresos o escritos.

el estado de cotización de un militante en cualquier momento. Dicha situación constituye una de las problemáticas que enfrentan a diario los Departamentos de Control, existiendo un cúmulo de incumplimientos por parte de los militantes en el momento del Cierre del Estado de la Militancia mensual, limitando una toma de decisiones oportuna.

Atendiendo a la situación antes descrita, se plantea como **problema de investigación**: el procesamiento manual de la información referente al proceso de gestión de ingresos por concepto de cotización de la UJC, limita la generación de reportes en tiempo real, teniendo como **objeto de estudio** el proceso de desarrollo de los sistemas de gestión de información para los procesos en las organizaciones, enmarcado al **campo de acción** sistemas de gestión de información asociados al proceso de gestión de ingresos por concepto de cotización en la Unión de Jóvenes Comunistas. Se plantea como **objetivo general** desarrollar un sistema que permita gestionar la información referente a los ingresos por concepto de cotización de la UJC en tiempo real.

A partir del objetivo planteado se definen las siguientes **preguntas científicas**:

- ❖ ¿Cuáles son las bases teóricas que fundamentan el proceso de gestión de ingresos por concepto de cotización en la Unión de Jóvenes Comunistas?
- ❖ ¿Cuáles son las características y capacidades que debe tener el sistema a desarrollar para que gestione los ingresos por concepto de cotización para la Unión de Jóvenes Comunistas?
- ❖ ¿Cuáles son los principios arquitectónicos y de diseño del Sistema de Gestión de Ingresos por concepto de Cotización para definir su estructura global?
- ❖ ¿Cómo traducir el diseño elaborado en un producto funcional?
- ❖ ¿Cómo validar el correcto funcionamiento del Sistema de Gestión de Ingresos por concepto de Cotización para la Unión de Jóvenes Comunistas?

Para cumplir con el objetivo planteado se proponen las siguientes **tareas de investigación**:

- ❖ Análisis de los elementos conceptuales que fundamentan el desarrollo del Sistema de Gestión de Ingresos por concepto de Cotización para la Unión de Jóvenes Comunistas, para elaborar el marco teórico de la investigación.
- ❖ Caracterización de la metodología, herramientas y tecnologías para el desarrollo del Sistema de Gestión de Ingresos por concepto de Cotización para la Unión de Jóvenes Comunistas.

- ❖ Identificación y descripción de los requisitos funcionales con el fin de evidenciar el análisis de la solución.
- ❖ Definición de los principios arquitectónicos y de diseño del Sistema de Gestión de Ingresos por concepto de Cotización para definir su estructura global.
- ❖ Implementación del diseño propuesto para la solución, en función de los principios arquitectónicos definidos para satisfacer las necesidades funcionales y tecnológicas del Sistema de Gestión de Ingresos por concepto de Cotización para la Unión de Jóvenes Comunistas.
- ❖ Realización de las pruebas de software para comprobar el correcto funcionamiento del Sistema de Gestión de Ingresos por concepto de Cotización para la Unión de Jóvenes Comunistas.

Para el desarrollo de la investigación se utilizaron los siguientes métodos:

Métodos Teóricos:

- ❖ Modelación: Permitió modelar la estructura, relaciones internas y características de la solución a través de diagramas, mediante la utilización del lenguaje de modelado UML, con el objetivo de representar la realidad objetiva en que se enmarca el sistema.
- ❖ Analítico-Sintético: Se realizó un estudio de los flujos de trabajo, características y procesos de la UJC, permitiendo identificar y sintetizar los elementos teóricos necesarios para el desarrollo de la investigación.

Métodos Empíricos:

- ❖ Entrevista: Se realizó una entrevista a dirigentes de la UJC nacional para determinar las deficiencias del proceso de gestión de ingresos por concepto de cotización, cómo se realiza, las relaciones del mismo con otros procesos de la UJC y los reportes y resúmenes que se generan en los organismos superiores para la toma de decisiones.
- ❖ Análisis de documentos: Permitió analizar reportes y resúmenes que se generan del proceso de cotización, permitiendo un mayor entendimiento del objeto de estudio con el fin de obtener una solución satisfactoria para el problema planteado. Se utilizó con el objetivo de obtener información mediante la recolección y selección de documentos relacionados con el cierre contable de la cotización mensual y con los resúmenes solicitados por los organismos superiores.

La estructura capitular de la presente investigación científica se propone de la siguiente manera:

Capítulo 1. Fundamentación teórica del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC: Se realiza un análisis de los conceptos y sistemas asociados a la gestión de información, los procesos que se realizan en la UJC que tienen relación directa con la cotización y los reportes generados a partir de este proceso, siendo estos necesarios para la comprensión de la solución del problema planteado. Se define la metodología, las herramientas y tecnologías que se utilizan para el desarrollo del “Sistema de Gestión de Ingresos por concepto de Cotización para la UJC”.

Capítulo 2. Análisis y diseño del Sistema de Gestión de Ingresos por concepto de Cotización para la UJC: En el presente capítulo se describe el proceso llevado a cabo durante el análisis y diseño, se definen los requisitos funcionales y no funcionales con que debe cumplir el sistema y la descripción de la arquitectura a utilizar. Se muestra la estructura del sistema a desarrollar a través de la representación visual de clases conceptuales en el modelo de dominio, así como las funciones en la solución a través del diagrama de casos de uso.

Capítulo 3. Implementación y prueba del Sistema de Gestión de Ingresos por concepto de Cotización para la UJC: Se describe el modelo de implementación a partir del diseño realizado por cada uno de los casos de uso identificados. Se describen las pruebas a realizar, con el objetivo de comprobar la correcta implementación de cada una de las funcionalidades definidas, verificando que los resultados de las pruebas sean los esperados.

Capítulo 1: Fundamentación teórica del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

Capítulo 1: Fundamentación teórica del Sistema de Gestión de Ingresos por concepto de Cotización para la UJC

En el presente capítulo se abordan los aspectos principales que se utilizan como base para la solución del problema planteado, analizando los conceptos relacionados con el objeto de estudio y el campo de acción. Se realiza el estudio de algunas de las soluciones existentes y se describen la metodología, herramientas y tecnologías a utilizar para el desarrollo de la solución.

1.1 Proceso de gestión de información

Según (Woodman, 2009) “la gestión de información es todo lo que tiene que ver con obtener la información correcta, en la forma adecuada, para la persona indicada, al costo correcto, en el momento oportuno, en el lugar indicado para tomar la acción precisa”. Observándola desde otro punto de vista (Moreiro, 1998) lo define como “el conjunto de políticas y normas relacionadas entre sí que se establecen para el acceso y tratamiento de los recursos de información, incluye los registros administrativos y los archivos, el soporte tecnológico de los recursos y el público a que se destina. En su evolución el sistema puede manejar la función de inteligencia corporativa y generar productos de inteligencia”. Por lo que se concluye que la gestión de la información es el proceso, tratamiento y obtención de la información correcta, para la persona que la necesite, en el tiempo que la necesite y al menor costo posible para su uso y beneficio. El objetivo que se persigue al gestionar la información es maximizar el valor y los beneficios, minimizar el costo de la obtención, procesamiento y uso de la misma, determinar responsabilidades para su uso efectivo y asegurar su suministro continuo.

Según (Bartle, 2011) para poder utilizar la información en la toma de decisiones, ésta debe gestionarse. Este proceso implica determinar la información que se precisa, recogerla y analizarla, registrarla y recuperarla cuando sea necesario, utilizarla y divulgarla. Para ello hay que saber qué información y de donde se necesita recabar, para tomar diferentes decisiones en distintos momentos, guardarla para que se pueda recuperar y utilizarla para solucionar problemas o planear proyectos futuros.

1.1.1. Los sistemas de gestión de información

Los sistemas de información en las empresas eran considerados una herramienta que permitía facilitar los procesos a nivel operativo. Con el desarrollo de las TIC se pudo incrementar la eficacia en la realización de las tareas, ahorrar el tiempo en el desarrollo de actividades y almacenar la mayor cantidad de información en el menor espacio posible, lo que permitió que las organizaciones se den cuenta de su importancia para la obtención de mejores resultados y competir con sus rivales (TRASOBARES, 2010).

Capítulo 1: Fundamentación teórica del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

Según (Kenneth C. Laudon, 2012) “Podemos plantear la definición técnica de un sistema de información como un conjunto de componentes interrelacionados que recolectan, procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control en una organización”, por lo que se define un sistema de información como el conjunto de recursos humanos y tecnológicos que permitan gestionar la información de una empresa u organización para el apoyo a la toma de decisiones.

1.1.2. Ventajas del uso de los sistemas de gestión de información

Luego de un estudio realizado sobre las ventajas que se obtienen del uso de los sistemas de gestión de información se pueden destacar las siguientes:

- ❖ Disponer de manera rápida y mejor la información, lo que permitirá mejorar la atención a los usuarios (Información en tiempo real).
- ❖ Integración de nuevas tecnologías y herramientas de vanguardia, evitando la pérdida de tiempo en la recopilación de información que se encuentra almacenada.
- ❖ Control efectivo de las actividades de la organización, permitiendo anticipar los requerimientos de la directiva.
- ❖ Posibilidad de generar informes, que permitan corregir fallas, además de comparar resultados alcanzados con los objetivos programados, con fines de evaluación y control.

1.2 Proceso de gestión de ingresos por concepto de 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. Para poder cobrar la misma, el CB tendrá en cuenta si el militante es estudiante o trabajador, siendo el estudiante quién cotizará de manera permanente veinte centavos y el trabajador del CB se registrará según lo definido en el Reglamento de la UJC, en el cual expresa que para determinar la cuota a pagar en cada mes se tendrán en cuenta en el caso de los militantes trabajadores del sector estatal los ingresos mensuales a partir del sistema de pago al que esté acogida la entidad, incluyendo los descuentos o retenciones sobre los salarios por créditos, pensiones, reforma urbana, sanciones judiciales y administrativas u otras similares y los ingresos que obtienen por concepto de vacaciones. En las empresas estatales en que su sistema de pago contempla la distribución de utilidades al final del año, el monto de lo que reciba se incluirá en el cálculo de la cuota a

Capítulo 1: Fundamentación teórica del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

pagar en el mes y en el caso de los trabajadores que tengan más de un empleo, la cuota a abonar estará en correspondencia con la suma de sus salarios y utilidades.

Se excluirán del cálculo de las cuotas los ingresos que provengan de premios y reconocimientos en metálico, por innovaciones y racionalizaciones realizadas en el proceso productivo, los estímulos laborales extraordinarios alcanzados como resultado de la emulación o de otras evaluaciones administrativas, los ingresos que provienen de premios por manifestaciones culturales, el descuento por el aporte de hasta el 5% a la seguridad social y los ingresos por concepto de venta de artículos personales. En todos estos casos se podrán aceptar contribuciones estrictamente voluntarias.

El monto del salario y utilidades se obtendrá mediante la declaración personal de cada uno de ellos, sustentada en el sector estatal a través de las nóminas de pago y en el no estatal a través de los comprobantes de la declaración jurada.

Según dispone el reglamento para poder pagar la cotización del mes actual será requisito indispensable abonar la cuota correspondiente al mes anterior, los militantes abonarán la cotización según la escala de pago que esté vigente aprobada por el Buró Nacional (*ver tabla 1*) y si como resultado de la aplicación de la escala, la cuota es un número decimal se redondeará la misma por defecto.

Tabla 1: Escala de pago de cotización vigente aprobada por el Buró Nacional (Buró Nacional, 2013)

Ingreso total (cup)	Cuota a pagar (cup)
No tienen ingresos	0.20
Hasta 125.00	0.50
126.00 – 199.99	1.00
200.00 – 299.99	3.00
300.00 – 999.99	2% del salario
1000.00 – 3999.99	3% del salario
A partir de 4000.00	120.00

Los militantes propuestos a sanciones externas o desactivación dejarán de pagar desde el momento en que se le propone la sanción o desactivación. En el caso que se decida rectificar la propuesta, este cotizará las cuotas dejadas de abonar, en caso de que se reintegre la condición de militante, pagará su cuota a partir de la fecha de su reincorporación.

Capítulo 1: Fundamentación teórica del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

Cuando un militante recibe ingresos en moneda nacional y en pesos convertibles deberá cotizar por la suma de todos sus ingresos, considerando la divisa como moneda nacional.

A los militantes que salen al exterior por contrato de trabajo o interés personal se les determinará el total de sus ingresos (sumando todas sus cuotas mensuales), tomándose la misma como moneda nacional y cotizando éste por la escala aprobada según corresponda a cada mes. Si la salida es por interés personal estos pueden antes de salir abonar todas las cuotas correspondientes a los meses que va a estar ausente, dejar a una persona encargada de abonarla o a su regreso liquidar todas las cuotas. El monto de la cuota será en correspondencia con los motivos de su salida al exterior. Los que viajen con carácter de estudiantes al exterior cotizarán 20 centavos en monedas en valor internacional, establecidas por el Banco Central de Cuba.

Los efectivos del PCC que mantienen la doble militancia abonarán una cuota de 20 centavos en la UJC y los de la UJC que estén en núcleos mixtos abonarán sus cuotas al partido según la escala establecida por Buró Nacional de la UJC. Los militantes trasladados provisionalmente cotizan en su CB de origen de forma mensual.

Cuando el militante se encuentra subsidiado por enfermedad, accidente, mutilado de guerra u otras causas, el monto se determinará sobre la base del subsidio que percibe durante el tiempo que se mantenga en esa condición, solo se exceptúan los que por concepto de divorcios u otros reciben una pensión para la manutención de sus hijos.

Las compañeras que se encuentran de licencia de maternidad pagarán la cotización sobre la base de los ingresos que reciban durante los meses que le ampara la licencia pre-natal y post-natal, considerándose el dinero recibido en cada momento como un mes normal, a partir de que se venza el ultimo certificado del período post-natal y se acoja a la resolución 11/2001³ referida al pago del salario sobre la base del 60% de lo acumulado en los últimos 12 meses antes de dicha licencia, cotizará según lo recibido cada mes. Al concluir este período, cuando su hijo cumpla el primer año de vida, se acogerá a los mismos principios de los que poseen licencias sin sueldo, en el caso de que no se incorpore al trabajo (UJC, 2014).

1.2.1 Procesos que se realizan en la UJC y se relacionan con la cotización

En la UJC, se realizan determinados procesos, a continuación, se hace una descripción de los que se relacionan con la gestión de ingresos por concepto de cotización:

³ El derecho laboral en Cuba. Fundamentos, actualidad, perspectiva.

Capítulo 1: Fundamentación teórica del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

- ❖ **Altas:** El proceso de alta es la incorporación a la organización de un militante, ya sea por el proceso de **crecimiento** o la incorporación a otra entidad o CB (**traslado**). Por el proceso de crecimiento “pueden ingresar a la UJC los jóvenes cubanos de 16 a 30 años que se hayan destacado por su fidelidad y consagración en el cumplimiento de las tareas de la Revolución y su activa participación en el trabajo, en el estudio y en la defensa de la Patria, que se distinguen por su laboriosidad, responsabilidad, honestidad, honradez, sencillez, modestia, sensibilidad humana, voluntad de vencer y demás valores que caracterizan a un revolucionario”. (UJC, 2014)
- ❖ **Crecimiento de un militante:** El proceso de crecimiento no es la simple suma de nuevos militantes, constituye el resultado de la labor político-ideológica con los jóvenes, principal objetivo de cada Organización de Base de la UJC. La aceptación a un militante se hace exclusivamente de forma individual y se basa en principios como la voluntariedad, selectividad y consulta con las masas. (UJC, 2014) Los militantes de nuevo ingreso pagarán su cotización el mes posterior a la fecha de incorporación al CB y cuando causen alta en el control oficial del municipio.
- ❖ **Bajas:** El proceso de baja a un militante se le realiza por los motivos salida de la organización, **sanciones externas** o por **traslado**. La salida de la UJC puede ser por las vías “Desactivación” o “Desactivación temporal”. Una de las sanciones externas es la separación de las filas y se aplica a un militante cuando se considera que los errores que ha cometido son tan graves que afectan el prestigio del mismo o de la Organización y que no pueden ser erradicados en un tiempo determinado militando en ella. (UJC, 2014) En el caso de los propuestos a separación de las filas o desactivación, si el Buró Municipal u organismo equivalente decide rectificar la propuesta del CB, el militante cotizará las cuotas dejadas de abonar.
- ❖ **Proceso de traslado:** El proceso de traslado a un militante es la acción de cambiar de CB. El militante de la UJC, dando cumplimiento a uno de los deberes consagrados en los Estatutos de la Organización, debe solicitar su traslado y cumplir con todos los trámites que se establecen en el Reglamento. Los mismos deberán cotizar en esa Organización de Base hasta el mes en que se realiza el traslado, cuando se incorpore en el nuevo CB, abonará la cuota correspondiente al mes posterior al reflejado en su boleta de traslado, en el escaque referido a “Último mes cotizado”. Si el traslado es de forma provisional, cotizarán en su CB de origen, de forma mensual.
- ❖ **Sanciones:** La Unión de Jóvenes Comunistas establece en sus estatutos que el militante que infringe los principios y normas de la UJC o de la sociedad deberá ser sancionado y podrá cumplir

Capítulo 1: Fundamentación teórica del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

la sanción fuera o dentro de la Organización. Las sanciones pueden ser internas o externas. Las **sanciones internas** pueden ser la amonestación, la separación del cargo o la limitación temporal de derechos. Las **sanciones externas** pueden ser separación o expulsión de las filas. Para la salida de la Organización de los militantes que por determinadas razones no deben continuar en sus filas y estas no constituye motivo de sanción externa, establece como vías la desactivación y la desactivación temporal. **La desactivación** puede acordarla un CB en una de las reuniones cuando:

- El militante salga del país de forma permanente para residir en el exterior y se considere que no existen problemas de principios.
- El militante lo solicite y se considere objetivamente que no está en condiciones de cumplir con las responsabilidades y tareas de la organización y su actitud no implique la adopción de una sanción interna.
- El CB considere que existe pérdida de la ejemplaridad por causas no atribuidas a él.
- No desee continuar en las filas de la UJC, muestre desinterés por ella o mantenga una actitud sumamente pasiva ante las tareas.
- Esté desvinculado, se compruebe que objetivamente no tiene opciones laborales en el territorio donde radica, pero esta situación puede afectar su prestigio o el de la organización.
- Esté desvinculado y teniendo opciones laborales o de estudio no deseen incorporarse a las mismas.

Los CB propondrán al Buró Municipal de la UJC **la desactivación temporal** de un militante cuando:

- Enferme mentalmente en un grado tal que no pueda responder por sus actos o presente otra enfermedad de carácter grave que dificulte su permanencia en las filas.
- Sea sancionado judicialmente y las causas que motiven esta medida no impliquen errores de principios.

Capítulo 1: Fundamentación teórica del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

- Esté desvinculado y no pueda incorporarse por un período de hasta seis meses porque no existen ofertas de empleo, por problemas de enfermedad u otra situación que no afecte su prestigio o el de la organización. (UJC, 2014)

Los militantes propuestos a **sanción externa** o **desactivación**, dejarán de pagar la cotización desde el momento desde que el CB acuerda proponer la sanción o desactivación al Comité Municipal, Distrital o equivalente. En el caso del joven al que se le reintegre la condición de militante, después de aplicada una sanción externa o desactivación, pagará su cotización a partir de la fecha de su reincorporación.

1.2.3 Reportes que se generan en la UJC para el análisis de la cotización y sus características

Con el objetivo de realizar análisis y valoraciones más precisas en la UJC para el cierre contable de la cotización se generan reportes de forma sistemática, que le permitan a la organización una oportuna toma de decisiones. Los mismos son “Cómputo de cotización”, “Modelo CIC-1 Y MTT” y la “Tabla de cierre de control”. Las oficinas de control de los Comités Municipales son las responsables de su custodia, trabajo y archivo. Dentro de los principios organizativos que poseen estos reportes se encuentran:

- ❖ Tendrán carácter mensual y se estructurarán por zonas de trabajo y sectores de la economía una vez cerrado el estado de la militancia. (Ejemplo: Para confeccionar el cómputo de cotización correspondiente al mes de octubre, debe haberse cerrado antes el cómputo del estado de la militancia del mes de septiembre).
- ❖ Es responsabilidad del Miembro del Buró Político Ideológico del municipio asegurar el completamiento por parte de cada responsable de la información que se solicita.
- ❖ Constituyen documentos de permanente consulta para el Buró Municipal, distrito o centros con derechos municipales y cuadros con el objetivo de realizar las valoraciones correspondientes que contribuyan al proceso de toma de decisiones.
- ❖ Servirán de base para el cierre de la cotización del municipio, distritos y centros con derechos municipales, y los análisis sobre sus resultados, por lo que se garantizará su cierre mensual por zonas de trabajo, así como una hoja resumen al final para resumir el territorio.
- ❖ El trabajo de los mismos inicia con el inicio del mes, por lo que como parte de la preparación para el despacho con los C/B, se debe completar los datos que se exigen y que no dependen del intercambio con la OB.

Capítulo 1: Fundamentación teórica del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

Cumpliendo con esto, los documentos plasmarán informaciones como:

- ❖ Toda la información referente al estado de cotización del militante.
- ❖ Desglose de lo cotizado según la escala de pago vigente.
- ❖ Información de cuotas atrasadas, recuperadas e irrecuperables.
- ❖ Información referente a lo cotizado de cada CB.
- ❖ Total de ingresos mensual, promedio y plan de ingreso por CB.
- ❖ Entre otras.

1.3 Sistemas informáticos vinculados al campo de acción

La UJC es una organización creada en Cuba, defiende el socialismo como única opción de desarrollo, justicia social y democracia, en una Patria independiente, que preserva la dignidad y libertad de los cubanos (Buró Nacional, 2013). Por estas características, no existe a nivel internacional precedentes de sistemas de gestión de información que se adecuen al campo de acción.

En nuestro país, del estudio realizado se concluyó que no existe documentado algún sistema de gestión de ingresos para la UJC, sin embargo, se estudiaron sistemas que gestionan procesos de la UJC, de los cuales se realiza un análisis a continuación:

SICOM-UJC: Mejora la distribución y organización de la información de los militantes en todo el país, además se consolida la organización desde los diferentes niveles. SICOM-UJC cuenta con cinco módulos, los que tienen como objetivo garantizar los procesos de captación y gestión de la información que describe de la estructura jerárquica que presenta la organización de la UJC, así como los militantes a la cual pertenecen, automatizar los procesos políticos a los distintos niveles de dirección, garantizar la seguridad del sistema en su totalidad, la generación de los reportes, donde el usuario podrá acceder a la opción de visualizarlos y de exportarlos en formato .xls (DATEC).

Sistema de Información para la UJC de la UCI. Subsistema para la gestión del ID2 de la UJC de la UCI (SIUJC): Con el objetivo de minimizar el tiempo empleado en: la consulta de los datos y/o la organización de los mismos, la generación del cómputo del ID2, un control sobre los movimientos realizados por los militantes, los secretarios de las organizaciones de base, entre otros reportes de importancia para la UJC se crea SIUJC. Permite consultar la información contenida desde cualquier estación de trabajo conectada a la red de la institución. Para acceder a la aplicación sólo se necesita de un navegador web y la conexión puede ser desde cualquier computadora que esté conectada a la red.

Capítulo 1: Fundamentación teórica del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

El Sistema Integral de Gestión de la UJC UCI (Módulo Militante): En la presente investigación se desarrolla el Módulo Militante, es módulo web que solo puede ser usado por los usuarios con rol de Funcionario de Procesos Políticos, Funcionario de Documentación y Control, funcionario de gestión de usuarios y Miembro del Buró UJC autenticados en el sistema. Por medio de este módulo los usuarios pueden gestionar estructuras organizativas, datos de los miembros de la organización, realizar los procesos de crecimiento a la UJC, crear el expediente a un militante y dar de alta o de bajas a estos.

Conclusiones de los sistemas estudiados:

A partir del estudio de estos sistemas se concluye que no es posible utilizar algunos de ellos ya que no presentan en sus funcionalidades la gestión de ingresos por concepto de cotización o algún flujo del mismo. No obstante, todos responden a un negocio específico según los procesos que realizan, lo que permitió identificar información vital en el negocio. Por otra parte, se pudo identificar aspectos de la metodología de desarrollo de software, lenguaje de modelado, lenguajes de programación y gestión de base de datos para el desarrollo del sistema, dando la oportunidad de reutilizar algunos de los elementos antes mencionados de SICOM-UJC.

1.4 Metodología de desarrollo del software

En el proceso de desarrollo de software en su inicio no solo necesita un personal altamente calificado, sino que se hace necesario seguir ciertas pautas predefinidas en el mismo y llevar un comportamiento sistemático, es decir seguir una metodología. Esta será la encargada de garantizar la calidad y reducir la pérdida de tiempo en un proyecto. El “Sistema de gestión de ingresos por concepto de cotización” fue desarrollado empleando la metodología OpenUP. A continuación, se describe la metodología y el porqué de su uso.

1.4.1 OpenUP

OpenUP es un proceso ágil y unificado, que contiene el conjunto mínimo de prácticas que ayudan a los equipos en el desarrollo de software. Se caracteriza por estar centrado en la arquitectura y guiado por los casos de uso. El ciclo de vida de un proyecto según esta metodología se divide en 4 fases fundamentales: Concepción, Elaboración, Construcción y Transición (OpenUP, 2002).

Por lo descrito anteriormente se decidió utilizar la metodología OpenUP ya que es una metodología ágil apropiada para proyectos pequeños y permite disminuir las probabilidades de fracaso e incrementar las probabilidades de éxito. Se acoge a las necesidades del equipo de desarrollo y posibilita detectar errores tempranos a través de un ciclo iterativo. Evita la elaboración de documentación, diagramas e iteraciones

Capítulo 1: Fundamentación teórica del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

innecesarios ya que es un proceso mínimo, completo y extensible. Tiene un enfoque centrado al cliente y con iteraciones cortas.

1.5 Lenguaje de Modelado

El modelado de sistemas de software es una técnica que ayuda al ingeniero de software a visualizar el sistema a construir. Adquiere mayor importancia cuanto mayor es la complejidad del sistema ya que permite estudiar en detalle sus partes elementales. De ahí que los modelos pueden utilizarse para la comunicación con el cliente, grupo de desarrollo y otros factores que intervienen en el proceso de desarrollo. La metodología OpenUP emplea para la especificación, visualización, construcción y documentación de los artefactos que genera durante el proceso de desarrollo del software el Lenguaje Unificado de Modelado (Jisel, y otros, 2015).

1.5.1 UML 2.0

El Lenguaje Unificado de Modelado (UML, del inglés *Unified Modeling Lenguaje*), es un lenguaje estándar para escribir planos de software, se puede utilizar para visualizar, especificar, construir y documentar los artefactos de un sistema que involucre software. Prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. Ofrece nueve diagramas en los cuales modelar sistemas. Es un lenguaje muy expresivo, que cubre todas las vistas necesarias para desarrollar y luego desplegar tales sistemas (UNAD, 2013).

En la presente investigación se hace uso de este lenguaje ya que permite realizar una correcta verificación y validación del modelo realizado. Otra de las ventajas que conllevó a su uso es que debido a que es un lenguaje visual es fácil de entender y permite que los modelos creados puedan servir como documentación del ciclo de vida del sistema. Se encarga de modelar, construir y documentar muchos de los artefactos que necesita OpenUP como metodología seleccionada, entre ellos se encuentran los diagramas de clases del diseño, de casos de uso del sistema, de componente y despliegue.

1.6 Herramienta CASE

Las herramientas CASE son un conjunto de herramientas y métodos asociados que proporcionan asistencia automatizada en el proceso de desarrollo del software a lo largo de su ciclo de vida. Fueron desarrolladas para automatizar esos procesos y facilitar las tareas de coordinación de los eventos que

Capítulo 1: Fundamentación teórica del Sistema de Gestión de Ingresos por Concepto de Cotización para la UIC.

necesitan ser mejorados en el ciclo de desarrollo de software (Meza, 2011). A continuación, se describe la herramienta CASE seleccionada para el desarrollo de la investigación.

1.6.1 Visual Paradigm para UML 8.0

Visual Paradigm para UML es una herramienta CASE que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un Software (Guión Visual Paradigm for UML, 2013-2014).

Las ventajas que proporciona Visual Paradigm para UML son:

- ❖ Dibujo. Facilita el modelado de UML, ya que proporciona herramientas específicas para ello. Esto también permite la estandarización de la documentación, ya que la misma se ajusta al estándar soportado por la herramienta.
- ❖ Corrección sintáctica. Controla que el modelado con UML sea correcto.
- ❖ Coherencia entre diagramas. Al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas, evitando duplicidades.
- ❖ Integración con otras aplicaciones. Permite integrarse con otras aplicaciones, como herramientas ofimáticas, lo cual aumenta la productividad.
- ❖ Generación de código. Permite generar código de forma automática, reduciendo los tiempos de desarrollo y evitando errores en la codificación del software.
- ❖ Generación de informes. Permite generar diversos informes a partir de la información introducida en la herramienta (Guión Visual Paradigm for UML, 2013-2014).

Considerando estos aspectos, se selecciona Visual Paradigm para UML como herramienta para el modelado de la solución que se propone, siendo esta una herramienta multiplataforma que cuenta, además, con versiones gratuitas.

1.7 Lenguaje de programación

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo de cómputo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo (CCM, 2015).

Capítulo 1: Fundamentación teórica del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

A continuación, se describe el lenguaje de programación PHP en su versión 5.6 que será utilizado para la realización del sistema:

1.7.1 PHP 5.6

PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML⁴. Con PHP se puede procesar información de formularios y generar páginas con contenidos dinámicos. Entre las habilidades de PHP se incluye la creación de imágenes a partir de datos. Es simple para principiantes, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales. Muchas aplicaciones web están construidas usando PHP. (González, 2014).

Se seleccionó para el desarrollo de la solución el lenguaje de programación PHP en su versión 5.6 ya que este está orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos. Además, es libre, por lo que se presenta como una alternativa de fácil acceso para todos y permite aplicar técnicas de programación orientada a objetos.

1.8 Entorno de desarrollo

Un entorno de desarrollo o IDE (del inglés *Integrated Development Environment*), es un programa informático compuesto por un conjunto de herramientas integradas para el desarrollo de software. Puede dedicarse a un solo lenguaje de programación o bien puede utilizarse para varios. Las herramientas se ejecutan a través de una interfaz de usuario y constan de un editor de código, un compilador, un depurador y un intérprete (Antonio, 2013).

1.8.1 NetBeans IDE 8.0

La implementación del sistema se hizo mediante el empleo del NetBeans en su versión 8.0, por ser un IDE que brinda un excelente entorno para programar en PHP. Es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en *Java*, es un producto libre y gratuito sin restricciones de uso. Este entorno de desarrollo ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación (NetBeans, 2016).

⁴ Lenguaje de Marcado para Hipertextos HTML (*HyperText Markup Language*) es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web.

Capítulo 1: Fundamentación teórica del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

Entre las principales ventajas de NetBeans IDE que se tuvieron en cuenta para su uso en la implementación del sistema se encuentran:

- ❖ La facilidad de uso durante todo el ciclo de desarrollo.
- ❖ Soporte para crear interfaces gráficas de forma visual.
- ❖ Es un IDE multiplataforma.
- ❖ Permite el completamiento de código, facilitándole el trabajo a los programadores.
- ❖ Es una herramienta fácil de instalar y consume pocos recursos.

1.9 Gestor de bases de datos

Un Sistema Gestor de Base de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos (Katty Ávila, 2015).

1.9.1 PostgreSQL 9.3

PostgreSQL, es un sistema gestor de base de datos relacional y libre, que agiliza la interacción de cliente, servidor y base de datos, donde PostgreSQL es el que realiza la mayoría del trabajo referente a bases de datos cuando se le hacen peticiones. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema (PostgreSQL, 2012).

A continuación, se muestran algunas de las características más importantes por las que se tuvo en cuenta el uso de este potente gestor de base de datos:

- ❖ Funciona bien con grandes volúmenes de datos.
- ❖ Alta concurrencia de usuarios accediendo al sistema.

Capítulo 1: Fundamentación teórica del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

- ❖ Es un sistema multiplataforma.
- ❖ Incorpora funciones para el manejo de fechas.
- ❖ Incluye herencia entre tablas.
- ❖ Permite la declaración de funciones propias, así como la definición de disparadores.
- ❖ Presenta una amplia funcionabilidad a través de su sistema de activación de disparadores.

1.9.2 PgAdmin III v1.22.0

PgAdmin III es una aplicación de diseño y manejo de bases de datos para PostgreSQL. Es un software libre publicado bajo la licencia PostgreSQL. Fue diseñado para responder a las necesidades de los usuarios, desde la escritura de consultas simples SQL hasta la elaboración de bases de datos complejas. La interfaz gráfica es compatible con todas las características de PostgreSQL y facilita su administración. Incluye un editor de la sintaxis SQL y un editor de código del lado del servidor. No se requieren controladores adicionales para comunicarse con la base de datos del servidor (Mariano Reingart, 2014).

1.10 Marcos de trabajo

Es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o extensiones de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio (Los Tecnicos, 2010).

1.10.1 Symfony 2.8.4

Symfony es un *framework* PHP que permite utilizar la arquitectura MVC (Modelo-Vista-Controlador). Fue escrito desde un origen para ser utilizado sobre la versión 5 de PHP ya que hace ampliamente uso de la orientación a objetos que caracteriza a esta versión. Fue diseñado para optimizar el desarrollo de aplicaciones web, proporcionando herramientas para agilizar aplicaciones complejas y guiando al desarrollador a acostumbrarse al orden y buenas prácticas dentro del proyecto.

Capítulo 1: Fundamentación teórica del Sistema de Gestión de Ingresos por Concepto de Cotización para la UIC.

Symfony cuenta con un gran número de librerías, herramientas y manuales de ayuda para el desarrollo de aplicaciones de forma más rápida para desarrollar una aplicación mucho más rápida que haciéndola de la manera tradicional. Además, cuenta con un amplio soporte para la seguridad del sitio (Ardissone, 2012).

Se utiliza Symfony como *framework* pues implementa el patrón arquitectónico Modelo Vista Controlador, presenta un código fácil de leer que incluye comentarios y que posibilita que el mantenimiento sea una tarea sencilla. Es fácil de instalar y configurar en la mayoría de las plataformas, además sigue las mejores prácticas de los patrones de diseño para la web. Symfony permite estructurar el sistema en módulos independientes y se enfoca en agilizar el proceso de construcción al mismo tiempo que reduce el trabajo del implementador.

1.10.2 ExtJS 3.4

ExtJS es una librería *JavaScript* que permite construir aplicaciones complejas para Internet. Esta librería incluye:

- ❖ Componentes de interfaz de usuario de alto desempeño y personalizables.
- ❖ Modelo de componentes extensibles.
- ❖ Un API fácil de usar.

Una de las grandes ventajas de utilizar *ExtJS* es que permite crear aplicaciones complejas utilizando componentes predefinidos, gracias a esto provee una experiencia consistente sobre cualquier navegador, evitando el problema de validar que el código escrito funcione bien en cada uno (Firefox, IE, Safari, entre otros).

Se decidió utilizar *ExtJS* por lo anteriormente planteado y presentar los siguientes beneficios:

- ❖ Existe un balance entre Cliente – Servidor. La carga de procesamiento se distribuye, permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo.
- ❖ Comunicación asíncrona. En este tipo de aplicación el motor de *render* puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente se dé cuenta.

Capítulo 1: Fundamentación teórica del Sistema de Gestión de Ingresos por Concepto de Cotización para la UIC.

- ❖ Eficiencia de la red. El tráfico de red puede disminuir al permitir que la aplicación elija que información desea transmitir al servidor y viceversa, sin embargo, la aplicación que haga uso de la pre-carga de datos puede que revierta este beneficio por el incremento del tráfico (Rosas, 2008)

1.11 Servidor Web

Un servidor web es un programa que utiliza HTTP (*Protocolo de Transferencia de Hipertexto*) para servir los archivos que forman las páginas web a los usuarios en respuesta a sus peticiones, las cuales son enviados por los clientes HTTP de sus computadoras (Rouse, 2015).

1.11.1 Apache 2.4

Es un servidor web HTTP de código abierto para la creación de páginas y servicios web. Es multiplataforma, gratuito, muy robusto y destaca por su seguridad y rendimiento (Cases, 2014). Algunas de las ventajas que conllevaron a su uso fueron:

- ❖ **Instalación/Configuración:** Software de código abierto.
- ❖ **Coste:** Es completamente gratuito.
- ❖ **Funcional y Soporte:** Alta aceptación en la red y muy popular, esto hace que muchos programadores de todo el mundo contribuyen constantemente con mejoras, que están disponibles para cualquier persona que use el servidor web y que Apache se actualice constantemente.
- ❖ **Multi-plataforma:** Se puede instalar en muchos sistemas operativos, es compatible con Windows, Linux y MacOS.
- ❖ **Rendimiento:** Capacidad de manejar más de un millón de visitas por día.

Conclusiones Parciales

El estudio de los conceptos asociados a la gestión de información permitió identificar la importancia para una empresa u organización de este proceso, así como la utilización de sistemas para ello. Por otra parte, con el estudio el proceso de gestión de ingresos por concepto de cotización, se identificó su relación con otros, los reportes que se generan y la importancia para la toma de decisiones de consultar dicha información en tiempo real. A partir del análisis de los sistemas estudiados se identificaron aspectos de la metodología de desarrollo de software, lenguaje de modelado, lenguaje de programación, gestión de base de datos y herramientas para el desarrollo de la solución.

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por concepto de Cotización para la UJC

En el presente capítulo se aborda todo lo referente al análisis y diseño del sistema para la solución de la investigación, definiéndose los artefactos de la ingeniería de *software* en correspondencia con la metodología empleada. Se definen los requisitos no funcionales y funcionales del sistema, agrupando estos últimos en casos de uso. Se muestran conceptos del problema con el uso del modelo de dominio y se definen los patrones de diseño y de la arquitectura a emplear en la solución.

2.1 Descripción de la solución propuesta

El sistema desarrollado tiene como objetivo fundamental gestionar la información del proceso de ingresos por concepto de cotización para la UJC, permitiendo la consulta de los reportes, resúmenes y datos de los militantes en los diferentes niveles de dirección. Para ello el sistema debe permitir al administrador gestionar diferentes usuarios y al JDC registrar estado de cotización del militante para saber si está atrasado o no en el pago de la cotización, calcular el monto a pagar según el sistema de pago y la escala vigente permitiendo modificar la misma. Gestionar toda la información relacionada con los CB, los militantes que la integran y su proceso de cotización. Generar y exportar a documentos con extensión .xls los reportes y resúmenes mensuales necesarios para el cierre contable de la cotización a partir de la información procesada. Por otra parte, debe ser capaz de alertar de forma automática si un militante está atrasado en el pago de la cotización durante el proceso de traslado y registrar la información generada de la visita económica a cada CB.

2.2 Modelo de dominio

Un modelo del dominio se utiliza con frecuencia como fuente de inspiración para el diseño de los objetos de software. Muestra (a los modeladores) clases conceptuales significativas en un dominio del problema; es el artefacto más importante que se crea durante el análisis orientado a objetos (Larmann, 2013).

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

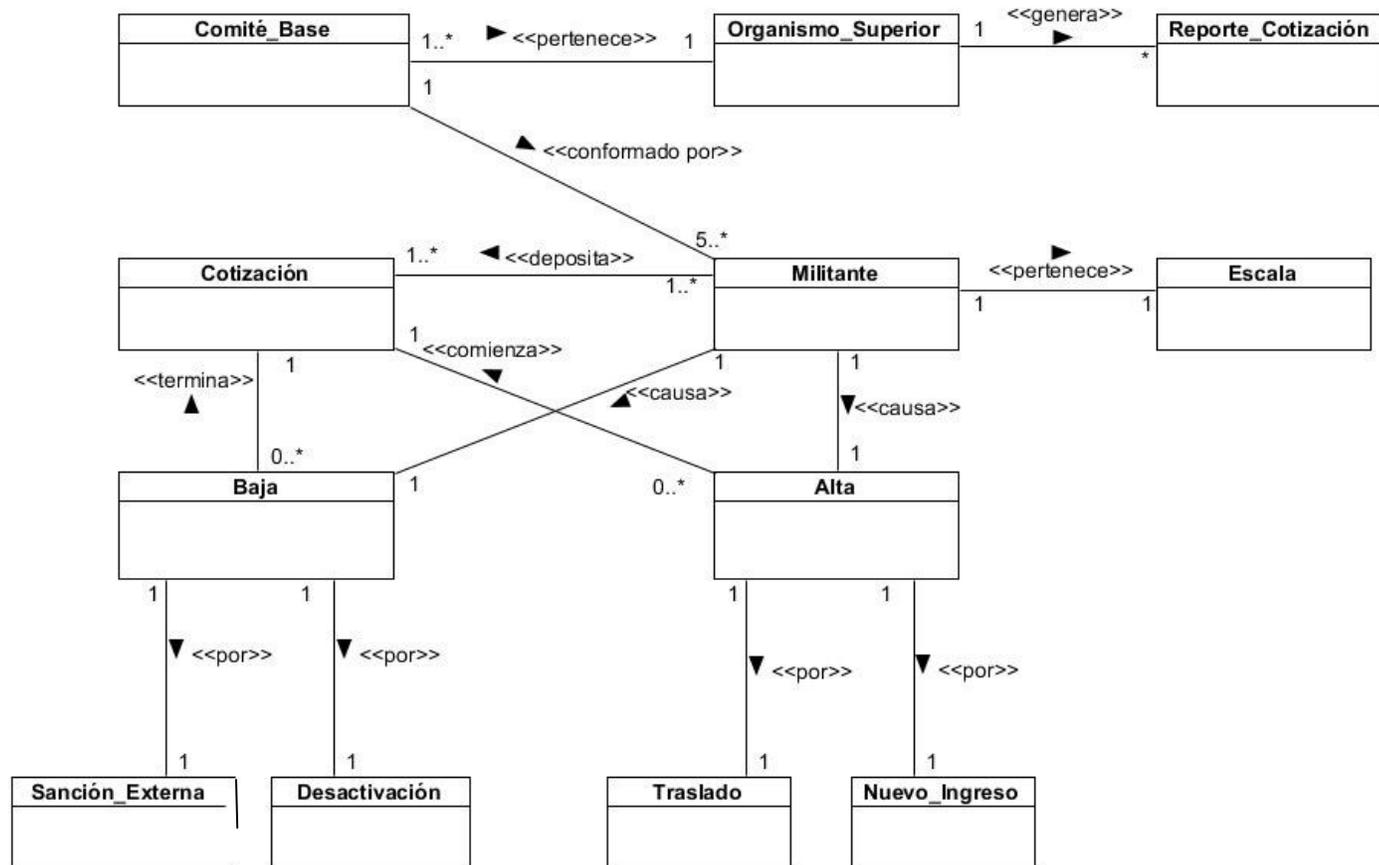


Fig. 1 Modelo de dominio (Fuente: Elaboración propia)

2.2.1 Descripción de las clases del modelo de dominio

A continuación, se muestra una descripción de las clases del modelo de dominio, para ello se realizó un análisis extenso al Reglamento de la UJC:

- ❖ **Organismo_Superior:** Estructura de dirección integrada por militantes de todos los sectores de la vida económica, política y social de los territorios, propuestos o consultados en sus organizaciones de base, teniendo en cuenta sus méritos y trayectoria.
- ❖ **Comité_Base:** Es la célula fundamental de la estructura organizativa de la UJC. Puede ser independiente, provisional o permanente.
- ❖ **Reporte_Cotización:** Texto en que se da cuenta del estado actual o de los resultados del estudio o investigación sobre el proceso de cotización, puede contener datos presentes o pasados ya comprobados. Se obtienen con el objetivo de realizar análisis y valoraciones más precisas del proceso contable en la UJC para el cierre de información y la toma oportuna de decisiones.

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

- ❖ **Militante:** Joven cubano que se encuentra entre los 16 y 30 años que se destaquen por su fidelidad y consagración en el cumplimiento de las tareas de la Revolución. Excepcionalmente y por interés de la propia organización pueden llegar hasta una edad límite de 35 años.
- ❖ **Nuevo_Ingreso:** Es el militante que ingresa a la organización como resultado del proceso de crecimiento.
- ❖ **Alta:** El proceso de alta es la incorporación a la organización de un militante, ya sea por el proceso de crecimiento o la incorporación desde otro CB.
- ❖ **Baja:** El proceso de baja a un militante se le realiza por los motivos salida de la organización, sanciones externas o por traslado.
- ❖ **Traslado:** El proceso de traslado a un militante es la acción de cambiar de CB, para ello se deben realizar los trámites pertinentes.
- ❖ **Sanción_Externa:** La Unión de Jóvenes Comunistas establece en sus estatutos que el militante que infringe los principios y normas de la UJC o de la sociedad deberá ser sancionado y podrá cumplir la sanción fuera o dentro de la Organización. Las **sanciones externas** pueden ser separación o expulsión de las filas.
- ❖ **Escala:** Escala por la que se rige la organización para determinar el monto a cotizar de cada uno de sus militantes. Es aprobada por el Buró Nacional de la UJC.
- ❖ **Cotización:** La cotización es el aporte económico individual de los miembros de la UJC para contribuir a sufragar los gastos de la misma en el cumplimiento de su misión.
- ❖ **Desactivación:** Salida de la organización de los militantes que por determinadas razones no deben continuar en sus filas y estas no constituyen un motivo de sanción externa.

2.3 Especificación de los requisitos de software

En la ingeniería del software, los requisitos se utilizan como datos de entrada en la etapa de diseño del producto y establecen qué debe hacer el sistema, pero no cómo hacerlo. Los requisitos son los servicios proporcionados por el sistema y sus restricciones operativas. (Sommerville, 2005).

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

2.3.1 Requisitos Funcionales

Los requisitos funcionales definen las funciones, capacidades o condiciones que el sistema debe ser capaz de realizar. Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas específicas y de cómo se debe comportar en situaciones particulares. Básicamente establecen los comportamientos del sistema (Sommerville, 2005). La presente investigación cuenta con 62 requisitos funcionales centrados en gestionar usuarios, datos de los militantes y los CB, administrar la información de los mismos, crear, generar y modificar los diferentes reportes que gestionan la información del proceso, modificar la escala de pago, alertar el estado de cotización del militante durante el proceso de traslado y listar los diferentes resúmenes que necesita el proceso.

- | | |
|--|---|
| RF1 Autenticar usuario. | RF22 Exportar listado de militantes que no cotizan por causa. |
| RF2 Insertar usuario. | RF23 Listar militantes según escala de pago por CB. |
| RF3 Editar usuario. | RF24 Exportar listado de militantes según escala de pago por CB. |
| RF4 Listar usuarios. | RF25 Listar militantes con cuotas pendientes por CB. |
| RF5 Eliminar usuario. | RF26 Exportar listado de militantes con cuotas pendientes por CB. |
| RF6 Adicionar datos de un CB. | RF27 Listar militantes analizados por no cotizar por meses. |
| RF7 Editar datos de un CB. | RF28 Exportar listado de militantes analizados por meses. |
| RF8 Listar datos de un CB. | RF29 Listar militantes según el sector ocupacional. |
| RF9 Eliminar datos de un CB. | RF30 Exportar listado de militantes según el sector ocupacional. |
| RF10 Dar alta a un militante. | RF31 Listar total de cuotas a recuperar de militante por CB. |
| RF11 Editar datos de un militante. | RF32 Exportar listado del total de cuotas a recuperar de militante por CB. |
| RF12 Listar datos de los militantes. | RF33 Listar total de cuotas recuperadas de militante por CB. |
| RF13 Dar baja a un militante. | |
| RF14 Registrar cotización mensual del militante. | |
| RF15 Registrar no cotización del militante. | |
| RF16 Registrar visita económica al CB. | |
| RF17 Listar militantes a cotizar por CB. | |
| RF18 Exportar listado de militantes a cotizar por CB. | |
| RF19 Listar militantes que cotizaron por CB. | |
| RF20 Exportar listado de militantes que cotizaron por CB. | |
| RF21 Listar militantes que no cotizan por causa. | |

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

- | | |
|---|--|
| RF34 Exportar listado del total de cuotas recuperadas de militante por CB. | RF48 Exportar listado de plan de ingreso mensual por CB. |
| RF35 Listar cuotas irrecuperables de militantes por CB. | RF49 Obtener Reporte de Cómputo. |
| RF36 Exportar listado de cuotas irrecuperables de militantes por CB. | RF50 Mostrar Reporte de Cómputo. |
| RF37 Listar el porciento de militantes que cotizan por CB. | RF51 Exportar Reporte de Cómputo. |
| RF38 Exportar el listado del porciento de militantes que cotizan por CB. | RF52 Obtener Modelo CIC- Y MTT. |
| RF39 Listar total cotizado por CB. | RF53 Mostrar Modelo CIC- Y MTT. |
| RF40 Exportar el listado del total cotizado por CB. | RF54 Exportar Modelo CIC- Y MTT. |
| RF41 Listar CB por Estado de pago de cotización. | RF55 Obtener Tabla de cierre de control. |
| RF42 Exportar listado del CB por Estado de pago de cotización. | RF56 Mostrar Tabla de cierre de control. |
| RF43 Listar CB con militantes que tienen cuotas atrasadas. | RF57 Exportar Tabla de cierre de control. |
| RF44 Exportar listado del CB con militantes que tienen cuotas atrasadas. | RF58 Crear informe de visita económica. |
| RF45 Listar percápita de ingreso por CB | RF59 Mostrar informe de visita económica por CB. |
| RF46 Exportar percápita de ingreso por CB. | RF60 Exportar informe de visita económica por CB. |
| RF47 Listar plan de ingreso mensual por CB. | RF61 Modificar escala de pago. |
| | RF62 Alertar estado de cotización del militante durante el proceso de traslado. |

2.3.2 Requisitos no funcionales

Los requisitos no funcionales representan características generales y restricciones de la aplicación o sistema que se esté desarrollando. Suelen presentar dificultades en su definición dado que su conformidad o no conformidad podría ser sujeto de libre interpretación, por lo cual es recomendable acompañar su definición con criterios de aceptación que se puedan medir (PMOinformatica, 2015). Los requisitos no funcionales que debe cumplir el sistema para garantizar su correcto funcionamiento son los siguientes:

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

RNF 1. Usabilidad

- ❖ La aplicación debe ser web, pero con características muy similares a las aplicaciones de escritorio en cuanto al diseño de las interfaces visuales.
- ❖ Los botones deben tener texto claro, cada vínculo o botón dentro de la aplicación debe llevar una descripción.
- ❖ Las condiciones de uso del usuario final deben estar ajustadas a un ambiente de oficina, donde la aplicación será instalada en un servidor web con vista nacional para garantizar el acceso a todo el país.

RNF 2. Confiabilidad

- ❖ El sistema debe enviar mensajes de confirmación o error entre el servidor y el cliente, al realizar operaciones de adición, eliminación y edición.

RNF 3. Restricciones del diseño e implementación

- ❖ El sistema deberá ser implementado en el lenguaje de programación PHP versión 5.6. Como marco de trabajo de desarrollo se usará Symfony2 el cual propone una arquitectura modelo, vista, controlador.
- ❖ Para el desarrollo de la vista se utilizará el marco de trabajo ExtJS en su versión 3.4 permitiendo crear interfaces visuales interactivas con apariencia de escritorio, usando tecnología AJAX para la comunicación con el servidor.

RNF 4. Software

PC Cliente:

- ❖ Sistema Operativo: GNU/Linux preferentemente Ubuntu GNU/Linux 14.04, Debian 4 GNU/Linux, Microsoft Windows XP.
- ❖ Navegador Web Mozilla Firefox 45.0.

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UIC.

Servidor de aplicaciones:

- ❖ Sistema Operativo: GNU/Linux preferentemente Ubuntu GNU/Linux 14.04, Debian 4.0 GNU/Linux.
- ❖ Paquetes: Comunes: apache2, php5, libapache2-modphp5, php5-cli, php5-xsl, php5-gd, php5-zip, php5-json, php5-xml.
- ❖ PostgreSQL: php5-pgsql.

Servidor de BD

- ❖ Sistema Operativo: GNU/Linux preferentemente Ubuntu GNU/Linux 14.04, Debian4.
- ❖ PostgreSQL versión 9.3.
- ❖ PgAdmin III o algún administrador para PostgreSQL.

RNF 5. Hardware

PC Cliente:

- ❖ Procesador Intel Pentium 4 a 1.7 GHz o AMD similar.
- ❖ 1GB de RAM.
- ❖ 20 GB de espacio en disco duro.

Servidor de aplicaciones:

- ❖ Procesador Intel Pentium 4 a 1.7 GHz o AMD similar.
- ❖ 1GB de RAM.
- ❖ 40 GB de espacio en disco duro.

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

Servidor de BD

- ❖ Procesador Intel Pentium 4 a 1.7 GHz o AMD similar.
- ❖ 1GB de RAM.
- ❖ 80 GB de espacio en disco duro.

RNF 6. Confidencialidad

- ❖ Los usuarios deben autenticarse para acceder al sistema.
- ❖ Las contraseñas son guardadas en la base de datos cifradas, usando el algoritmo sha512.

RNF 7. Integridad

- ❖ Se deben validar de los datos en el servidor y en el cliente para evitar estados inconsistentes.
- ❖ La información manejada por el sistema estará protegida del acceso no autorizado.
- ❖ Se debe realizar la confirmación sobre acciones irreversibles como eliminaciones.

RNF 8. Disponibilidad

- ❖ Los mecanismos utilizados para lograr la seguridad no obstruyen el acceso a la información, luego de autenticado el usuario tiene pleno acceso a los datos.

2.4 Modelación del sistema

Según (Jacobson, Booch, & Rumbaugh, 2000) el modelo de casos de uso del sistema es un modelo que contiene actores, casos de uso y sus relaciones. Describe lo que el sistema debería hacer por sus usuarios y bajo qué restricciones. Permite que los desarrolladores y clientes lleguen a un acuerdo sobre los requisitos y proporciona la entrada fundamental para el diseño y las pruebas.

2.4.1 Descripción de los actores del sistema

Un actor es una entidad externa al sistema que se modela y que puede interactuar con él. Puede ser una persona o un grupo de personas homogéneas, externos al sistema que se va a desarrollar. Por lo tanto, al identificarlos se está comenzando a delimitar el sistema y a definir su alcance (Jojoa,

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

2015). Para el Sistema de Gestión de Ingresos por concepto de Cotización para la UJC (SGIC) se identificaron como actores el administrador y JDC. A continuación, se muestra una tabla describiendo los objetivos que persiguen cada uno.

Tabla 2. Descripción de los actores del sistema (Fuente: Elaboración propia)

Actor	Objetivo
Administrador	Encargado de gestionar los usuarios del sistema.
Jefe del Departamento de Control (JDC).	Es el encargado de llevar a cabo todos los procesos que se desarrollan en el SGIC.

2.4.2 Casos de Uso

Los casos de uso se crean para refinar un conjunto de requisitos de acuerdo con una función o tarea. En lugar de la tradicional lista de requisitos que quizá no trate de forma directa el uso de la solución, los casos de uso reúnen requisitos comunes basados en el tipo de función u objetivo. Los casos de uso definen qué harán los usuarios o funciones en la solución (IBM, 2013). A continuación, se enumeran los CU identificados según los requisitos funcionales:

CU1 Autenticar usuario.

CU2 Gestionar usuario.

CU3 Gestionar datos del CB.

CU4 Gestionar datos del militante.

CU5 Administrar información del militante.

CU6 Administrar información del CB.

CU7 Exportar información.

CU8 Administrar Reporte de Cómputo.

CU9 Administrar Modelo CIC- Y MTT.

CU10 Administrar Tabla de cierre de control.

CU11 Administrar Informe de visita económica.

CU12 Modificar escala.

CU13 Alertar estado de cotización del militante.

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

2.4.3 Diagrama de Casos de Uso del sistema

Un diagrama de casos de uso es una representación del contexto del sistema, muestra los límites de un sistema, lo que permanece fuera de él y cómo se utiliza. Sirve como herramienta de comunicación que resume el comportamiento de un sistema y de sus actores (Larmann, 2013). Ver Fig. 2 Diagrama de Casos de Uso.

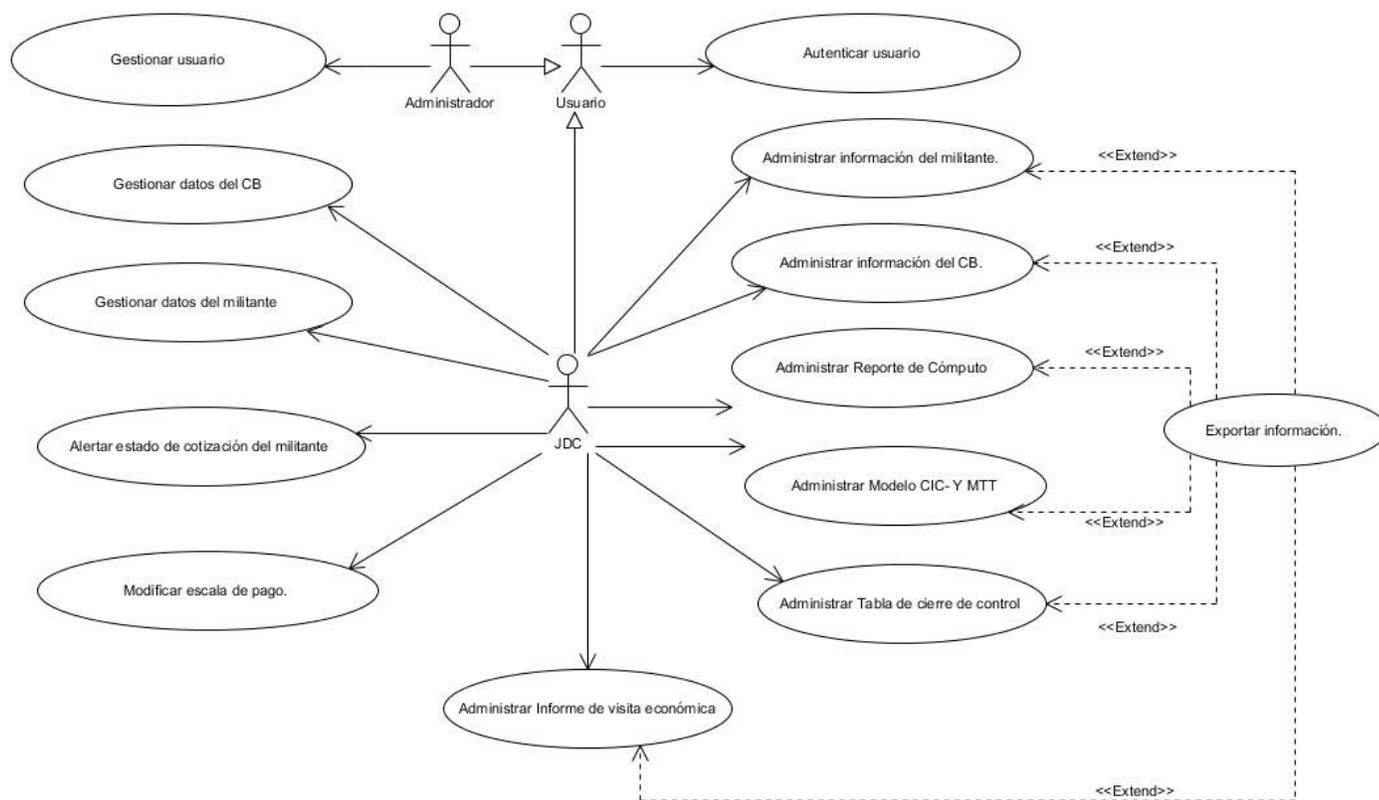


Fig. 2 Diagrama de Casos de Uso (Fuente: Elaboración propia)

2.4.4 Patrones de Casos de Uso utilizados

Los Casos de Uso (CU) se pueden relacionar entre ellos. La organización de los CU mediante relaciones no influye en el comportamiento o los requisitos del sistema. Más bien, es simplemente una forma de organizar para mejorar la comunicación y la comprensión de los CU, reducir la duplicación de texto y mejorar la gestión de los documentos de CU (Larmann, 2013).

Del DCU realizado se identificaron los siguientes patrones:

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

- ❖ **CRUD Total:** Se evidencia en el CU Gestionar datos del CB, el mismo crea, modifica, lista y elimina datos del CB.
- ❖ **Concordancia (Commonality) de adición:** Evidenciado en el CU Exportar información, el mismo extrae una subsecuencia de acciones que aparecen en diferentes lugares del flujo de CU y es expresado por separado (Larmann, 2013). Extiende de los CU Administrar información del militante, Administrar información del CB, Administrar Reporte de Cómputo, Administrar Modelo CIC- Y MTT, Administrar Tabla de cierre de control y Administrar Informe de visita económica.
- ❖ **Múltiples actores roles comunes:** Se observa en el actor usuario del que heredan administrador y JDC realizando el CU Autenticar usuario.

2.4.5 Relación entre los Requisitos Funcionales y los Casos de Uso

Tabla 3. Relación de los CU y los RF (Fuente: Elaboración propia)

Casos de uso	Requisitos funcionales
CU1 Autenticar usuario.	RF1
CU2 Gestionar usuario.	RF2, RF3, RF4, RF5
CU3 Gestionar datos del CB.	RF6, RF7, RF8, RF9
CU4 Gestionar datos del militante.	RF10, RF11, RF12, RF13
CU5 Administrar información del militante.	RF14, RF15, RF17, RF19, RF21, RF23, RF25, RF27, RF29, RF32, RF31, RF33, RF35, RF37
CU6 Administrar información del CB.	RF16, RF39, RF41, RF43, RF45, RF47
CU7 Exportar información.	RF18, RF20, RF22, RF24, RF26, RF28, RF30, RF32, RF34, RF36, RF38, RF40, RF42, RF44, RF46, RF48, RF51, RF54, RF57, RF60
CU8 Administrar Reporte de Cómputo.	RF49, RF50

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

CU9 Administrar Modelo CIC- Y MTT.	RF52, RF53
CU10 Administrar Tabla de cierre de control.	RF55 RF56
CU11 Administrar Informe de visita económica.	RF58, RF59
CU12 Modificar escala.	RF61
CU13 Alertar estado de cotización del militante.	RF62

Descripción de los Casos de Uso del sistema

A continuación, se presenta la descripción del CU Administrar información del militante, en el mismo se registran y listan datos del proceso de cotización.

Tabla 4. Descripción del CU Administrar información del militante (Fuente: Elaboración propia)

Caso de Uso	Administrar información del militante.
Actores	Jefe del departamento de Control (JDC)
Propósito	Este caso de uso se realiza con el objetivo de registrar y listar toda la información relacionada con el militante y su estado de cotización.
Resumen	El caso de uso inicia cuando el usuario desea realizar algunas de las siguientes operaciones: Registrar o Listar información asociada al proceso de cotización del militante.
Complejidad	Media
Precondiciones	El actor tiene que estar autenticado.
Flujo Normal de Eventos 1.1	
Respuesta del Sistema	
1. El caso de uso se inicia cuando el actor: 1.1 Selecciona un CB para registrar datos de los militantes que lo integran. (Ver interfaz 1.1) 1.2 Da clic en la opción Reportes en la Interfaz Principal. (Ver Flujo normal de eventos/Respuesta del sistema 2.2)	2.1.1 El sistema muestra el listado con los militantes pertenecientes al CB seleccionado y sus datos: (Ver Interfaz 1.2) <ul style="list-style-type: none"> • Ci • Nombre

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

	<ul style="list-style-type: none"> • Apellido1 • Apellido2 • CB • Sexo • Fecha alta • Fecha última cotización • Salario • Paga • Cotiza • Causa no cotiza • Sector ocupacional
--	---

<p>3.1 El actor selecciona el militante al cual se le van a registrar los datos.</p>	<p>4.1 El sistema muestra la fila perteneciente a los datos del militante seleccionado resaltada en azul claro, permitiendo escoger una de las siguientes opciones:</p> <ul style="list-style-type: none"> • Adicionar • Eliminar • Registrar cotización • Registrar no cotización • Traslado
---	---

Prototipo de Interfaz

Gestión de Comités de Bases y Militantes

Adicionar	Eliminar	Registrar Visita Económica							
ID	Código	Nombre	Comité Primario	Causa de Activación	Sector o Rama	Fecha Activación	Tipo	Versión	
10100001000001	01	FABICA DE TABAC	COMITÉ PRIMARIO	RAZONES ORGANIZACION	INDUSTRIA	24/04/2014	CBI	3	
10100001000018	08	2DO MEDICINA HOSPITAL PEDRIATICO	COMITÉ PRIMARIO	RAZONES ORGANIZACION	EDUCACIÓN SUPERIOR	28/09/2012	CBI	2	
10100001000020	10	PRODUCCION ARGENTINA	COMITÉ PRIMARIO	RAZONES ORGANIZACION	AZUCARERA CENTRAL	17/07/2008	CBI	2	
10100001000021	11	BANDEC	COMITÉ PRIMARIO	RAZONES ORGANIZACION	AZUCARERA CENTRAL	27/03/2013	CBI	2	

Adicionar	Eliminar	Registrar Cotización	Registrar No Cotización	Traslado								
CI	Nombre	Apellido 1	Apellido 2	C/B	Sexo	Fecha de A...	Última Cotización	Salario	Paga	Cotiza	Causa	Sector Ocupacional
87082322736	YANISLEY	DOMÍNGUEZ	VIERA	BANDEC	FEMENINO	28/05/2008	02/05/2016	\$800.00	\$16.00	SI		EDUCACIÓN
85110418425	MANUEL	RODRIGUEZ	DEL TORO	BANDEC	MASCULINO	13/06/2016	13/06/2016	\$300.00	\$6.00	SI		EDUCACIÓN
91112189673	LEONARD	MÁRQUEZ	ROQUE	BANDEC	MASCULINO	22/06/2016	13/06/2016	\$0.00	\$0.20	SI		MEDICINA
93051376908	THALIA	SEGURA	PÉREZ	BANDEC	FEMENINO	26/06/2007	04/05/2016	\$345.00	\$6.90	SI		EDUCACIÓN

Fig. 1 Interfaz 1.1

Flujo Normal de Eventos Sección “Registrar cotización”

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

Acciones del Actor	Respuesta del Sistema
1. El actor da clic en la opción <i>Registrar cotización</i> .	2. El sistema muestra una ventana al actor con un formulario para introducir los datos de: (Ver interfaz 1.2) <ul style="list-style-type: none"> • Monto • Fecha • Donación
3. El actor introduce los datos que le solicitan, siendo todos los campos obligatorios y da clic en el botón <i>Enviar</i> .	4. El caso de uso termina cuando el sistema guarda todos los datos, actualiza la fecha de última cotización del militante y muestra el mensaje: "Cotización registrada correctamente".
Flujo Alterno sección "Registrar cotización"	
3.1 El actor no introduce alguno de los datos obligatorios y da clic en el botón <i>Enviar</i> .	3.2 El sistema resalta los campos incorrectos en color rojo y muestra un mensaje comunicando: "El formulario no es válido". Se realiza el paso 3 de la Sección "Registrar cotización".
3.1 El actor introduce alguno de los datos que se solicitan con error y da clic en el botón <i>Enviar</i> .	3.2 El sistema resalta los campos incorrectos en color rojo y muestra un mensaje comunicando: "El formulario no es válido". Se realiza el paso 3 de la Sección "Registrar cotización".
3.1 El actor introduce los datos que se solicitan y da clic en el botón <i>Cancelar</i> .	3.2 El sistema cierra la interfaz de "Registrar cotización" sin realizar cambios y muestra la Interfaz Principal.

Prototipo de Interfaz

The image shows a software window titled "Registrar Cotización" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window, there are three input fields arranged in a grid. The first row contains "Monto:" followed by an empty text box. The second row contains "Fecha:" followed by a date picker control. The third row contains "Donación:" followed by a text box containing the number "0". At the bottom of the window, there are two buttons: "Enviar" and "Cancelar".

Fig. 2 Interfaz 1.2

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

Flujo Normal de Eventos Sección “Registrar no cotización”	
Acciones del Actor	Respuesta del Sistema
1. El actor da clic en la opción <i>Registrar no cotización</i>	2. El sistema muestra una ventana al actor con los campos necesarios a introducir para registrar la no cotización, los mismos son: <ul style="list-style-type: none"> • Fecha • Irrecuperable • Análisis • Fecha análisis • Motivo de no cotización
3. El actor introduce los datos que le solicitan, siendo todos los campos obligatorios y da clic en el botón <i>Guardar</i> .	4. El caso de uso termina cuando el sistema guarda todos los datos y muestra el mensaje: <i>“Registro de no cotización guardada correctamente”</i> .
Flujo Alterno sección “Registrar no cotización”	
3.1 El actor no introduce alguno de los datos obligatorios y da clic en el botón <i>Enviar</i> .	3.2 El sistema resalta los campos incorrectos en color rojo y muestra un mensaje comunicando: <i>“El formulario no es válido”</i> . Se realiza el paso 3 de la Sección “Registrar no cotización del militante”.
3.1 El actor introduce alguno de los datos que se solicitan con error y da clic en el botón <i>Enviar</i> .	3.2 El sistema resalta los campos incorrectos en color rojo y muestra un mensaje comunicando: <i>“El formulario no es válido”</i> . Se realiza el paso 3 de la Sección “Registrar no cotización del militante”.
3.1 El actor introduce los datos que se solicitan y da clic en el botón <i>Cancelar</i> .	3.2 El sistema no realiza cambios, cierra la ventana y muestra la Interfaz Principal.
Flujo normal de eventos 1.2	
Acciones del Actor	Respuesta del Sistema
1.2 Da clic en la opción Reportes en la Interfaz Principal.	2.1.2 El sistema muestra los criterios por los que se pueden generar reportes, dando la opción de escoger: <ul style="list-style-type: none"> • <i>Comités de Base</i> • <i>Oficiales</i>
<ul style="list-style-type: none"> • 3. El actor selecciona el término <i>Comités de Base</i>. 	4. El sistema muestra un listado con los diferentes reportes a generar: <i>(Ver Interfaz 1.3)</i> <ul style="list-style-type: none"> • Militantes a cotizar por CB. • Militantes que cotizaron por CB. • Militantes con cuotas pendientes por CB. • Militantes que no cotizan por causa. • Militantes analizados por no cotizar por meses.

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

- Militantes según el sector ocupacional.
- Militantes total cuotas a recuperar por CB.
- Militantes total cuotas recuperadas por CB.
- Militantes total cuotas a atrasadas por CB.
- Militantes cuotas irrecuperables por CB.
- Militantes porciento que cotizan por CB.
- Percápita de ingresos por CB.
- Total de ingresos por CB.
- Estado de pagos de cotización por CB.
- Listar total cotizado por CB.

Prototipo de Interfaz

Listado de Militantes a Cotizar por CB

Exportar a Excel

Nombre	CI	Apellido 1	Apellido 2	Comité de Base
MANUEL	85110418425	RODRÍGUEZ	DEL TORO	BANDEC
YANISLEY	87082322736	DOMÍNGUEZ	VIERA	BANDEC
LEONARD	91112189673	MÁRQUEZ	ROQUE	BANDEC
THALIA	93051376908	SEGURA	PÉREZ	BANDEC
MANUEL	88040223880	BORGES	PINO	INDUSTRIA AZUCARERA
INDIRA	92120134520	SUÁREZ	BARRIOS	INDUSTRIA ARGENTINA

Fig. 3 Listado de militantes a cotizar por CB.

Flujo Normal de Eventos Sección “Militantes a cotizar por CB”

- | | |
|--|--|
| 1. El actor da clic en la opción del menú <i>Militantes a cotizar por CB</i> . | 2. El sistema muestra en una ventana, el <i>listado de militantes a cotizar</i> , dividido por CB. |
| 3. El actor da clic en la opción cerrar ventana. | 4. El caso de uso termina cuando el sistema cierra la ventana y muestra la Interfaz Principal. |

Flujo Normal de Eventos Sección “Militantes que cotizaron por CB.”

- | | |
|--|--|
| 1. El actor da clic en la opción del menú <i>Militantes que cotizaron por CB</i> | 2. El sistema muestra en una ventana, el <i>listado de militantes que cotizaron</i> , dividido por CB. |
| 3. El actor da clic en la opción cerrar ventana. | 4. El caso de uso termina cuando el sistema cierra la ventana y muestra la Interfaz Principal. |

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UIC.

Sección “Militantes con cuotas pendientes por CB.”	
1. El actor da clic en la opción del menú <i>Militantes con cuotas pendientes por CB.</i>	2. El sistema muestra en una ventana, el <i>listado de militantes con cuotas pendientes por CB</i> , dividido por CB.
3. El actor da clic en la opción cerrar ventana.	4. El caso de uso termina cuando el sistema cierra la ventana y muestra la Interfaz Principal.
Flujo Normal de Eventos Sección “Militantes que no cotizan por causa”.	
1. El actor da clic en la opción del menú <i>Militantes que no cotizan por causa.</i>	2. El sistema muestra en una ventana, el listado de militantes que no cotizan por causa, dividido por CB.
3. El actor da clic en la opción cerrar ventana.	4. El caso de uso termina cuando el sistema cierra la ventana y muestra la Interfaz Principal.
Flujo Normal de Eventos Sección “Militantes analizados por no cotizar por meses”.	
1. El actor da clic en la opción del menú <i>Militantes analizados por no cotizar por meses.</i>	2. El sistema muestra en una ventana, el <i>listado de militantes analizados por no cotizar por meses</i> , dividido por CB.
3. El actor da clic en la opción cerrar ventana.	4. El caso de uso termina cuando el sistema cierra la ventana y muestra la Interfaz Principal.
Flujo Normal de Eventos Sección “Militantes según el sector ocupacional”.	
1. El actor da clic en la opción del menú <i>Militantes según el sector ocupacional.</i>	2. El sistema muestra en una ventana, el <i>listado de militantes según el sector ocupacional</i> , dividido por CB.
3. El actor da clic en la opción cerrar ventana.	4. El caso de uso termina cuando el sistema cierra la ventana y muestra la Interfaz Principal.
Flujo Normal de Eventos Sección “Militantes total cuotas a recuperar por CB”.	
1. El actor da clic en la opción del menú <i>Militantes total cuotas a recuperar por CB.</i>	2. El sistema muestra en una ventana, el <i>listado de militantes total cuotas a recuperar por CB</i> , dividido por CB.
3. El actor da clic en la opción cerrar ventana.	4. El caso de uso termina cuando el sistema cierra la ventana y muestra la Interfaz Principal.
Flujo Normal de Eventos Sección “Militantes total cuotas recuperadas por CB”.	
1. El actor da clic en la opción del menú <i>Militantes total cuotas recuperadas por CB.</i>	2. El sistema muestra en una ventana, el <i>listado de militantes total cuotas recuperadas por CB</i> , dividido por CB.

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

3. El actor da clic en la opción cerrar ventana.	4. El caso de uso termina cuando el sistema cierra la ventana y muestra la Interfaz Principal.
Flujo Normal de Eventos Sección “Militantes total cuotas a atrasadas por CB”.	
1. El actor da clic en la opción del menú Militantes total cuotas a atrasadas por CB.	2. El sistema muestra en una ventana, el <i>listado de militantes total cuotas atrasadas por CB</i> , dividido por CB.
3. El actor da clic en la opción cerrar ventana.	4. El caso de uso termina cuando el sistema cierra la ventana y muestra la Interfaz Principal.
Flujo Normal de Eventos Sección “Militantes cuotas irrecuperables por CB”.	
1. El actor da clic en la opción del menú Militantes cuotas irrecuperables por CB.	2. El sistema muestra en una ventana, el <i>listado de militantes cuotas irrecuperables por CB</i> , dividido por CB.
3. El actor da clic en la opción cerrar ventana.	4. El caso de uso termina cuando el sistema cierra la ventana y muestra la Interfaz Principal.
Flujo Normal de Eventos Sección “Militantes porciento que cotizan por CB”.	
1. El actor da clic en la opción del menú Militantes porciento que cotizan por CB.	2. El sistema muestra en una ventana, el <i>listado de militantes porciento que cotizan por CB</i> , dividido por CB.
3. El actor da clic en la opción cerrar ventana.	4. El caso de uso termina cuando el sistema cierra la ventana y muestra la Interfaz Principal.

2.5 Arquitectura de software propuesta

Una arquitectura de software consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información (Alegsa, 2010). En la definición de la arquitectura del sistema se utiliza el patrón arquitectónico que se describe en el siguiente acápite.

2.5.3 Patrón arquitectónico utilizado

Modelo Vista Controlador (MVC)

La selección del patrón arquitectónico MVC se basa en la utilización del marco de trabajo Symfony2 para el desarrollo de la solución. MVC es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UIC.

del código y la separación de conceptos. Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman Modelo, Vista y Controlador.

Modelo: Es la capa donde se trabaja con los datos, contiene los mecanismos para acceder a la información y también para actualizar su estado. Los datos se tienen habitualmente en una base de datos, por lo que en los modelos estarán todas las funciones que accederán a las tablas y harán los correspondientes seleccionar, actualizar, insertar, etcétera (Álvarez, 2014).

Vista: Contienen el código de la aplicación que va a producir la visualización de las interfaces de usuario. En las vistas nada más se tendrán los códigos HTML y JavaScript que permite mostrar la salida. Generalmente se trabaja con los datos, sin embargo, no se realiza un acceso directo a éstos (Álvarez, 2014).

Controlador: Contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, realizar una compra, una búsqueda de información, etcétera. Es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de la aplicación. Sin embargo, su responsabilidad no es manipular directamente datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo (Álvarez, 2014).

La figura 3 muestra un diagrama con la representación de los diferentes componentes de la arquitectura MVC y sus relaciones.

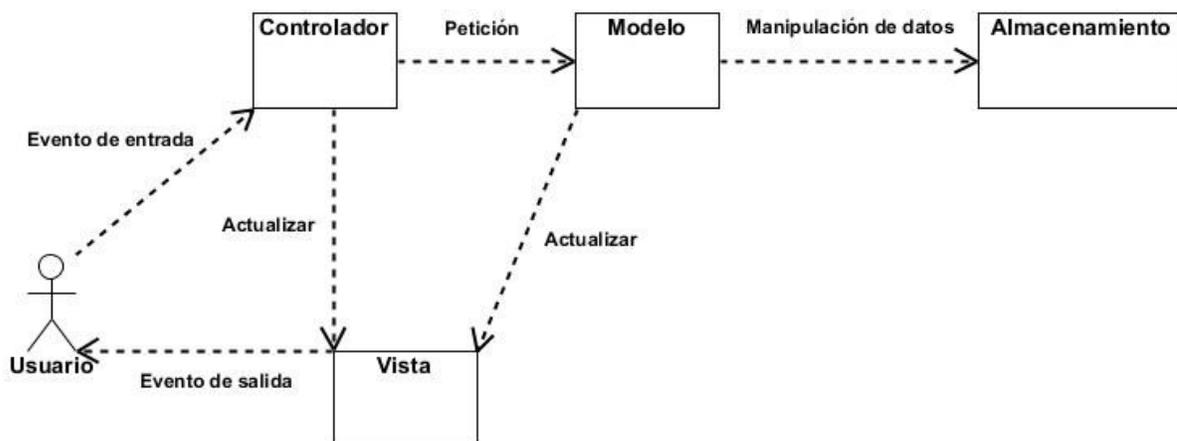


Fig. 3 Representación del Patrón Arquitectónico Modelo Vista Controlador (Álvarez, 2014)

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

2.5.2 Modelo físico de la base de datos

El diseño físico de la base de datos optimiza el rendimiento a la vez que asegura la integridad de los datos al evitar repeticiones innecesarias de datos. Durante el diseño físico, se transforman las entidades en tablas, las instancias en filas y los atributos en columnas (IBM Knowledge Center, 2016). A continuación, se muestra el modelo físico diseñado para el sistema

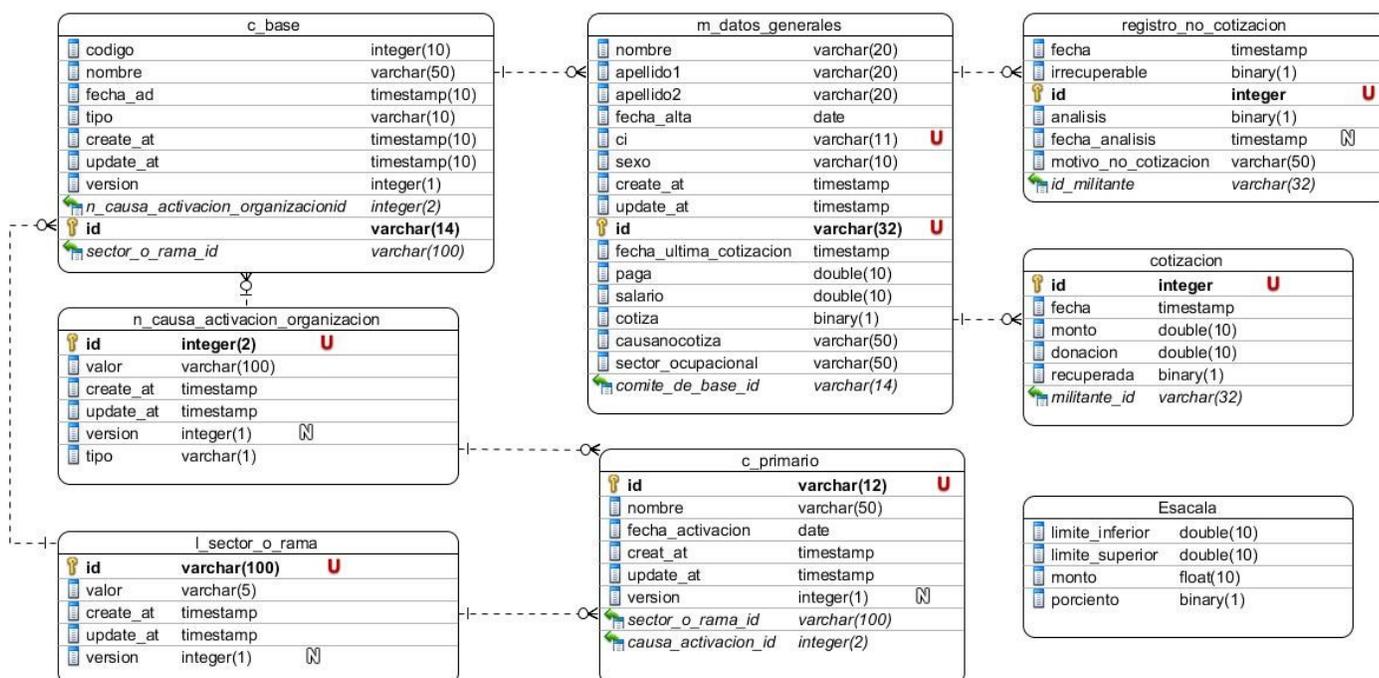


Fig. 4 Modelo físico de la Base de Datos (Fuente: Elaboración propia)

2.5.3 Diagrama de clases del diseño

“Un diagrama de clases de diseño (DCD) representa las especificaciones de las clases e interfaces de software.” (Larmann, 2013) A continuación se muestra el DCD del CU: Administrar información del militante:

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

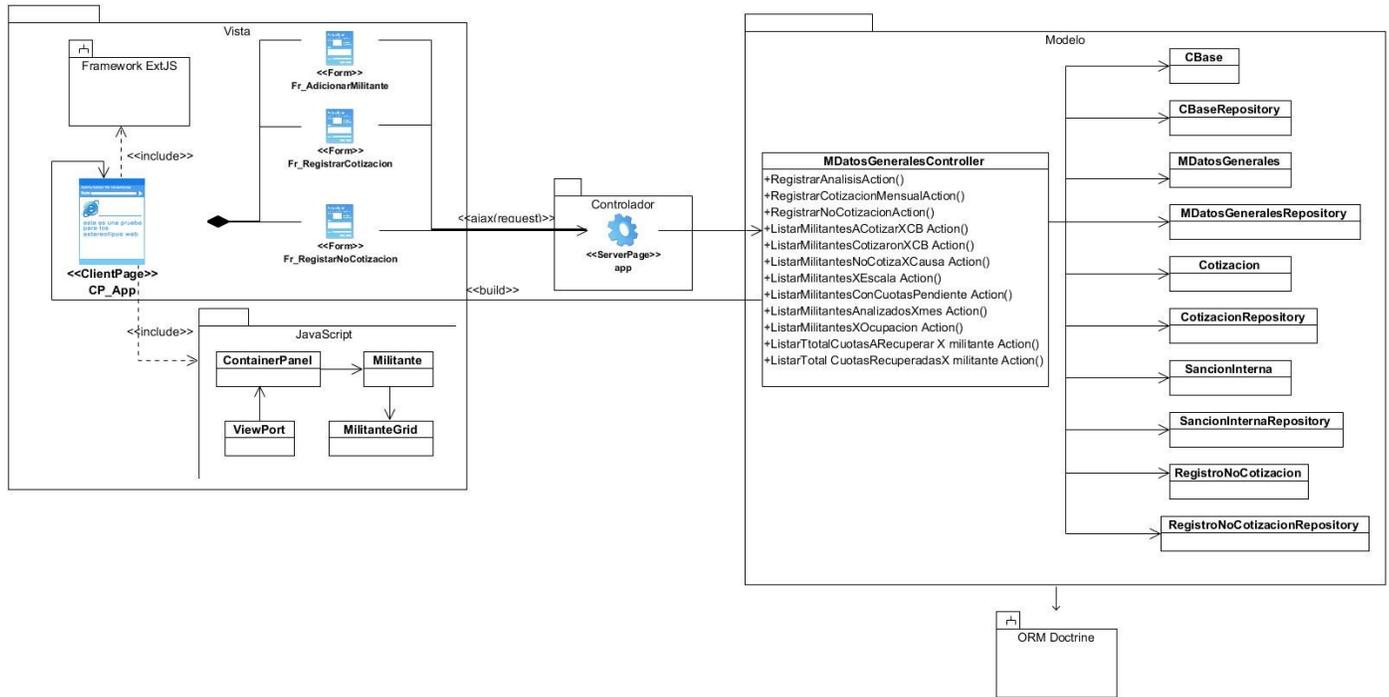


Fig. 5 Diagrama de clases del diseño del CU Administar información del militante (Fuente: Elaboración propia)

2.5.4 Patrones de diseño utilizados en la solución

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces, para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haberse comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias (Prieto, 2009).

Los patrones de diseño se encuentran agrupados en dos grandes grupos fundamentalmente, estos son:

- ❖ Patrones GRASP (*General Responsibility Assignment Software Patterns*).
- ❖ Patrones GOF (*Gang of four*).

2.5.5 Patrones GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. El nombre se eligió para indicar la importancia de captar

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

estos principios, si se quiere diseñar eficazmente el software orientado a objetos (Grosso, 2011). Entre los diferentes tipos de patrones GRASP que se utilizaron para el desarrollo del sistema se encuentran:

Experto: Es el encargado de asignar responsabilidades al experto de la información, es decir, a la clase que tiene la información necesaria para realizar la tarea. Se evidencia en la clase *MDatosGenerales.php*, que contiene todas las funciones que permiten gestionar la información de los militantes.

Creador: Tiene la información necesaria para realizar la creación del objeto. Usa directamente las instancias creadas del objeto, almacena o maneja varias instancias de la clase y contiene o agrega la clase. En la figura 8 se muestra como la clase *UsuarioController.php* es la encargada de crear los objetos que representan a los usuarios con el fin de añadirlos a la base de datos.

```
public function crearUsuarioAction(Request $request) {  
  
    $username = json_decode($request->get('username'));  
    $password = json_decode($request->get('password'));  
    // $salt = json_decode($request->get('salt'));  
    $roll = json_decode($request->get('roll'));  
  
    $em = $this->getDoctrine()->getManager();  
  
    $usuario = new Usuario();  
    $usuario->setUsername($username);  
    $salt = md5(uniqid(null, true));  
    $usuario->setSalt($salt);  
    $user = $this->getUser();  
    $encoder = $this->container->get('security.encoder_factory')->getEncoder($user);  
    $new_pwd_encoded = $encoder->encodePassword($password, $salt);  
    $usuario->setPassword($new_pwd_encoded);  
  
    $usuario->setRoll($roll);  
  
    $em->persist($usuario);  
}
```

Fig. 6 Fragmento de código donde se evidencia la utilización del patrón GRASP: Creador (Fuente: Elaboración propia)

Controlador: Asigna la responsabilidad de gestionar un mensaje de un evento del sistema a una clase. Este se evidencia en la clase *app.php* que es la encargada de capturar la información proveniente de la vista, dicha clase identifica la acción a ejecutar y la envía al controlador correspondiente.

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UIC.

Alta Cohesión: Asigna responsabilidades de manera que la información que almacena una clase sea coherente y esté relacionada con la clase. Para garantizar la alta cohesión se diseñó el sistema de manera tal que por cada entidad se implementó una clase controladora.

Bajo Acoplamiento: Es el encargado de diseñar correctamente las clases, con el objetivo que se encuentren lo menos ligadas entre sí, de tal forma que, en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto. Potencia la reutilización, y disminuye la dependencia entre ellas. Se evidencia cuando el controlador solo se relaciona con las clases entidades necesarias para cumplir su función.

2.5.6 Patrones GOF

Patrones publicados por Gamma, Helm, Johnson y Vlossodes en 1995: patrones de la banda de los cuatro (del inglés, Gang of Four). Esta serie de patrones permiten ampliar el lenguaje, aprender nuevos estilos de diseño y además introducir más notación UML. Existen 23 patrones GoF de los que 15 se utilizan con frecuencia (Ingeniería del Software II, 2011).

Los patrones de diseño del grupo GoF se clasifican en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento. De los mismos se ha tenido en cuenta para la modelación del sistema los siguientes:

- ❖ **Estructurales:** Describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionales pueden ser incluso objetos simples u objetos compuestos. A continuación, se mencionan los patrones que se ponen de manifiesto en la implementación:
 - **Facade (Fachada):** Provee de una interfaz unificada simple para acceder a una interfaz o grupo de interfaces de un subsistema. Se evidencia en la clase *app.php* que es la encargada de manejar los eventos y repartir las responsabilidades.
 - **Adapter (Adaptador):** Adapta una interfaz para que pueda ser utilizada por una clase que de otro modo no podría utilizarla. Se evidencia en la clase *viewPort.js*, encargada de crear las vistas y agregar los componentes visuales.

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

❖ **Comportamiento:** Los patrones de comportamiento ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos. A continuación, un ejemplo del patrón que se utilizó en la implementación:

- **Iterator** (Iterador): *Permite realizar recorridos sobre objetos compuestos independientemente de la implementación de estos (Angelfire, 2011). Se evidencia en la clase CBaseController.php en la cual se recorre el arreglo de objetos de la clase CBase.php con la finalidad de devolver sus datos.*

```
foreach ($comites as $comite) {
    $item = array();
    $item['id'] = $comite->getId();
    $item['montocotizar'] = $comite->getMontoCotizar();
    $item['sectorEconomia'] = $comite->getSectorEconomia();
    $item['noCotizanPorNegligencia'] = $comite->getNoCotizanPorNegligencia();

    if ($comite->getPlanillaCotizacion()) {
        $item['planillaCotizacion'] = 'Si';
    } else {
        $item['planillaCotizacion'] = 'No';
    }

    if ($comite->getPlanillaMilitancia()) {
        $item['planillaMilitancia'] = 'Si';
    } else {
        $item['planillaMilitancia'] = 'No';
    }

    if ($comite->getDespachaCuadro()) {
        $item['despachaCuadro'] = 'Si';
    } else {
        $item['despachaCuadro'] = 'No';
    }

    $item['centroConDerecho'] = $em->getRepository('CentroConDerecho')->findCentroById($comite->getCentroConDerecho());
    $item['municipio'] = $em->getRepository('ComiteMunicipal')->findMunicipioById($comite->getMunicipio());
    $result[] = $item;
}
```

Fig. 7 Fragmento de código donde se evidencia la utilización del patrón GOF: Iterador (Fuente: Elaboración propia)

2.5.7 Diagrama de despliegue

Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Muestra la configuración de los elementos de hardware (nodos) y cómo los elementos y artefactos del software se trazan en esos nodos.

Nodo: es un elemento de hardware o software (Sparx Systems Argentina, 2016).

Capítulo 2: Análisis y diseño del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

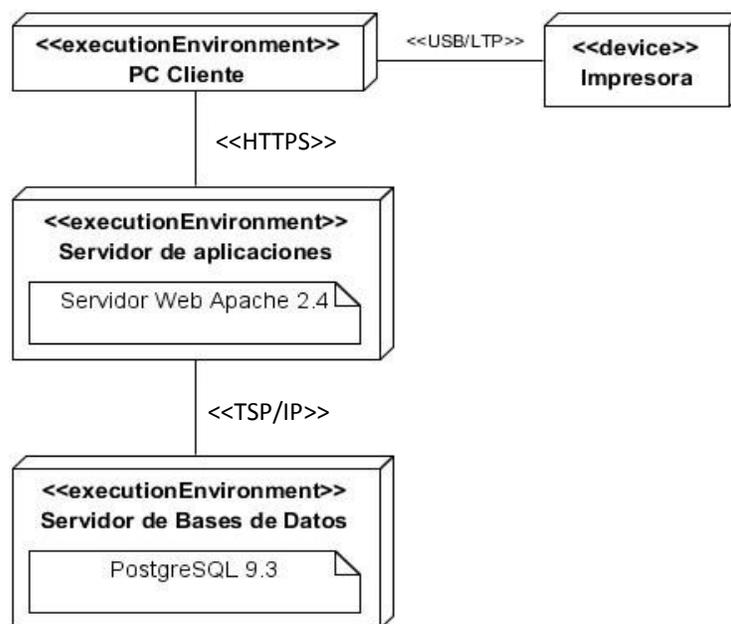


Fig. 8 Diagrama de despliegue (Fuente: Elaboración propia)

Descripción de los nodos:

- ❖ <<executionEnvironment>> PC Cliente: A través de este nodo el usuario podrá acceder a la aplicación mediante el protocolo HTTPS a través del puerto 443.
- ❖ <<executionEnvironment>> Servidor de aplicaciones: La función de este nodo es tener instalada la aplicación a la que el usuario tendrá acceso.
- ❖ <<executionEnvironment>> Servidor de Base de Datos: Permite almacenar los datos del sistema.
- ❖ <<device>> Impresora: Se encarga de permitir al usuario imprimir los reportes y listados que se generan.

Conclusiones Parciales

Con el desarrollo de la fase de análisis y diseño se definieron los artefactos que conforman el expediente de proyecto, en correspondencia con la metodología empleada. La realización del diagrama de clases del dominio permitió un mejor entendimiento del negocio e identificar elementos indispensables para el modelo físico de la Base de Datos. A partir del levantamiento de requisitos se obtuvieron 62 requisitos funcionales, agrupados en 13 casos de uso y 8 categorías de requisitos no funcionales. El diseño de los diagramas de clases brindó la posibilidad de mostrar la estructura de clases del sistema. Se seleccionó el patrón arquitectónico MVC, lográndose un correcto diseño del sistema y con la realización del diagrama de despliegue se mostró la distribución física de la aplicación.

Capítulo 3: Implementación y prueba del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

Capítulo 3. Implementación y prueba del Sistema de Gestión de Ingresos por concepto de Cotización para la UJC

En el presente capítulo se describe el modelo de implementación a partir del diseño realizado por cada caso de uso identificado. Se definen los estándares de codificación a utilizar en el desarrollo del sistema. Se describen las pruebas a realizar, con el fin de comprobar que exista una correcta implementación de cada una de las funcionalidades que se definieron, verificando que los resultados sean los esperados y que el sistema tenga un correcto funcionamiento.

3.1 Modelo de Implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se pueden encontrar datos, archivos, ejecutables, código fuente y directorios. Fundamentalmente se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos (Hernández, 2013).

3.1.1 Diagrama de componentes

Un diagrama de componentes muestra los elementos del diseño de un sistema de software. Permite visualizar la estructura de alto nivel del sistema y el comportamiento del servicio que estos componentes proporcionan y usan a través de interfaces (Visual Studio, 2015).

Capítulo 3: Implementación y prueba del Sistema de Gestión de Ingresos por Concepto de Cotización para la UIC.

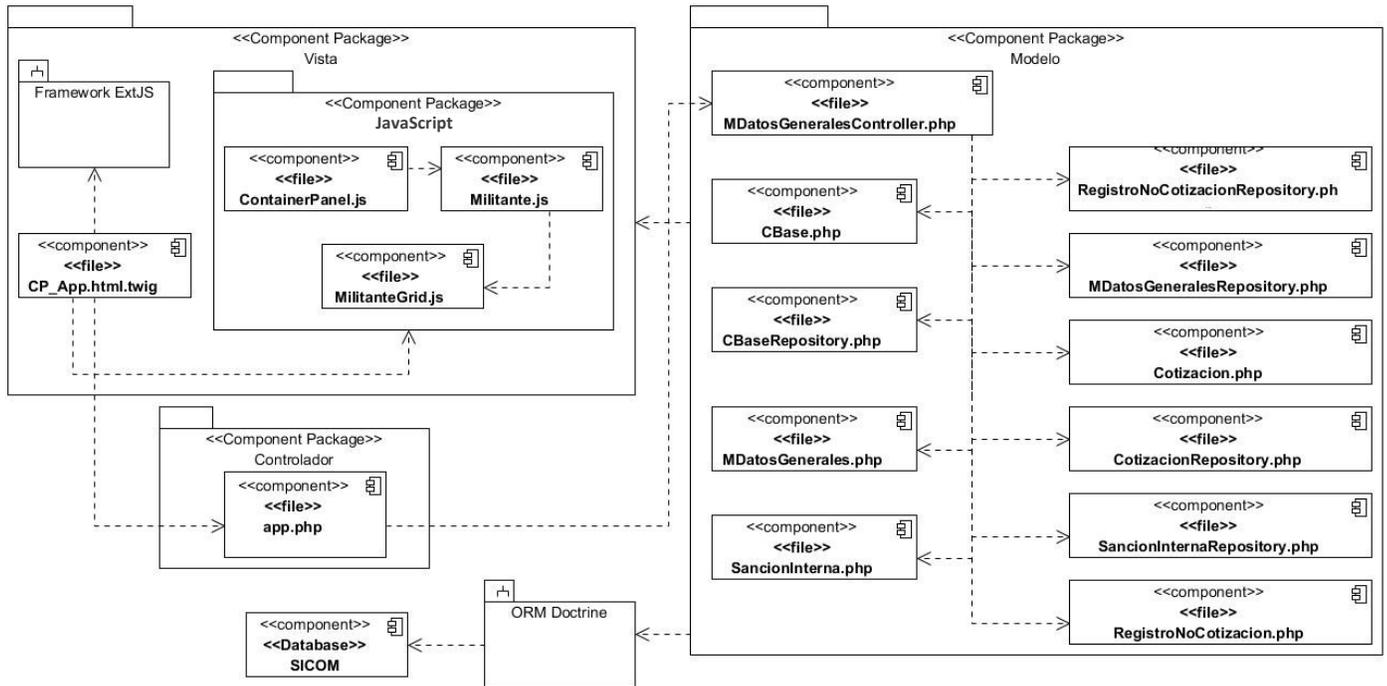


Fig. 9 Diagrama de componentes (Fuente: Elaboración propia)

3.2 Estándares de codificación.

La ISO10 define un estándar como: “Acuerdos documentados que contienen especificaciones técnicas u otros criterios precisos para ser usados consistentemente como reglas, guías o definiciones de características para asegurar que los materiales, productos, procesos y servicios cumplan con su propósito.”

Estilo de codificación utilizado:

- ❖ Los bloques de código siempre deben estar encerrados por llaves.
- ❖ El nombre de métodos y variables, debe permitir que con solo leerlo se conozca el propósito de la misma.
- ❖ Todos los métodos se escriben en estructura lowerCamelCase, este estilo define la primera letra de cada palabra en mayúscula excepto la primera palabra que la pone en minúscula completa.
- ❖ Colocar espacios en blanco entre operadores lógicos-aritméticos y sus operandos.
- ❖ Todas las variables y atributos se escriben en estructura lowerCamelCase.

Capítulo 3: Implementación y prueba del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

- ❖ Todas las funciones y clases estarán comentadas para explicar el flujo del código y el propósito de las funciones o variables.
- ❖ Todas las etiquetas PHP deben ser completas (<?php ?>)... no reducidas (<? ?>).
- ❖ Se usará una indentación para los siguientes casos Tamaño = 4 espacios para:
 - Declaraciones dentro de las clases.
 - Enunciado dentro de métodos y funciones.
 - Enunciados dentro de bloques de comandos.

3.3 Pruebas del software

Las pruebas de software son las investigaciones empíricas y técnicas para proporcionar información objetiva e independiente sobre la calidad del producto. Esta actividad forma parte del proceso de control de calidad global. Las pruebas son básicamente un conjunto de actividades dentro del desarrollo de software y dependiendo del tipo de pruebas, estas actividades podrán ser implementadas en cualquier momento del proceso de desarrollo (Jhonattan Fiestas, 2014).

Al aplicarles las pruebas al software se deben seguir un conjunto de estrategias para lograr que estas se hagan en el menor tiempo posible y con la calidad requerida, además de garantizar que arrojen los resultados esperados. Las estrategias de pruebas que se utilizaron para la implementación del sistema fueron las pruebas unitarias encargadas de verificar el código mediante la técnica de caja blanca y pruebas funcionales mediante la técnica de caja negra, destinadas a evaluar si al finalizar una iteración se lograron las funcionalidades propuestas. También, se realizarán pruebas de aceptación.

3.3.1 Prueba de caja blanca

La prueba de caja blanca del software se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado.

Dentro del método de caja blanca se utilizó la técnica del camino básico. Esta permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa

Capítulo 3: Implementación y prueba del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa (Pressman, 2001).

En el siguiente fragmento de código se muestra la prueba de camino básico que se le aplicó al método *listarCotizanComiteBaseAction*.

```
public function listarCotizanComiteBaseAction() {  
  1 $em = $this->getDoctrine()->getManager();  
    $comites = $em->getRepository('SGIBundle:CBase')->findAll();  
  
    $result = array();  
  
  2 foreach ($comites as $comite) {  
  3     $militantes = $comite->getMDatosGenerales();  
  4     foreach ($militantes as $militante) {  
        if ($militante->getCotiza()) {  
  5         $item = array();  
            $item['cb'] = $comite->getNombre();  
            $item['ci'] = $militante->getCi();  
            $item['nombre'] = $militante->getNombre();  
            $item['apellido1'] = $militante->getApellido1();  
            $item['apellido2'] = $militante->getApellido2();  
            $result[] = $item;  
        }  
    } 6  
  } 7  
  
  8 $finalResult['success'] = 'true';  
    $finalResult['items'] = $result;  
    $json = json_encode($finalResult);  
  
    return new Response($json);  
}
```

Fig. 10 Código del método *listarCotizanComiteBaseAction* (Fuente: Elaboración propia).

Paso 1: Usando el diseño o el código como base, se dibuja el correspondiente grafo de flujo.

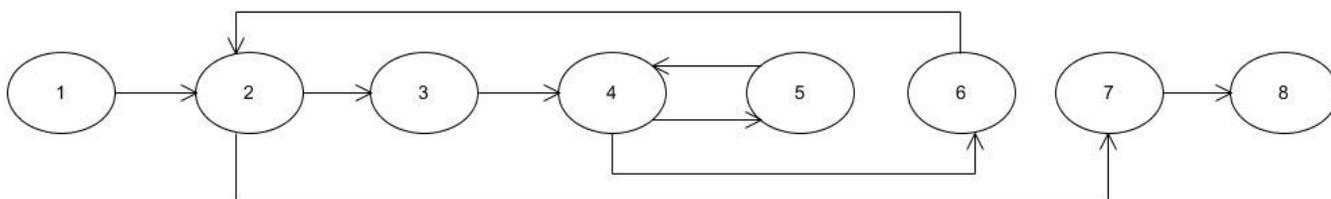


Fig. 11 Grafo de flujo del método *listarCotizanComiteBaseAction* (Fuente: Elaboración propia).

Capítulo 3: Implementación y prueba del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

Paso 2: Se determina la complejidad ciclomática ($V(G)$) del grafo de flujo resultante.

$$V(G)=A-N+2$$

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

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

$$V(G)=9-8+2$$

$$V(G)=2+1$$

$$V(G)=3$$

$$V(G)=3$$

$$V(G)=3$$

Siendo A: Aristas, N: Número de nodos, P: Cantidad de nodos predicado.

Paso 3: Se determina un conjunto básico de caminos linealmente independientes.

1: 1-2-7-8 **2:** 1-2-3-4-5-4-6-2-7-8 **3:** 1-2-3-4-6-2-7-8

Paso 4: Se preparan los casos de prueba que forzarán la ejecución de cada camino del conjunto básico.

Tabla 5. Caso de prueba para el camino básico No.1 (Fuente: Elaboración propia)

Camino	1-2-7-8
Resultado esperado	Listar CB que cotizan.
Resultado de la prueba	Listar CB que cotizan.

Tabla 6. Caso de prueba para el camino básico No.2 (Fuente: Elaboración propia)

Camino	1-2-3-4-5-4-6-2-7-8
Resultado esperado	Listar CB que cotizan.
Resultado de la prueba	Listar CB que cotizan.

Tabla 7. Caso de prueba para el camino básico No.3 (Fuente: Elaboración propia)

Camino	1-2-3-4-6-2-7-8
Resultado esperado	Listar CB que cotizan.
Resultado de la prueba	Listar CB que cotizan.

3.3.2 Prueba de caja negra

Las pruebas de caja negra, también denominada prueba de comportamiento, se centran en los requisitos funcionales del software. Permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa (Pressman, 2001).

La prueba de caja negra intenta encontrar errores de las siguientes categorías:

- ❖ Funciones incorrectas o ausentes.

Capítulo 3: Implementación y prueba del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

- ❖ Errores de interfaz.
- ❖ Errores en estructuras de datos o en accesos a bases de datos externas.
- ❖ Errores de rendimiento.
- ❖ Errores de inicialización y de terminación.

La partición equivalente es un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar (Pressman, 2001).

Por tal razón se decidió utilizar esta técnica en las pruebas del sistema. A continuación, se presenta la Sección (SC) “Registrar cotización”, del caso de prueba realizado al CU “Administrar información del militante”.

Tabla 8. Caso de prueba del CU5 Administrar información del militante (Fuente: Elaboración propia)

Escenario	Descripción	Variables				Respuesta del sistema	Flujo central
		Donación	Monto	Análisis	Fecha_análisis		
EC 1.1 Registrar cotización.	Se registra la cotización correctamente.	V = 0.85	V = 10.00	V="No"	V= 15/02/2015	Muestra el siguiente mensaje luego de registrada la cotización "Cotización registrada correctamente"	1- El sistema guarda todos los datos de la cotización del militante.
EC 1.2 Registrar cotización sin definir algún dato de entrada.	Se insertan los datos y el sistema verifica que no existan campos vacíos.	V = 0.00	I = "	V="No"	V= 21/04/2015	Resalta el (los) campo(s) vacío en rojo y muestra un mensaje comunicando: "El formulario no es válido"	1- Dejar al menos un campo vacío.
EC 1.3 Registrar cotización con alguno de los datos que se solicitan con error.	Se insertan los datos y el sistema verifica que no tengan errores.	I = "cuatro"	V = 12.00	V="No"	V= 05/01/2016	Resalta el (los) campo(s) incorrecto (s) en rojo y muestra un mensaje comunicando: "El formulario no es válido"	1-Introducir datos que no cumpla con el tipo definido

Capítulo 3: Implementación y prueba del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

La tabla que se muestra a continuación describe las variables que se encuentran asociadas al caso de prueba representado, SC “Registrar cotización”.

Tabla 9. Descripción de las variables del caso de prueba (Fuente: Elaboración propia)

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Donación	Campo de texto	No	Cadena de 1 a 6 caracteres de valor numérico ej:100.50
2	Monto	Campo de texto	No	Cadena de 1 a 6 caracteres de valor numérico ej:100.50
3	Análisis	Campo de selección	No	Valor que puede tomar, Si o No.
4	Fecha_análisis	Campo de fecha	No	Se refiere a la fecha en que se le realiza el análisis del militante en formato 'd/m/Y'

3.3.3 Pruebas de aceptación

Las pruebas de aceptación se usan para permitir que el cliente valide todos los requisitos del sistema. Las realiza el usuario final en lugar del responsable del desarrollo del sistema, una prueba de aceptación puede ir desde un informal paso de prueba hasta la ejecución sistemática de una serie de pruebas bien planificadas (Pressman, 2001).

3.4 Resultados de las pruebas

Caja negra

Después de realizar las pruebas funcionales mediante el método de caja negra, aplicando la técnica de partición equivalente, se comprobó el correcto funcionamiento del SGIC para la UJC. Durante las cuatro iteraciones realizadas, se detectaron un total de diecinueve no conformidades. En la primera iteración se encontraron seis de validación y trece de funcionalidad. En la segunda iteración se resolvieron cuatro de validación, seis de funcionalidad, y se detectaron cuatro de ortografía. En la tercera iteración se resolvieron dos de validación, dos de funcionalidad y cuatro de ortografía, quedando para la cuarta iteración solo cinco de funcionalidad por implementar, que limita de exportar alguna información, pero no afecta el cumplimiento del objetivo, como se muestra en la figura 12.

Capítulo 3: Implementación y prueba del Sistema de Gestión de Ingresos por Concepto de Cotización para la UJC.

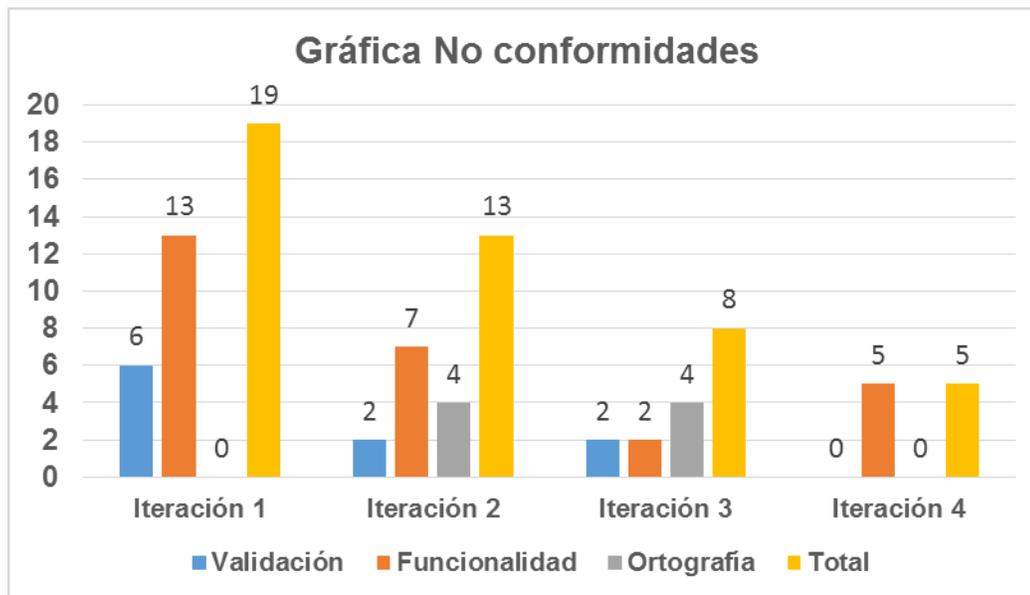


Fig. 12 Resultados de la prueba de caja negra (Fuente: Elaboración propia)

Aceptación

Las pruebas de aceptación se realizaron con el objetivo de validar la solución a través de un encuentro con el JDC, el cual comprobó que el sistema cumple con el funcionamiento esperado y da muestra de su conformidad en el documento oficial Carta de Aceptación del cliente donde quedan plasmados los resultados. Dicha carta se encuentra en el Expediente de Proyecto como parte de los documentos elaborados.

Conclusiones Parciales

La modelación del diagrama de componentes permitió estructurar el funcionamiento de los componentes del sistema. Se emplearon estándares de codificación que brindan mayor legibilidad al código de la aplicación. Para comprobar el resultado de la implementación se realizaron pruebas funcionales sobre la interfaz del sistema, pruebas de caja blanca, mediante la técnica del camino básico que permitieron disminuir considerablemente el número de errores existentes en el software. La prueba de caja negra mediante la técnica de partición de equivalencia permitió analizar los resultados que arroja el sistema para cada uno de los casos, teniendo un total de diez no conformidades en tres iteraciones, quedando nueve resueltas. La prueba de aceptación permitió que el cliente validara los requisitos del sistema.

Conclusiones Generales

Al concluir la presente investigación se obtienen resultados que dan cumplimiento satisfactorio al objetivo general planteado inicialmente. A partir de estos resultados se arriba a las siguientes conclusiones:

- ❖ Se analizaron los procesos de la UJC y la información relacionada a la gestión de ingresos por concepto de cotización y de los procesos que se relacionan con este. Permitiendo elaborar el marco teórico de la investigación, así como al entendimiento del negocio a desarrollar.
- ❖ Con la selección de la metodología, herramientas y tecnologías a utilizar para el desarrollo del SGIC se identificaron y describieron los requisitos del sistema permitiendo con ello modelar la propuesta de solución final.
- ❖ Al definir y describir la arquitectura del sistema se logró un mayor entendimiento del problema garantizando una correcta implementación de la solución.
- ❖ Al realizar una adecuada validación de la solución a través del diseño y ejecución de las pruebas se comprobó el funcionamiento del SGIC, obteniendo como resultado un sistema que permite gestionar los ingresos por concepto de cotización para la UJC y disponer en tiempo real de la información para un mejor desempeño en la toma de decisiones.

Recomendaciones

Con el objetivo de lograr mejoras en el funcionamiento del SGIC, se recomienda:

- ❖ Implementar las funcionalidades de obtener, mostrar y exportar Modelo CIC-1 Y MTT y Tabla de Cierre de Control.
- ❖ Integrar el SGIC a SICOM-UJC.
- ❖ Desarrollar un mecanismo de captura de datos.

Referencias Bibliográficas

Comité Nacional. 2014. *Documento de trabajo sobre las modificaciones a la escala de la cotización que aplica la UJC.* 2014.

Angelfire. 2016. Angelfire.com. [En línea] 2016. <http://www.angelfire.com/empire2/ivansanes/bywbox.htm>.

Antonio. 2013. Entorno de Desarrollo Integrado o IDE. [En línea] 25 de Enero de 2013. [Citado el: 25 de Diciembre de 2015.] <https://antonionll.wordpress.com/2013/01/25/entorno-de-desarrollointegrado-o-ide/>.

Apache JMeter. 2016. Apache JMeter. [En línea] Mayo de 2016. <http://jmeter.apache.org/>.

Arango, Luis Ángel. 2015. Biblioteca Virtual. [En línea] 2015. [Citado el: 10 de 12 de 2015.] <http://www.banrepcultural.org/blaavirtual/ayudadetareas/economia/ingresos>.

Ardissone, Juan. 2012. Introducción a Symfony 2. [En línea] 15 de febrero de 2012. [Citado el: 15 de Enero de 2016.] <http://www.maestrosdelweb.com/curso-symfony2-introduccion-instalacion/>.

B, Alexander Oré. 2009. CalidadySoftware.com. [En línea] 2009. http://www.calidadysotware.com/testing/pruebas_funcionales.php.

CCM. 2015. CCM. [En línea] Diciembre de 2015. [Citado el: 27 de Diciembre de 2015.] <http://es.ccm.net/contents/304-lenguajes-de-programacion>.

Chico, Leidys García. 2013. Cubahora. *Cubahora*. [En línea] 9 de mayo de 2013. <http://www.cubahora.cu/blogs/pensar-digital/tic-en-cuba-empleo-y-aprovechamiento>.

2013. CUBAHORA. [En línea] 9 de 5 de 2013. [Citado el: 27 de 10 de 2015.] <http://www.cubahora.cu/blogs/pensar-digital/tic-en-cuba-empleo-y-aprovechamiento>.

Darisleidi Acosta Carrera, Malvin Urrutia Gallardo. 2013. *Módulo para la gestión de plantillas de cursos de la plataforma de teleformación Moodle 2.3.x.* 2013.

DATEC. *Manual de usuario Módulo de Gestión de Información.*

Manual de usuario Módulo de Procesos Políticos.

Manual de usuario Módulo Seguridad.

Manual de usuario Módulo Visor de Reportes.

Fajardo Díaz, Marlen. Biblioteca de la Universidad de las Ciencias Informáticas. [En línea] [Citado el: 11 de 11 de 2015.] http://repositorio_institucional.uci.cu/jspui/handle/ident/2.

González, Enrique. 2014. Aprender a programar. [En línea] 2014. [Citado el: 27 de Diciembre de 2015.] http://www.aprenderaprogramar.com/index.php?option=com_content&id=492:i-que-es-php-y-i-para-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&Itemid=193.

Google Doc. [En línea] [Citado el: 10 de 12 de 15.] <https://docs.google.com/document/d/11DAAIU61UFrhPyAr7n5ewKYGgdRi4rAQZbm7BjLWss0/edit?hl=es>.

Grosso, Andrés. 2011. Prácticas de Software. [En línea] 21 de Marzo de 2011. [Citado el: 28 de Diciembre de 2015.] [Patrones GRASP _ Prácticas de Software.htm](#).

Guerrero, Juliet Armas. 18/5/2013. *0213_Estado del Arte del Producto a desarrollar*. 18/5/2013.

2013. *0213_Estado del Arte del Producto a desarrollar*. 2013.

Guía Ubuntu. 2012. Guía Ubuntu. [En línea] 2012. [Citado el: 25 de Diciembre de 2015.] http://www.guia-ubuntu.org/index.php?title=PgAdmin_III.

Guión Visual Paradigm for UML. 2013-2014. *Guión Visual Paradigm for UML*. 2013-2014.

Henry, Shawn. 2008. *Simplemente pregunta: Integración de la accesibilidad en el diseño*. 2008.

Hernández, Leovigilda. 2013. [En línea] 01 de Junio de 2013. [Citado el: 23 de Febrero de 2016.] <http://ithleovi.blogspot.com/2013/06/unidad-5-modelo-deimplementacion-el.html>.

Jhonattan Fiestas. 2014. ElevenPaths. [En línea] 03 de Septiembre de 2014. <http://blog.elevenpaths.com/2014/09/qa-pruebas-para-asegurar-la-calidad-del.html>.

Jojoa . 2015. Jojoa - tecnología, marketing y crm. [En línea] 2015. [Citado el: 05 de Enero de 2016.] <https://sites.google.com/site/jojoa/analisis-de-sistemas/definicion-de-actor-que-es-un-actor>.

Katty Ávila. 2015. Computer Audio Video Systems Integrator. [En línea] 2015. [Citado el: 25 de Diciembre de 2015.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.

Larmann, Craig. *UML y patrones(2da edición)*.

Larmann, Craig. 2013. UML y Patrones. 2ª Edición. *UML y Patrones. 2ª Edición.* 2013.

Los Tecnicos. 2010. Técnicas de Ingeniería de Software. [En línea] 01 de Mayo de 2010. [Citado el: 25 de Diciembre de 2015.] <http://klcjw10.blogspot.com/2010/05/marco-de-trabajo.html>.

Meza, Mirna. 2011. Herramientas CASE. [En línea] 02 de Abril de 2011. [Citado el: 28 de Diciembre de 2015.] <http://fds-herramientascase.blogspot.com/>.

Mills, Chris Davis. 2015. Mozilla Developer Network. [En línea] 26 de Noviembre de 2015. [Citado el: 27 de Diciembre de 2015.] <https://developer.mozilla.org/es/docs/Web/JavaScript>.

OpenUP. 2002. OpenUP, Introduction to. *Introduction to OpenUP (Open Unified Process).* [En línea] 2002. [Citado el: 20 de Diciembre de 2015.]

PostgreSQL. 2012. PostgreSQL. [En línea] Noviembre de 2012. [Citado el: 04 de Enero de 2016.] <http://postgresql-dbms.blogspot.com.ar/p/limitaciones-puntos-de-recuperacion.html>.

Prieto, Félix. 2009. Patrones de diseño. [En línea] 2009. http://www.infor.uva.es/~felix/datos/priiii/tr_patrones-2x4.pdf...

Reglamento Unión de Jóvenes Comunistas. 2014. *Reglamento Unión de Jóvenes Comunistas.* 2014.

Rincon, Mafe. 2015. Prezi. [En línea] 2015. <https://prezi.com/>.

Rosas, Juan Eladio Sánchez. 2008. Desarrollo en Web. [En línea] 22 de Octubre de 2008. [Citado el: 22 de Diciembre de 2015.] [http://ExtJS lo bueno, lo malo y lo feo _ Desarrollo en Web.htm](http://ExtJS%20lo%20bueno,%20lo%20malo%20y%20lo%20feo_%20Desarrollo%20en%20Web.htm).

Sommerville, Ian. 2005. Ingeniería del software. Séptima edición. *Ingeniería del software. Séptima edición.* Madrid (España) : s.n., 2005.

UCI. 2014. Universidad de las Ciencias Informáticas. [En línea] 2014. [Citado el: 12 de Octubre de 2015.] <http://www.uci.cu/?q=mision>.

2012. Universidad de las Ciencias Informáticas. [En línea] 2012. [Citado el: 12 de Octubre de 2015.] <http://www.uci.cu/?q=mision>.

UJC. 2013. *Estatutos Unión de Jóvenes Comunistas.* 2013.

2014. *Reglamento Unión de Jóvenes Comunistas.* 2014.

Referencias Bibliográficas

UNAD. 2013. Lenguaje Unificado de Modelado UML. [En línea] 2013. [Citado el: 04 de 01 de 2016.] <http://datateca.unad.edu.co/contenidos/200609/exeuml/index.html>.

Visual Studio. 2015. Microsoft. [En línea] 2015. [Citado el: 19 de Febrero de 2016.] <https://msdn.microsoft.com/es-es/library/dd409390.aspx>.

Bibliografía

Comité Nacional. 2014. *Documento de trabajo sobre las modificaciones a la escala de la cotización que aplica la UJC.* 2014.

Angelfire. 2016. Angelfire.com. [En línea] 2016. <http://www.angelfire.com/empire2/ivansanes/bywbox.htm>.

Antonio. 2013. Entorno de Desarrollo Integrado o IDE. [En línea] 25 de Enero de 2013. [Citado el: 25 de Diciembre de 2015.] <https://antonionll.wordpress.com/2013/01/25/entorno-de-desarrollointegrado-o-ide/>.

Apache JMeter. 2016. Apache JMeter. [En línea] Mayo de 2016. <http://jmeter.apache.org/>.

Arango, Luis Ángel. 2015. Biblioteca Virtual. [En línea] 2015. [Citado el: 10 de 12 de 2015.] <http://www.banrepcultural.org/blaavirtual/ayudadetareas/economia/ingresos>.

Ardissone, Juan. 2012. Introducción a Symfony 2. [En línea] 15 de febrero de 2012. [Citado el: 15 de Enero de 2016.] <http://www.maestrosdelweb.com/curso-symfony2-introduccion-instalacion/>.

B, Alexander Oré. 2009. CalidadySoftware.com. [En línea] 2009. http://www.calidadysotware.com/testing/pruebas_funcionales.php.

CCM. 2015. CCM. [En línea] Diciembre de 2015. [Citado el: 27 de Diciembre de 2015.] <http://es.ccm.net/contents/304-lenguajes-de-programacion>.

Chico, Leidys García. 2013. Cubahora. *Cubahora*. [En línea] 9 de mayo de 2013. <http://www.cubahora.cu/blogs/pensar-digital/tic-en-cuba-empleo-y-aprovechamiento>.

2013. CUBAHORA. [En línea] 9 de 5 de 2013. [Citado el: 27 de 10 de 2015.] <http://www.cubahora.cu/blogs/pensar-digital/tic-en-cuba-empleo-y-aprovechamiento>.

Darisleidi Acosta Carrera, Malvin Urrutia Gallardo. 2013. *Módulo para la gestión de plantillas de cursos de la plataforma de teleformación Moodle 2.3.x.* 2013.

DATEC. *Manual de usuario Módulo de Gestión de Información.*

Manual de usuario Módulo de Procesos Políticos.

Manual de usuario Módulo Seguridad.

Manual de usuario Módulo Visor de Reportes.

Fajardo Díaz, Marlen. Biblioteca de la Universidad de las Ciencias Informáticas. [En línea] [Citado el: 11 de 11 de 2015.] http://repositorio_institucional.uci.cu/jspui/handle/ident/2.

González, Enrique. 2014. Aprender a programar. [En línea] 2014. [Citado el: 27 de Diciembre de 2015.] http://www.aprenderaprogramar.com/index.php?option=com_content&id=492:i-que-es-php-y-i-para-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&Itemid=193.

Google Doc. [En línea] [Citado el: 10 de 12 de 15.] <https://docs.google.com/document/d/11DAAIU61UFrhPyAr7n5ewKYGgdRi4rAQZbm7BjLWss0/edit?hl=es>.

Grosso, Andrés. 2011. Prácticas de Software. [En línea] 21 de Marzo de 2011. [Citado el: 28 de Diciembre de 2015.] [Patrones GRASP _ Prácticas de Software.htm](#).

Guerrero, Juliet Armas. 18/5/2013. *0213_Estado del Arte del Producto a desarrollar*. 18/5/2013.

2013. *0213_Estado del Arte del Producto a desarrollar*. 2013.

Guía Ubuntu. 2012. Guía Ubuntu. [En línea] 2012. [Citado el: 25 de Diciembre de 2015.] http://www.guia-ubuntu.org/index.php?title=PgAdmin_III.

Guión Visual Paradigm for UML. 2013-2014. *Guión Visual Paradigm for UML*. 2013-2014.

Henry, Shawn. 2008. *Simplemente pregunta: Integración de la accesibilidad en el diseño*. 2008.

Hernández, Leovigilda. 2013. [En línea] 01 de Junio de 2013. [Citado el: 23 de Febrero de 2016.] <http://ithleovi.blogspot.com/2013/06/unidad-5-modelo-deimplementacion-el.html>.

Jhonattan Fiestas. 2014. ElevenPaths. [En línea] 03 de Septiembre de 2014. <http://blog.elevenpaths.com/2014/09/qa-pruebas-para-asegurar-la-calidad-del.html>.

Jojoa . 2015. Jojoa - tecnología, marketing y crm. [En línea] 2015. [Citado el: 05 de Enero de 2016.] <https://sites.google.com/site/jojoa/analisis-de-sistemas/definicion-de-actor-que-es-un-actor>.

Katty Ávila. 2015. Computer Audio Video Systems Integrator. [En línea] 2015. [Citado el: 25 de Diciembre de 2015.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.

Larmann, Craig. *UML y patrones(2da edición)*.

- Larmann, Craig. 2013.** UML y Patrones. 2ª Edición. *UML y Patrones. 2ª Edición.* 2013.
- Los Tecnicos. 2010.** Técnicas de Ingeniería de Software. [En línea] 01 de Mayo de 2010. [Citado el: 25 de Diciembre de 2015.] <http://klcjw10.blogspot.com/2010/05/marco-de-trabajo.html>.
- Meza, Mirna. 2011.** Herramientas CASE. [En línea] 02 de Abril de 2011. [Citado el: 28 de Diciembre de 2015.] <http://fds-herramientascase.blogspot.com/>.
- Mills, Chris Davis. 2015.** Mozilla Developer Network. [En línea] 26 de Noviembre de 2015. [Citado el: 27 de Diciembre de 2015.] <https://developer.mozilla.org/es/docs/Web/JavaScript>.
- OpenUP. 2002.** OpenUP, Introduction to. *Introduction to OpenUP (Open Unified Process).* [En línea] 2002. [Citado el: 20 de Diciembre de 2015.]
- PostgreSQL. 2012.** PostgreSQL. [En línea] Noviembre de 2012. [Citado el: 04 de Enero de 2016.] <http://postgresql-dbms.blogspot.com.ar/p/limitaciones-puntos-de-recuperacion.html>.
- Prieto, Félix. 2009.** Patrones de diseño. [En línea] 2009. http://www.infor.uva.es/~felix/datos/priiii/tr_patrones-2x4.pdf...
- Reglamento Unión de Jóvenes Comunistas. 2014.** *Reglamento Unión de Jóvenes Comunistas.* 2014.
- Rincon, Mafe. 2015.** Prezi. [En línea] 2015. <https://prezi.com/>.
- Rosas, Juan Eladio Sánchez. 2008.** Desarrollo en Web. [En línea] 22 de Octubre de 2008. [Citado el: 22 de Diciembre de 2015.] [http://ExtJS lo bueno, lo malo y lo feo _ Desarrollo en Web.htm](http://ExtJS%20lo%20bueno,%20lo%20malo%20y%20lo%20feo_%20Desarrollo%20en%20Web.htm).
- Sommerville, Ian. 2005.** Ingeniería del software. Séptima edición. *Ingeniería del software. Séptima edición.* Madrid (España): s.n., 2005.
- UCI. 2014.** Universidad de las Ciencias Informáticas. [En línea] 2014. [Citado el: 12 de Octubre de 2015.] <http://www.uci.cu/?q=mision>.
- 2012.** Universidad de las Ciencias Informáticas. [En línea] 2012. [Citado el: 12 de Octubre de 2015.] <http://www.uci.cu/?q=mision>.
- UJC. 2013.** *Estatutos Unión de Jóvenes Comunistas.* 2013.
- 2014.** *Reglamento Unión de Jóvenes Comunistas.* 2014.

UNAD. 2013. Lenguaje Unificado de Modelado UML. [En línea] 2013. [Citado el: 04 de 01 de 2016.]
<http://datateca.unad.edu.co/contenidos/200609/exeuml/index.html>.

Visual Studio. 2015. Microsoft. [En línea] 2015. [Citado el: 19 de Febrero de 2016.]
<https://msdn.microsoft.com/es-es/library/dd409390.aspx>.

Anexos

Anexo1: Entrevista a directivos de la UJC Nacional.

Universidad de Ciencias Informáticas, 19 de septiembre del 2015.

Entrevista realizada a directivos de la UJC Nacional:

Objetivo: La siguiente entrevista se realizó con el objetivo de determinar las deficiencias del proceso de gestión de ingresos por concepto de cotización, cómo se realiza, las relaciones del mismo con otros procesos de la UJC y los reportes y resúmenes que se generan del mismo.

1. Determinar las deficiencias del proceso.
 - ❖ ¿Qué es el proceso de gestión de ingresos por concepto de cotización?
 - ❖ ¿Con qué frecuencia se cotiza?
 - ❖ ¿Los militantes que tienen ingresos cotizan de la misma forma que los que no tienen ingresos?
 - ❖ ¿Quién es el encargado de recoger la cotización?
 - ❖ ¿Cómo el encargado de recoger la cotización sabe lo que cotiza cada uno de sus militantes?
 - ❖ ¿Los Comités de Base también cotizan?
 - ❖ ¿En los diferentes niveles de dirección se cotiza de la misma manera?
2. Cómo se realiza el proceso.
 - ❖ Podría explicarnos paso a paso como se realiza el proceso de gestión de ingresos por concepto de cotización, desde que un militante cotiza hasta la información que genera dicha cotización en los organismos superiores.
3. Relación del proceso de gestión de ingresos por concepto de cotización con otros procesos de la UJC.
 - ❖ ¿Qué procesos se realizan en la UJC?
 - ❖ ¿Cuál de estos procesos se relacionan con la cotización?
4. Identificar los reportes y resúmenes que se manejan como resultado del proceso.
 - ❖ ¿Con qué documentos debe un CB despachar la cotización?
 - ❖ ¿Qué reportes o resúmenes se generan al cierre mensual de cotización para los diferentes niveles?