



Universidad de las Ciencias Informáticas

Facultad 1

“Sistema web para la gestión de la información del Departamento de Cuadros del Comité Central del Partido Comunista de Cuba”

Trabajo de Diploma para optar por el título de

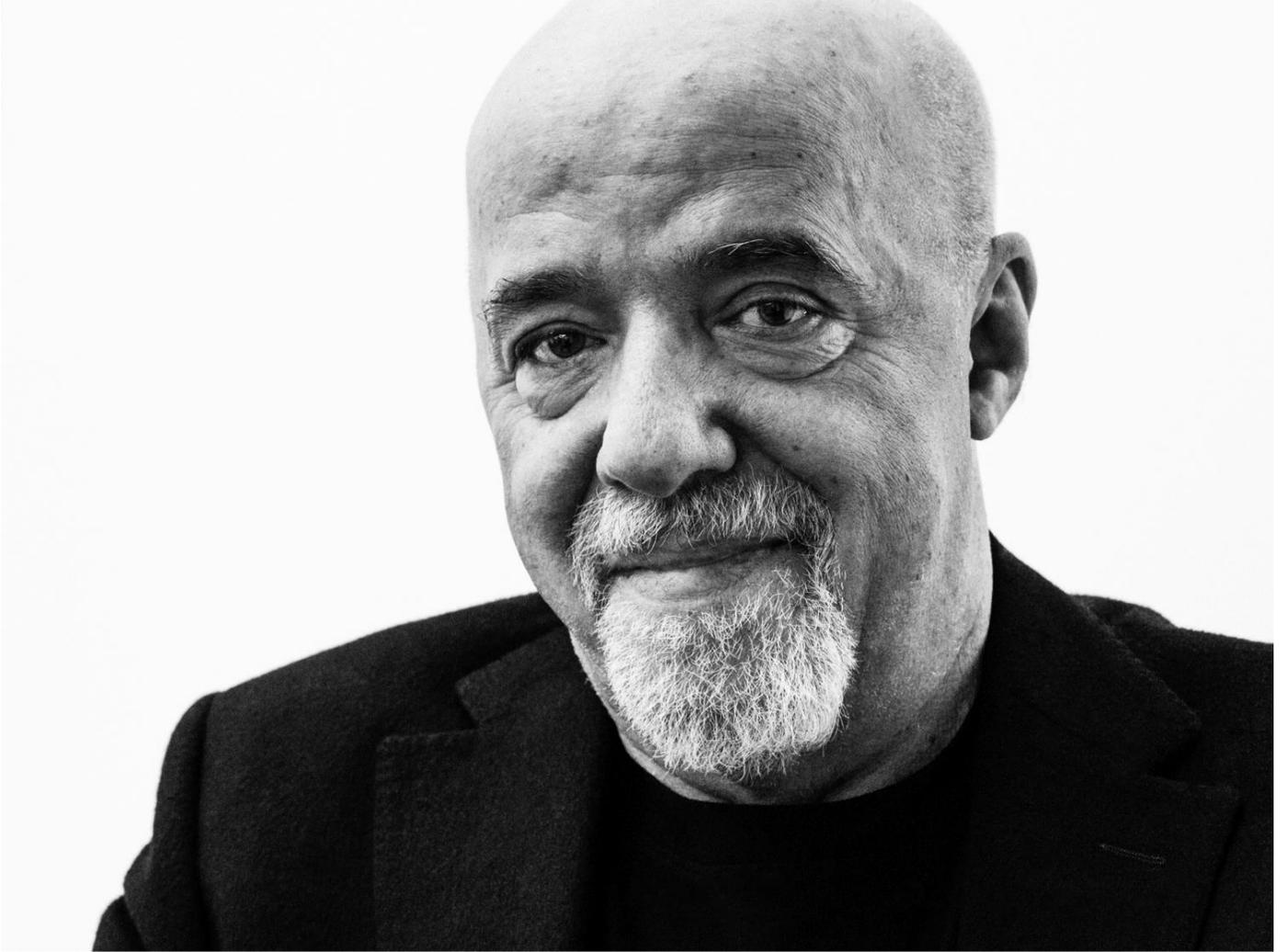
Ingeniero en Ciencias Informáticas

Autor: Rachel Cristina Morales Izquierdo

Tutores: MSc. Alién García Hernández

Ing. Osay González Fuentes

La Habana, 2016



“Hay en el mundo un lenguaje que todos comprenden: es el lenguaje del entusiasmo, de las cosas hechas con amor y con voluntad, en busca de aquello que se desea o en lo que se cree”.

Paulo Coelho

A mi abuelito One, donde quiera que esté.

A mis abuelas Digna y Mima, por tanto cariño.

A mi mamita, eterna compañera de vida, te amo.

A mi mamita, por ser mi amiga, mi hermana, mi guía, mi todo... gracias por tanto amor, entrega y preocupación. Eres el mejor regalo que la vida me pudo dar, el centro de mi universo. Gracias por tu ayuda incondicional, por tus consejos nunca errados, gracias por ser la eterna constructora de mis sueños. Gracias por educarme como lo has hecho, por sembrar en mi sentimientos tan puros y nobles. Gracias por tus abrazos, por las noches en vela, por ser tu puerta siempre la única abierta cuando todos los caminos se cerraban. Esta meta que hoy se cumple, te la debo a ti, como todo lo demás en mi vida.

A mi papito, por ser un ejemplo de lucha y fortaleza, por ayudarme a construir mis sueños y darme los mejores abrazos del mundo. Gracias por todo tu cariño, porque sin saber lo que era me lo diste de la mejor manera y con eso me quedo para toda la vida. Fuiste, eres y siempre serás mi primer amor, el hombre de mi vida. Por todo, gracias.

A mi novio Yoel, compañero de tesis y de vida, gracias por demostrarme que no hay meta imposible de lograr. Gracias por tanto amor y tanta paciencia, por no dejarme desistir nunca, por sostener bien firme mi mano cuando este camino se tornaba tan difícil. Eres lo mejor que me ha pasado en la vida, sabes que te admiro y mucho. Te amo nenito.

A mis tutores, por ayudarme y guiarme durante el desarrollo de la tesis, y contribuir a la materialización de este trabajo.

A mis profesores de la universidad, en especial a Alién, Ilmaris, Joel Arencibia, Niurvis, Delly, Yadier, Yaili, José Ramón y Ángela.

A la profesora Aneyty, por tanta disposición ante cada una de mis dudas, por tanta ternura.

A mi tía Lola, por su inmenso cariño y preocupación, por ser otra madre para mí.

A las personas que se han convertido en mi familia a pesar de no llevar la misma sangre, a Marta, Rey (La Bestia), Miriam y One, por todo... eternamente agradecida.

A mi hermanita de la vida Mayi, gracias por ser mi camaroncito duro, mi confesora y confidente.

A mis amigos y compañeros de estudio Gedric y Angel. El primero bastante loco, pero sin dudas una bella persona, de sentimientos muy nobles, no tengas nunca miedo de mostrar tu verdadera personalidad, porque es genial. El segundo, ha sido todos estos años un ejemplo de perseverancia y sacrificio, gracias

por todo tu apoyo y cariño, por llamarme “chiquitica”. Agradecer también a Olguita, mamá de Angel, pendiente de cada detalle, al tanto de cada nota, por exigirme lo que no me exigía mi mamá.

A todas mis amistades de la UCI que son muchas, en especial a Agnes, Ceci, Evelyn, Yule (a todas gracias por tanto cariño, por ser mi familia en la universidad), a Yasmany, por demostrarme que siempre existen personas maravillosas en el mundo, a Dayron y Yaisel, por tanta alegría compartida y al enano de José Angel.

Agradecer a Rebeca y Reynaldo, mis suegros, por acogerme tan bien en su casa, por hacerme sentir la niña que no tuvieron.

A mi vecina Mercy, por tanto cariño y preocupación durante estos cinco años y a mi amiga Sheyla, su hija.

No por último menos importante quiero agradecer a una personita muy especial y es a Alién García, padre de esta investigación. Mil dos gracias por confiar en mí, por no dudar ni un segundo en ayudarme, gracias por tanto amor y tanta entrega, cuando ni siquiera había empezado a recorrer este camino ya estabas a mi lado, apoyándome y asegurándome que todo iba a salir bien, que tuviera calma. Has sido el mejor profesor que he podido tener en la vida, eres muy grande. No tengo palabras para agradecerte todo lo que has hecho por mí. Gracias por tu ejemplo, te quiero.

Resumen

El objetivo principal de esta investigación es desarrollar un sistema que permita gestionar la información del Departamento de Cuadros del Comité Central del Partido Comunista de Cuba. Surge ante la necesidad de mejorar la gestión de los procesos de solicitud, presentación de la información y evaluación de los cuadros del Partido Comunista de Cuba, procesos que tienen lugar hoy día en el Comité Central del Partido Comunista de Cuba. Actualmente para ejecutar estos procesos, no cuentan con un sistema informático que gestione la información que manejan, esto contribuye a que se vuelva muy engorroso elaborar informes a nivel central y provoca, además, que estos informes no se ajusten en su totalidad a la realidad objetiva, de ahí el objetivo de la presente investigación.

Para la búsqueda y recopilación de información se emplearon los métodos de investigación: análisis documental, analítico sintético, modelación y entrevista. Se llevó a cabo un estudio de diferentes sistemas de gestión de información, enfocándose fundamentalmente en su estructura y funcionamiento para adquirir una mayor visión a la hora de realizar la propuesta de solución. A partir de la realización de este trabajo se obtuvo como resultado un software que resuelve la problemática antes planteada y constituye a la vez un aporte tecnológico al Departamento de Cuadros del Comité Central del Partido Comunista de Cuba.

Palabras clave: aporte tecnológico, gestión de información, información, sistema informático.

Índice

Introducción	1
Capítulo 1: Fundamentación teórica	9
Introducción	9
1.1. Los sistemas de información para la gestión de organizaciones.....	9
1.1.1. Las organizaciones y el uso de las TIC	9
1.1.2. Conceptos asociados a los sistemas de gestión de la información	10
1.1.3. Principales sistemas de gestión de la información para organizaciones	12
1.1.4. Estado actual del proceso de gestión de la información referente a los cuadros en el CCPCC	15
1.2. Metodología de desarrollo de software.....	17
1.3. Herramientas y tecnologías.....	18
1.3.1. Lenguaje de modelado	18
1.3.2. Herramientas CASE.....	19
1.3.3. Lenguaje de programación.....	21
1.3.4. Servidor web.....	22
1.3.5. Entorno de desarrollo.....	22
1.3.6. Sistema Gestor de Bases de Datos	23
1.3.7. Marcos de trabajo	24
Conclusiones del capítulo.....	24
Capítulo 2: Análisis y diseño de la solución	26
Introducción	26
2.1. Modelo de dominio	26
2.2. Descripción de la propuesta de solución.....	27
2.3. Requisitos del sistema	28
2.3.1. Requisitos funcionales (RF)	28
2.3.2. Requisitos no funcionales (RnF)	32
2.4. Historias de Usuario (HU).....	33

2.5.	Planificación.....	36
2.5.1.	Plan de entrega.....	36
2.5.2.	Plan de iteraciones	36
2.6.	Diseño de la propuesta de solución	37
2.6.1.	Patrones de diseño utilizados en la solución.....	37
2.6.2.	Arquitectura de software	39
2.6.3.	Diagrama de clases del diseño	40
2.6.4.	Modelo de datos.....	41
	Conclusiones del capítulo	42
	Capítulo 3: Implementación y prueba de la propuesta de solución	44
	Introducción	44
3.1	Diagrama de componentes	44
3.2	Diagrama de despliegue	46
3.3	Estándares de codificación	46
3.4	Pruebas de software.....	48
3.4.1.	Pruebas de unidad.....	49
3.4.2.	Pruebas de integración.....	51
3.4.3.	Pruebas de validación.....	52
3.4.4.	Pruebas de sistema.....	58
	Conclusiones del capítulo	59
	Conclusiones.....	60
	Recomendaciones	61
	Bibliografía.....	62

Índice de figuras

Figura 1. Sistemas de Gestión de la Información. Elaboración propia	12
Figura 2. Modelo de dominio. Elaboración propia.....	26
Figura 3. Funcionamiento del MTV de Django. (Infante, 2012)	39
Figura 4. Diagrama de clases del diseño. Elaboración propia.....	41
Figura 5. Fragmento del modelo de datos que representa las entidades del módulo “Movimientos e Investigaciones”. Elaboración propia.....	42
Figura 6. Diagrama de componentes. Elaboración propia.....	45
Figura 7. Diagrama de despliegue. Elaboración propia.....	46
Figura 8. Fragmento de código fuente de “views.py”. Elaboración propia.....	47
Figura 9. Fragmento de código fuente de “views.py”. Elaboración propia.....	47
Figura 10. Fragmento de código fuente de “models.py”. Elaboración propia.....	47
Figura 11. Fragmento de código fuente de “models.py”. Elaboración propia.....	48
Figura 12. Resultado de las pruebas de integración. Elaboración propia.....	52
Figura 13. Resultado de las pruebas de validación. Elaboración propia.....	57
Figura 14. Resultado de las pruebas de rendimiento. Elaboración propia.....	59

Índice de tablas

Tabla 1. Requisitos funcionales del Módulo Datos Personales. Elaboración propia.	28
Tabla 2. Requisitos funcionales del Módulo Evaluaciones y Caracterización. Elaboración propia.	30
Tabla 3. Requisitos funcionales del Módulo Nomencladores. Elaboración propia.	31
Tabla 4. Requisitos funcionales del Módulo Movimientos e Investigaciones. Elaboración propia.	32
Tabla 5. Requisitos funcionales del Módulo Reportes. Elaboración propia.	32
Tabla 6. HU Insertar datos personales del cuadro. Elaboración propia.	34
Tabla 7. HU Realizar evaluación del cuadro. Elaboración propia.	35
Tabla 8. Plan de entrega. Elaboración propia.	36
Tabla 9. Plan de iteraciones. Elaboración propia.	37
Tabla 10. Caso de prueba unitaria “RealizarEvaluacionesTest”. Elaboración propia.	49
Tabla 11. Caso de prueba unitaria “ListarReconocimientosTest”. Elaboración propia.	50
Tabla 12. Caso de prueba de integración “MovimientosDetailViewTest”. Elaboración propia.	51
Tabla 13. Caso de prueba Realizar Evaluación. Elaboración propia.	52
Tabla 14. Caso de prueba Modificar Evaluación. Elaboración propia.	53
Tabla 15. Caso de prueba Eliminar Evaluación. Elaboración propia.	54
Tabla 16. Caso de prueba Mostrar evaluaciones del cuadro. Elaboración propia.	55
Tabla 17. Descripción de las variables de entrada para los casos de prueba Realizar y Modificar evaluación. Elaboración propia.	55

Introducción

Desde los inicios de la humanidad el ser humano ha decidido agruparse a fin de satisfacer sus necesidades básicas. Inicialmente lo hacía en hordas, “agrupaciones que nacieron en el paleolítico por la necesidad de protegerse unos a otros” (Diamon, 2006, p.558). Al intercambiar experiencias lograron inventar sus primeras herramientas y así comenzaron a organizarse en grupos, dedicados a la caza, la pesca y la recolección. De esta forma pasaron del miedo a la confianza, de la rivalidad a la cooperación, y percibieron que el camino para su desarrollo pasaba por el contacto social y la necesidad de organizarse, lo que ponía en juego su creatividad.

Invariablemente ha sido vital para todo grupo de personas, acumular información. Los hombres primitivos lo hacían a través de la pintura rupestre, actividad donde tallaban en las paredes de las cavernas sus más disímiles experiencias y predicciones. Asimismo, todas las sociedades hasta la actualidad, han tenido la imperiosa necesidad de hacer persistir la información, puesto que es un elemento fundamental para el desarrollo.

En la era de la información, uno de los principales problemas es su exceso (Aja, 2002), en este sentido, es de gran ayuda auxiliarse de las nuevas Tecnologías de la Información y las Comunicaciones (TIC) para gestionar los procesos en las organizaciones y procesar los grandes volúmenes de datos que estas manejan. La gestión de información se caracteriza por la reducción de etapas y recursos, además, brinda la oportunidad de disminuir el tiempo de algunos procesos para invertirlo en la realización de otros. También trae consigo la simplificación, debido a que intenta reducir el número de personas y departamentos implicados, lo que posibilita que el personal encargado de un proceso pueda diseminarse y encargarse de otras tareas o simplemente se logra aligerar laboralmente a las personas que tengan mucha carga de trabajo (Díaz y González, 2009).

En busca de facilitar el procesamiento de la información en las organizaciones, surgen, haciendo uso de las TIC, los sistemas de gestión de la información (SGI). En este sentido, Arjonilla (2002) afirma que:

Un sistema de información está formado por un conjunto de elementos integrados e interrelacionados que persiguen el objetivo de capturar, depurar, almacenar, recuperar, actualizar y

tratar datos para proporcionar, distribuir y transmitir información en el lugar y momento en el que sea requerido en la organización. (p.219)

Muchas empresas utilizan de alguna forma un SGI. Las empresas más pequeñas se auxilian de las herramientas que proveen los productos de oficina, dígase, por ejemplo, el Excel. Otras, compran un SGI externo y lo integran con el software que utilizan. El tipo de SGI que las empresas escogen, depende de cuánto valor pueda traer este para la compañía (Hamlett y Media, 2015).

Los SGI poseen características que los distinguen de otros sistemas. Hamlett y Media (2015) consideran que la principal característica que tienen los SGI es su capacidad para almacenar datos y que, a la vez, estos datos puedan ser de fácil recuperación por los usuarios del sistema. Otra característica muy importante viene dada en forma de informes, pues la mayoría de los SGI proporcionan múltiples plantillas de informes mientras que otros ofrecen la posibilidad de crear informes específicos y guardarlos como una plantilla para que otras personas las utilicen. La capacidad de producir información es un atributo clave en este tipo de sistemas, y ayuda al proceso de toma de decisiones.

Múltiples son los beneficios que se pueden obtener al implementar en una institución un SGI: acceso rápido a la información; generación de informes, que permiten corregir fallas difíciles de detectar y controlar con un sistema manual; evasión de la pérdida de tiempo al recopilar información que ya está almacenada en bases de datos que se pueden compartir; solución del problema de falta de comunicación entre las diferentes instancias, y a nivel directivo la comunicación se hace más efectiva (Bracho, 2011).

La gestión de la información, a través de las TIC, también se abre paso en las organizaciones políticas, tal es el caso de la Organización de Naciones Unidas (ONU). Gorita y Kuyama (2002) afirman en este sentido que:

Para las organizaciones del sistema de las Naciones Unidas ha adquirido creciente importancia el aprovechamiento de la tecnología de la información y las comunicaciones (TIC), cuya evolución es rápida, como base para el cumplimiento eficiente y eficaz de sus mandatos. Se estima que las organizaciones del sistema de las Naciones Unidas han destinado ya una cantidad importante de recursos (...) a lo largo de los últimos diez años a la implantación de sistemas informatizados de

información para la gestión (...) destinados a facilitar un mejoramiento de la gestión en las esferas de los recursos humanos, las finanzas y la administración. (p.V)

Las empresas y organizaciones cubanas no están ajenas al empleo de las TIC en la gestión de la información, esta es una tarea que se atiende con suma importancia, pues constituye uno de los principios básicos del proceso de informatización de la sociedad que se lleva a cabo hoy día.

Como una de las organizaciones de mayor prestigio en Cuba se encuentra el Partido Comunista de Cuba (PCC), sucesor del Primer Partido Comunista de Cuba fundado entre el 16 y 17 de agosto de 1925 por Carlos Baliño y Julio A. Mella. Lo que es hoy el PCC, fue fundado el 3 de octubre de 1965 en el acto solemne en el cual Fidel Castro leyera la carta de despedida del Guerrillero Heroico Ernesto Che Guevara, quien fuera uno de los inspiradores de la creación de un Partido (PCC, 2011).

El PCC, marxista, leninista y martiano, en su condición de Partido único de la nación cubana, tiene como fortaleza y misión principal unir a todos los patriotas y sumarlos a los intereses supremos de construir el Socialismo, preservar las conquistas de la Revolución y continuar luchando por nuestros sueños de justicia para Cuba y la humanidad toda. (PCC, 2012)

Una de las tareas fundamentales en la actividad del Partido (...), es la relativa al desarrollo de una correcta política en la formación, selección, ubicación y promoción de los cuadros de dirección, así como su preparación y educación. De la labor de los cuadros dependen, en gran medida, los éxitos en las diversas esferas del trabajo partidista, estatal, económico y político en general. (Rojas, 1978, p.57)

El Comandante Ernesto “Che” Guevara (Guevara, 1970) definió al cuadro como:

“... es un individuo que ha alcanzado el suficiente desarrollo político como para poder interpretar las grandes directivas emanadas del poder central, hacerlas suyas y transmitir las como orientación a la masa, percibiendo además las manifestaciones que ésta haga de sus deseos y sus motivaciones más íntimas. Es un individuo de disciplina ideológica y administrativa, que conoce y practica el centralismo democrático y sabe valorar las contradicciones existentes en el método para aprovechar al máximo sus múltiples facetas; que sabe practicar en la producción el principio de la discusión

colectiva y decisión y responsabilidad únicas, cuya fidelidad está probada y cuyo valor físico y moral se ha desarrollado al compás de su desarrollo ideológico, de tal manera que está dispuesto siempre a afrontar cualquier debate y a responder hasta con su vida de la buena marcha de la Revolución. Es, además, un individuo con capacidad de análisis propio, lo que le permite tomar las decisiones necesarias y practicar la iniciativa creadora de modo que no choque con la disciplina.

El cuadro, pues, es un creador, es un dirigente de alta estatura, un técnico de buen nivel político que puede, razonando dialécticamente, llevar adelante su sector de producción o desarrollar a la masa desde su puesto político de dirección.”. (p.123)

Dicha organización plantea que cada cuadro de dirección del país debe cumplir con el Código de Ética de los Cuadros del Estado Cubano, del 17 de julio de 1996, el cual tiene como primer precepto: “Ser sincero, no ocultar ni tergiversar jamás la verdad. Luchar contra la mentira, el engaño, la demagogia y el fraude” además de, “fomentar y cumplir la disciplina, el respeto y la lealtad conscientes al Partido, a la Constitución y demás leyes” (PCC, 1996).

El PCC como organización, también se encuentra inmerso en el proceso de informatización de la sociedad cubana, y a raíz de esto ha decidido informatizar los procesos que se llevan a cabo en el área del Departamento de Cuadros. Para ello ha solicitado el servicio de la Universidad de las Ciencias Informáticas (UCI).

Actualmente el Centro de Identificación y Seguridad Digital (CISED) de la UCI, trabaja en un sistema para el control de los militantes del PCC, a solicitud del Departamento de Funcionamiento del Comité Central del Partido Comunista de Cuba (CCPCC), el cual contiene un subsistema que se especializa en el manejo de la información referente a los cuadros del PCC. Dentro de las funciones que tiene el CCPCC está la de llevar el control de todos los cuadros profesionales del PCC y darles un seguimiento, a través de evaluaciones que se desarrollan periódicamente.

Actualmente esta tarea se realiza de forma manual, muchas veces las personas del Departamento de Cuadros se auxilian de herramientas del paquete de Microsoft Office, tales como el Word y el Excel, pues no cuentan con un sistema informático que gestione la información referente a todos sus cuadros. Los datos de los cuadros se encuentran descentralizados, lo que evita que se pueda acceder a ellos en cualquier

momento y desde cualquier parte del país, esto incide notablemente en demoras a la hora de reunir información. En ocasiones la información se pierde por deterioro de documentos, debido a su uso de forma manual.

A la hora de recopilar información y actualizar datos, muchas veces se necesita para ello de viajes interprovinciales, lo que puede tomar varios días y cuando la información que se necesita es urgente, esta demora entorpece el proceso; aunque vale destacar que también se usa el correo electrónico en casos donde el volumen de datos no es muy grande, de ser así, envían y reciben la información de sus cuadros de dirección de manera cifrada. De igual manera esto no es eficiente, pues muchas veces los datos llegan incompletos y no se ajustan a la realidad, la persona que los llena no tiene en cuenta que muchos de ellos son de carácter obligatorio.

Es una realidad que en muchas ocasiones el expediente del cuadro solo se llena cuando este empieza a ejercer como tal, luego, no se actualiza más, y esta tarea debe ser obligatoria y periódica. Existen duplicados de expedientes de un mismo cuadro, lo que trae redundancia en los datos y quita exactitud en el momento de generar reportes estadísticos. Todo lo antes mencionado hace que se vuelva muy engorroso elaborar informes a nivel central y provoca, además, que estos informes no se ajusten en su totalidad a la realidad objetiva; lo que influye negativamente en el proceso de toma de decisiones de la organización.

Por tanto, en términos de preguntas de investigación, el **problema** se formula de la siguiente forma: ¿Cómo gestionar la información del Departamento de Cuadros del Comité Central del Partido Comunista de Cuba?

El **objeto de estudio** lo constituye, por tanto, el proceso de gestión de la información en las organizaciones.

Para brindarle una solución efectiva al problema en cuestión, se plantea como **objetivo general** desarrollar un sistema web que gestione la información del Departamento de Cuadros del Comité Central del Partido Comunista de Cuba.

El **campo de acción** se enmarca en el proceso de gestión de la información en las organizaciones a través de los sistemas web.

Para alcanzar el objetivo general se plantean los siguientes objetivos específicos:

- Realizar un estudio de los principales referentes teóricos que sustentan la implementación de un Sistema web para la gestión de la información en el Departamento de Cuadros del Comité Central del Partido Comunista de Cuba.
- Realizar el análisis y diseño del Sistema web para la gestión de la información en el Departamento de Cuadros del Comité Central del Partido Comunista de Cuba.
- Implementar el Sistema web para la gestión de la información en el Departamento de Cuadros del Comité Central del Partido Comunista de Cuba.
- Validar el sistema web a través de pruebas internas para probar su correcto funcionamiento.

La presente investigación se sustenta en la siguiente **idea a defender**: el desarrollo de una solución informática que permita gestionar la información asociada a los procesos de solicitud, presentación de la información y evaluación de los cuadros, favorecerá la gestión de estos procesos en el Departamento de Cuadros del Comité Central del Partido Comunista de Cuba.

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

- Sistematización de los principales referentes teóricos que permiten la elaboración del marco teórico de la investigación.
- Análisis valorativo de sistemas existentes que gestionan información en las organizaciones.
- Definición de la metodología, herramientas y tecnologías a utilizar para el desarrollo del Sistema web para la gestión de la información en el Departamento de Cuadros del Comité Central del Partido Comunista de Cuba.
- Identificación de los requisitos funcionales y no funcionales del Sistema web para la gestión de la información en el Departamento de Cuadros del Comité Central del Partido Comunista de Cuba.
- Diseño del Sistema web para la gestión de la información en el Departamento de Cuadros del Comité Central del Partido Comunista de Cuba.
- Implementación del Sistema web para la gestión de la información en el Departamento de Cuadros del Comité Central del Partido Comunista de Cuba.
- Validación del sistema web a través de pruebas internas para probar su correcto funcionamiento.

Justificación de la investigación:

Con el desarrollo de esta investigación se proporcionará una herramienta que permitirá gestionar la información referente a los cuadros del Partido Comunista de Cuba. Además, se realizará un aporte tecnológico al Departamento de Cuadros del Comité Central del Partido Comunista de Cuba.

El **resultado que se espera** al concluir la investigación es la obtención de un software que permita gestionar la información asociada a los procesos de solicitud, presentación de la información y evaluación de los cuadros que se llevan a cabo en el Departamento de Cuadros del Comité Central del Partido Comunista de Cuba; dicho software debe centralizar todos los datos.

Métodos para la búsqueda y procesamiento de la información:

- Análisis documental: Para la revisión bibliográfica, la revisión de las fuentes básicas de información, el estudio de documentos clasificados, como planillas de registro de los cuadros, informes de movimientos, certificados de evaluación, entre otros.
- Analítico-sintético: Al descomponer el problema de investigación en elementos por separado, propició el proceso de construcción del marco teórico de la investigación.
- Modelación: Para la lograr una abstracción del objeto de estudio con vista a explicar la realidad de manera simplificada. Específicamente se utilizó el modelo analógico, el cual permite, a través de un diagrama o esquema, reflejar la estructura de relaciones y determinadas propiedades de la realidad del objeto.
- Entrevista: Para dialogar con funcionarios y trabajadores del Comité Central del Partido Comunista de Cuba y recoger la información tanto primaria como secundaria que tributa al desarrollo del Sistema web para la gestión de información del Departamento de Cuadros del Comité Central del Partido Comunista de Cuba. Esta entrevista permitió conocer el estado actual de los procesos que en esta institución se llevan a cabo. (Ver Anexo 1)

El presente documento está estructurado en tres capítulos, los cuales se especifican a continuación:

Capítulo 1: Dedicado a la fundamentación teórica. En este capítulo se abordan los principales presupuestos teóricos asociados a la investigación, se describen detalladamente las herramientas y técnicas empleadas. Además, se adopta una metodología para guiar el proceso de desarrollo del sistema web para mejorar la gestión de la información del Departamento Cuadros del CCPCC.

Capítulo 2: Dedicado al análisis y diseño de la solución. En este capítulo se hace un levantamiento de los requisitos funcionales y no funcionales necesarios para que el sistema funcione de forma correcta. Se modela el esquema de bases de datos y se construyen los artefactos generados por la metodología seleccionada en el capítulo anterior.

Capítulo 3: Dedicado a la implementación y a las pruebas. En este capítulo se muestra el modelo de implementación como resultado del diseño anteriormente desarrollado. Se describen las pruebas a realizar, con el objetivo de probar el correcto funcionamiento de los objetivos del sistema informático para mejorar la gestión de la información del Departamento Cuadros del Comité Central del Partido Comunista de Cuba.

Capítulo 1: Fundamentación teórica

Introducción

En este capítulo se abordan los conceptos y definiciones que están relacionados con el objeto de estudio, en aras de lograr una mejor comprensión de la investigación; también se presentan las tecnologías y herramientas empleadas en el proceso de desarrollo, además de la fundamentación de la metodología que guía el proceso.

1.1. Los sistemas de información para la gestión de organizaciones

Las TIC han transformado la forma de trabajar y gestionar los recursos en las organizaciones, constituyen un elemento clave para hacer que las empresas sean más productivas. Si se hace un buen uso de ellas, se podrá producir más cantidad, más rápido, con mejor calidad y en menos tiempo (ServiciosTIC, 2015).

1.1.1. Las organizaciones y el uso de las TIC

La información es un recurso que es necesario gestionarlo eficazmente pues de ello depende que se puedan tomar decisiones certeras, lo mismo pasa con los recursos humanos y financieros. Las TIC constituyen medios eficaces para el mejoramiento de la gestión en las organizaciones. Al aumentar considerablemente el volumen de información en la actualidad, muchas instituciones se plantean que las TIC van a contribuir a lograr objetivos de gestión, entendiéndose por esto aumentar la eficiencia en los procesos, reducir costos y minimizar esfuerzos, entre otros.

Pierano y Suárez (como se citó en Saavedra y Tapia, 2013) plantean que las vías por las cuales las TIC ayudan a mejorar el desempeño en las empresas son:

1. **Automatización:** Influyen sobre los procesos rutinarios. El aumento más que proporcional en la eficiencia respondería a la relación que surge a partir de la posibilidad de disminuir el trabajo humano directo, al tiempo que se generan registros.

2. **Accesibilidad a la información:** La posibilidad de acceder a información relevante y precisa con un costo bajo y en tiempo real permite tomar decisiones con la ayuda de una gran variedad de datos.
3. **Costos de transacción:** La información se puede transmitir de manera instantánea y a bajo costo, y esto permite reducir los costos de coordinación tanto al interior como al exterior de la empresa.

Dicho todo esto, la autora considera que, sin el empleo de las TIC en las organizaciones, sus procesos serían mucho más lentos y propensos a errores.

1.1.2. Conceptos asociados a los sistemas de gestión de la información

Para el desarrollo de la solución se tuvieron en cuenta conceptos importantes tales como: gestión de la información y sistemas de gestión de la información. No pocos han sido los autores que han dado criterios y/o definiciones al respecto, tales argumentos expuestos se mencionan a continuación.

La gestión de la Información en las organizaciones

Según Vidal y Araña (2012):

La gestión de la información no es más que el proceso de organizar, evaluar, presentar, comparar los datos en un determinado contexto, controlando su calidad, de manera que esta sea veraz, oportuna, significativa, exacta y útil y que esta información esté disponible en el momento que se le necesite. Ella se encamina al manejo de la información, documentos, metodologías, informes, publicaciones, soportes y flujos en función de los objetivos estratégicos de una organización.

Según el Consejo de Estado (2011) la gestión de la información es:

Conjunto organizado de personas, procesos y recursos, incluyendo la información y sus tecnologías asociadas, que interactúan de forma dinámica, para satisfacer las necesidades informativas que posibilitan alcanzar los objetivos de una o varias o varias organizaciones. (p. 29)

Rojas (como se citó en Torres, 2015) refiere que:

La gestión de información, es el proceso mediante el cual se obtienen, despliegan o utilizan recursos básicos (económicos, físicos, humanos, materiales) para manejar información dentro y para la

sociedad a la que sirve. Tiene como elemento básico, la gestión del ciclo de vida de este recurso y se desarrolla en cualquier organización.

A partir del análisis de los conceptos de gestión de la información anteriormente expuestos se decide utilizar el brindado por Vidal y Araña (2012), al ser uno de los más completos porque, además de exponer todos los procesos que forman parte de la gestión de la información, le da mayor prioridad a la gestión documental.

Los sistemas de gestión de la información

Según Roblejo (2012) un sistema de gestión de la información es:

Aplicación que contiene un conjunto integrado de procesos, principalmente formales, que la organización conoce y sabe cómo utilizar (los informales no están excluidos), y son registrados en datos a través de bases de datos. Desarrollada en un entorno usuario-ordenador y operando, sobre un conjunto de datos estructurados (bases de datos), utilizando hardware y software computacional, redes de telecomunicaciones, técnicas de administración u otras formas de tecnología de información. Se caracteriza por la disponibilidad de la información cuando es necesaria. (p.16)

Según Medina (2005):

Un sistema de información (SI) se define (...) como un conjunto de componentes interrelacionados que incluyen hardware y software que capturan, almacenan, procesan y distribuyen la información para apoyar la toma de decisiones, el control, análisis y visión en una institución.

Según la Confederación de Empresarios de Andalucía (CEA, 2010):

Se entiende por “sistema de información” la conjunción de información con herramientas informáticas, es decir, con programas informáticos o software.

La autora de esta investigación, luego de haber analizado varias bibliografías acerca del tema, define como sistema de gestión de la información (SGI):

Herramientas informáticas que contribuyen, de manera positiva, a la gestión de la información en las distintas organizaciones. Son aplicaciones que modelan los diversos procesos que se llevan a cabo

diariamente en las organizaciones. Su función principal es satisfacer las necesidades de información de la institución donde se instaure y tener disponible dicha información siempre que sea necesaria.



Figura 1. Sistemas de Gestión de la Información. Elaboración propia

1.1.3. Principales sistemas de gestión de la información para organizaciones

En la actualidad existen diversas soluciones informáticas que se dedican a gestionar la información de múltiples empresas, entre ellas se encuentran:

SIGA

Es un sistema de gestión integral desarrollado en la Universidad Autónoma de San Luis Potosí (México). Su función es sistematizar, por medio de la gestión archivística y el uso de TIC, los fondos archivísticos que conforman las unidades de archivo de concentración e histórico del Congreso del Estado de San Luis Potosí, México, los cuales se encuentran físicamente resguardados en dos recintos legislativos, Presidente Juárez y Jardín Hidalgo. El sistema SIGA permite registrar nuevos expedientes, clasificarlos, generar reportes de acuerdo a una serie de parámetros que ya se encuentran definidos y permite, además, realizar diferentes tipos de búsqueda. Sin embargo, SIGA se dedica solamente a la gestión archivística de la información

referente al Estado de San Luis Potosí, no da la posibilidad de generar reportes personalizados, pues posee una única plantilla de reportes y su código fuente no está disponible públicamente.

Tekové

Es un sistema diseñado para el control y seguimiento de cada persona privada de libertad dentro de las penitenciarías y centros educativos de la República del Paraguay. Su función específica es la de controlar el estado procesal de las personas privadas de libertad, así como también el seguimiento correspondiente a los expedientes judiciales tales como, traslados a comparecencias, traslados a otras penitenciarias y/o centros educativos, traslados a hospitales, libertades y conformación del núcleo familiar del interno.

La herramienta Tekové ayudará a agilizar y transparentar la información penitenciaria, pues solamente no ayudará al trabajo de todos los órganos del sistema de Justicia, sino que también beneficiará a todos los internos, porque con esta ficha digital ellos mismos podrán saber cómo se mueve su causa.

A pesar de que esta herramienta constituye un hecho sin precedencia en la historia penitenciaria, no cumple con las funcionalidades que necesita el Departamento de Cuadros del CCPCC para gestionar la información referente a los cuadros de dirección de país, pues este sistema no permite realizar evaluaciones de las entidades que maneja, en este caso, personas privadas de libertad. Además, su código fuente no está disponible públicamente por estar asociado a órganos del sistema de Justicia.

Durango Digital

Es un sistema para la gestión gubernamental del estado de Durango, México. Dicho sistema permite disminuir tiempos y recursos en la gestión gubernamental del Despacho Ejecutivo, promueve el intercambio y flujo de información interna, en un ambiente de seguridad y confidencialidad. Además, facilita la toma de decisiones bajo un esquema simple, funcional y confiable y proporciona al gobierno del estado una fuente importante de localización y clasificación de las necesidades sociales.

Este sistema al igual que el resto, está diseñado para acciones específicas, en este caso, actividades referentes al gobierno de Durango, lo que imposibilita ajustarlo a las necesidades que especifica el Departamento de Cuadros del CCPCC, ya que no es de código abierto. Por tanto, no se puede adaptar este sistema al entorno deseado.

Módulo servidor para la gestión de la información en la Dirección de Cuadros de la Administración Provincial (AP) de Artemisa

Es un sistema que pretende mejorar los procesos de gestión de la información en la Dirección de Cuadros de la AP de Artemisa. El sistema cuenta con las funcionalidades correspondientes a la gestión de los cuadros existentes en la provincia y sus municipios. Además, permite gestionar la superación, vacaciones, evaluaciones, medidas disciplinarias, estimulaciones, chequeos médicos y movimientos de los cuadros en la provincia, y de esta forma garantizar la eficiencia y confiabilidad de esos procesos.

Como se puede apreciar, este es uno de los sistemas que más se ajusta a la solución que se propone en esta investigación a la situación que hoy tiene lugar en el Departamento de Cuadros del CCPCC, sin embargo, dicho sistema no cubre completamente las necesidades de información que presenta el departamento, pues los funcionarios que allí radican solicitan que la generación de reportes sea lo más flexible posible, o sea, que no exista una plantilla estricta de reportes sino que se puedan generar de acuerdo a las necesidades del momento. Para la presente investigación resultan como desventajas en este sistema que no permite gestionar los nomencladores¹ y que su implementación está definida en el lenguaje de programación Java, lo que dificulta la integración con el sistema en desarrollo por parte de CISED.

SRCC

Es un sistema para la gestión de cuadros en las organizaciones cubanas, desarrollado por cadetes insertados del MININT en la UCI. Permite registrar la información referente a los cuadros como sus trayectorias, estudios realizados, relaciones con familiares, tenencia de vehículos o armamento, actualizaciones políticas y padecimientos. Además, da la posibilidad de mantener actualizadas las reservas por cada cargo, emitir evaluaciones a los propios cuadros y generar reportes de manera configurable.

Este sistema de igual forma se asemeja en gran medida a la solución que propone la autora de esta investigación, sin embargo, no se ajusta a las necesidades que especifica el Departamento de Cuadros del CCPCC; además dicho sistema no ha sido implantado en ninguna organización, por lo que no está validado funcionalmente para poder asumirlo.

¹ Lista de términos o nombres de algún elemento que ya está definida.

El estudio realizado de estos y otros sistemas similares, ha demostrado que ninguno de ellos satisface las necesidades de información que hoy presenta el Departamento de Cuadros del CCPCC. Algunos porque no son de código abierto, lo que imposibilita la obtención y modificación del código fuente para ser utilizado por la organización, y otros porque, a pesar de que se asemejan bastante, están desarrollados en lenguajes de programación diferentes y dificultan la integración o no están validados funcionalmente en otras instituciones. Sin embargo, a pesar de que ninguno de estos sistemas constituye una solución integral al problema en cuestión, su estudio permitió obtener una visión más amplia de la solución y proporcionó funcionalidades de interés para enriquecer la solución, como por ejemplo la tenencia de un módulo que posea las funcionalidades correspondientes para mantener el sistema lo más extensible posible, de manera tal que si en algún momento se desea aumentar la información predeterminada que este posee, no haya necesidad de recurrir a la implementación de un nuevo sistema. Lo mencionado anteriormente enfatizó más la necesidad de desarrollar un sistema específico para el Departamento de Cuadros del CCPCC que responda a las necesidades planteadas.

1.1.4. Estado actual del proceso de gestión de la información referente a los cuadros en el CCPCC

Para conocer el estado actual del proceso de gestión de la información referente a los cuadros en el CCPCC se han propuesto los siguientes indicadores: acceso a la información, corrección de errores, recopilación de información y comunicación entre entidades; tales se detallan más adelante.

El método empleado para obtener esta información fue la entrevista. Se realizó de forma planificada y estructurada a los trabajadores del Departamento de Cuadros y a un funcionario general del CCPCC.

La entrevista perseguía los siguientes objetivos:

- Definir el problema de investigación y la importancia del Sistema web para la gestión de la información del Departamento de Cuadros del CCPCC.
- Trazar el objetivo general del Sistema web para la gestión de la información del Departamento de Cuadros del CCPCC.
- Identificar qué procesos del Departamento de Cuadros iban a ser llevados al sistema web.
- Determinar los requisitos funcionales.

Acceso a la información

La información que se necesita para tomar decisiones respecto a determinados cuadros muchas veces se encuentra descentralizada, ya sea en varias computadoras o archivos físicos, ubicados en la entidad donde radica el cuadro, o sea, que puede estar en cualquier lugar del país.

El hecho de que la información no se encuentre centralizada, evita que se pueda acceder a ella en cualquier momento y desde cualquier parte del país, por lo que se evidencia que hay limitaciones en el acceso, incidiendo esto, notablemente, en demoras al momento de reunir información.

Corrección de errores

Los informes muchas veces se generan con errores y se hace complicado corregirlos en el momento en que se detectan, la mayoría de las veces hay que volverlos a crear. Cuando estos errores no son tan relevantes los informes se entregan en esas propias condiciones, lo que trae consigo baja calidad en los mismos.

Recopilación de información

El proceso de recopilación de información para generar informes es bastante engorroso. A pesar de que en el Departamento de Cuadros utilizan tecnologías como el correo electrónico, para enviar y recibir información, el proceso sigue siendo trabajoso, pues el correo electrónico tiene limitantes en cuanto a volumen de datos se refiere. También por esta vía muchas veces los datos llegan incompletos, lo que provoca que no se ajusten a la realidad, pues la persona que los envía puede omitir muchos de ellos que son de carácter obligatorio.

Comunicación entre entidades

Los cuadros de dirección se encuentran dispersos por todo el país, según la entidad a la que pertenezcan. La comunicación entre los directivos de cada una de estas entidades es por vía telefónica, correo electrónico y, en puntuales casos, a través de viajes interprovinciales. Las vías de comunicación tan poco seguras que utilizan, ponen en riesgo la confidencialidad de los datos que se tratan.

1.2. Metodología de desarrollo de software

Determinados especialistas como Arias, Blanco y Bagarotti (2013) han referido que:

Desde el punto de vista informático una metodología de desarrollo de software es el conjunto de procedimientos, técnicas, herramientas y un soporte documental a la hora de desarrollar un producto de software, que indica quién, cuándo y cómo hacer algo. La presencia de una metodología en el desarrollo de un proyecto garantiza un producto con más calidad y reduce el tiempo de entrega del producto ya que al tener una mejor planificación la producción permite cumplir con la fecha establecida. (p.14)

En consonancia con Arias et al. (2013), INTECO (2009) afirma que:

Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de software detallado y completo; un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. (p.39)

Visto esto se puede concluir que, en el desarrollo de software es importante la selección de una adecuada metodología para lograr el éxito del producto final. A continuación, se describe la metodología seleccionada para guiar el proceso de desarrollo de software de la presente investigación, su elección se debe a que es la empleada por el centro CISED:

AUP-UCI es una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. En aras de aumentar la calidad del software que se produce esta metodología se apoya en el Modelo CMMI-DEV v1.3. (por sus siglas en inglés *Capability Maturity Model Integration Development*, Integración de Modelos de Madurez de Capacidades para Desarrollo). El mismo constituye una guía para aplicar las mejores prácticas y obtener un producto o servicio con calidad en una entidad desarrolladora (Rodríguez, 2015).

1. **Inicio:** El objetivo de esta fase es llevar a cabo las actividades relacionadas con la planeación del proyecto. Se realiza un estudio inicial de la organización que actúa como cliente y se obtiene

información clave acerca del alcance del proyecto, se realizan estimaciones de tiempo y esfuerzo, y finalmente se decide si se ejecuta o no.

2. **Ejecución:** En esta etapa se ejecutan las actividades requeridas para desarrollar el software. Durante el desarrollo se modela el negocio, se obtienen los requisitos, se elabora la arquitectura y el diseño, se implementa y se libera el producto.
3. **Cierre:** En esta fase se analizan tanto los resultados del proyecto como su ejecución y se llevan a cabo las actividades formales de cierre de proyecto.

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

1.3. Herramientas y tecnologías

Las herramientas son aplicaciones, programas o meramente instrucciones que ofrecen la posibilidad de realizar determinadas funcionalidades con disímiles propósitos. De otro modo, las tecnologías son el conjunto de conocimientos técnicos, ordenados científicamente que permiten diseñar y crear bienes y servicios (Noda y Campanioni, 2015). A continuación, se muestran todas las herramientas y tecnologías utilizadas en el desarrollo del Sistema de web para la gestión de la información en el Departamento de Cuadros del CCPCC, estas fueron seleccionadas porque el sistema que se desea implementar forma parte del sistema para el control de militantes del PCC.

1.3.1. Lenguaje de modelado

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

de los lenguajes naturales, sino que, habitualmente, lo hacen en forma de iconos o dibujos para facilitar la visualización formal de los conceptos, que son abstractos (Gonzalez y Hug, 2015).

UML

UML (por sus siglas en inglés, *Unified Modeling Language*) es un lenguaje de propósito general para el modelado orientado a objetos. UML es también un lenguaje de modelamiento visual que permite una abstracción de un sistema y sus componentes (Rumbaugh, Jacobson y Booch, 1999).

Mediante este lenguaje se pueden especificar todas las decisiones de análisis y diseño, construyéndose así modelos precisos, no ambiguos y completos. Además, puede conectarse con lenguajes de programación (ingeniería directa e inversa) y permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones). (Peralta y Durán, 2015, p.15)

Para llevar a cabo el sistema web se utilizará UML en su versión 2.0 por las siguientes características:

- **Visualizar:** Permite expresar de una forma gráfica un sistema de forma tal que otro lo pueda entender.
- **Especificar:** Permite especificar cuáles son las características de un sistema antes de su construcción.
- **Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados.
- **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

1.3.2. Herramientas CASE

Dentro de las herramientas claves para el desarrollo de aplicaciones informáticas se encuentran las herramientas de Ingeniería de Software Asistida por Computadora (por sus siglas en inglés *Computer-Aided Software Engineering*, CASE). Según Pressman (2002) este tipo de herramienta se utiliza para ayudar a actividades del proceso de software tales como la ingeniería de requisitos, el diseño, el desarrollo de aplicaciones y las pruebas.

Visual Paradigm

Visual Paradigm es una herramienta UML que permite modelar el ciclo de vida completo del desarrollo de un software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite construir todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (Peralta y Durán, 2015).

Para el proceso de desarrollo del sistema web se emplea Visual Paradigm en su versión 8.0, por ser una herramienta de software libre que permite realizar ingeniería tanto inversa como directa.

ER/Studio

ER/Studio es una herramienta CASE para modelado de datos, fácil de usar y multinivel para el diseño de bases de datos a nivel físico y lógico. Esta herramienta está equipada para crear y manejar diseños de bases de datos funcionales y confiables. Además, ofrece fuertes capacidades de diseño lógico, sincronización bidireccional de los diseños físicos y lógicos, construcción automática de bases de datos, documentación y fácil creación de reportes (Danysoft, 2006).

Según plantea Danysoft (2006):

ER/Studio es una herramienta de base de datos que le ayuda a diseñar, generar y mantener aplicaciones de base de datos de calidad y alto rendimiento. Desde un modelo lógico de sus requerimientos de información y reglas del negocio que definen su base de datos, hasta un modelo físico optimizado por las características específicas de su base de datos de destino, ER/Studio le permite visualizar la estructura adecuada, los elementos clave y un diseño optimizado de su base de datos. (p. 3)

Para el desarrollo de la aplicación se utilizará para el modelado de las bases de datos la herramienta ER/Studio en su versión 8.0.

1.3.3. Lenguaje de programación

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

A continuación, se muestra el lenguaje de programación utilizado para el desarrollo del Sistema web para la gestión de la información en el Departamento de Cuadros del CCPCC:

Python

Python es un lenguaje de programación fuerte y fácil de aprender, además posee un enfoque de programación orientada a objetos. Este lenguaje cuenta con estructuras de datos eficientes y de alto nivel. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para el desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas (Rossum, 2009).

Según refiere Dayley (2007) las ventajas de usar Python son:

- **Portabilidad:** Funciona en casi todos los sistemas operativos.
- **Fácil:** Es muy fácil comenzar a desarrollar programas en Python. Su sintaxis clara, gracias a la notación indentada (con márgenes), hace al código cómodo de leer.
- **Poder:** Posee extensiones escritas en el propio lenguaje de Python que permiten, entre otras cosas, el acceso a la base de datos, la edición de audio y video, interfaz gráfica de usuario y el desarrollo web.
- **Código abierto:** Python es un lenguaje de código abierto, lo que significa que se puede utilizar libremente.

Para el desarrollo del sistema web se utilizó Python en su versión 2.7, ya que es la última versión estable de Python 2.x y, según sus desarrolladores, la última de esta serie (Sánchez, 2015).

1.3.4. Servidor web

Un servidor web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución realizadas por un cliente. El servidor web se encarga de contestar a estas peticiones de forma adecuada a través del protocolo HTTP (por sus siglas en inglés *Hypertext Transfer Protocol*, Protocolo de Transferencia de Hipertexto), entregando como resultado una página web. El servidor web también contiene intérpretes de diferentes lenguajes de programación, estos son los encargados de ejecutar el código embebido dentro del HTML de las páginas web antes de enviar el resultado al cliente (Guerra y Matos, 2013).

Apache

Apache es uno de los servidores web más utilizados en la red. Es uno de los mayores triunfos del software libre, pues logró vencer a competidores como Netscape y Microsoft, como plataforma de servidores web. Es un servidor modular, multiplataforma, extensible, popular (fácil de conseguir ayuda) y gratuito (Kabir, 2002).

Características que hacen popular a este software libre:

- Provee una muy buena base para la seguridad del sistema, gracias a los módulos de Autenticación, Autorización y Control de Acceso al Servidor Web.
- La extensibilidad por módulos lo hace extremadamente flexible y fácil de usar, así como de configurar.

Para el desarrollo del sistema se utiliza el servidor web Apache en su versión 2.2, con el módulo `mod_python` 3.0. Dicho módulo permite que Apache pueda interpretar código Python.

1.3.5. Entorno de desarrollo

Un entorno de desarrollo integrado, o bien conocido como IDE (por sus siglas en inglés *Integrated Development Environment*), “es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios” (Galiano, 2015, p.119).

PyCharm IDE

PyCharm IDE es un IDE del lenguaje de programación Python que posee un juego completo de herramientas para el desarrollo productivo. Entre las características que valen destacar de este entorno de desarrollo se encuentran:

1. **Asistencia de codificación inteligente:** Proporciona soporte de primera clase para Python, JavaScript, CSS, plantillas de idiomas populares y más. Además, posibilita detectar errores y corregir el código sobre la marcha.
2. **Navegación de código inteligente:** Usa la búsqueda inteligente para saltar a cualquier clase, archivo o símbolo, o incluso a cualquier ventana de acción del IDE o herramientas.
3. **Refactorización rápida y segura:** Permite limpiar el código de manera segura, lo que ayuda a realizar cambios en todo el proyecto.

Para llevar a cabo la implementación del sistema web se hará uso de PyCharm en su versión 4.5.4.

1.3.6. Sistema Gestor de Bases de Datos

La autora de esta investigación coincide con Bertino y Martino (1993) en que un Sistema Gestor de Base de Datos (SGBD) es un software que permite la definición de bases de datos. Un SGBD les da a los usuarios la posibilidad de procesar, describir, administrar y recuperar los datos almacenados en una base de datos.

PostgreSQL

Según PostgreSQL-es (2010):

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD² y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarles a otras bases de datos comerciales. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

² BSD: Distribución de Software de Berkeley, (por sus siglas en inglés *Berkeley Software Distribution*, BSD). Este tipo de licencia permite a los programadores utilizar, modificar y distribuir a terceros el código fuente del software original.

Para desarrollar el Sistema web para la gestión de la información del Departamento de Cuadros del CCPCC se va a utilizar como SGBD PostgreSQL en su versión 9.4.

1.3.7. Marcos de trabajo

Un framework es un marco de trabajo diseñado para apoyar el desarrollo de aplicaciones web. Es un software o conjunto de librerías que tratan de facilitar aquellas actividades comunes que son realizadas durante el desarrollo de una aplicación web, como, por ejemplo: acceso a la base de datos, uso de plantillas, manejo de sesiones; además, promueve la reutilización de código (Mendoza, 2011).

Django

Django es un framework que promueve el desarrollo rápido, se construyen aplicaciones en cuestión de días y con el conocimiento suficiente esos días se pueden reducir a horas. Django impulsa el desarrollo de código limpio al promover buenas prácticas de desarrollo web, sigue el principio DRY (conocido también como Una vez y sólo una). Usa una modificación de la arquitectura Modelo-Vista-Controlador (MVC), llamada MTV (*Model – Template – View*), que sería Modelo-Plantilla-Vista, esta forma de trabajar permite que sea pragmático³(Infante, 2012).

Para el desarrollo del sistema web se decidió usar Django en su versión 1.7, pues se centra en el desarrollo rápido, la reutilización de código y la seguridad. Además, cuenta con una comunidad de desarrollo online bastante amplia.

Conclusiones del capítulo

Para llevar a cabo el desarrollo del Sistema web para la gestión de la información del Departamento de Cuadros del CCPCC se realizó un estudio de los conceptos principales asociados a la gestión de la información y a los sistemas de gestión de la información, y se pudo concluir que, aunque varios autores ofrezcan definiciones bastantes similares, estos dos conceptos difieren entre ellos, pues el primero se limita al propio manejo de la información en las organizaciones, mientras que el segundo ya incluye a las

³ Significa práctico, la ejecución o la realización de las acciones y no a la teoría o a la especulación.

tecnologías informáticas como soporte a esa gestión de la información. Se definió en este capítulo, un concepto propio de sistema de gestión de la información.

Se evidenció el papel protagónico que juegan las TIC en las organizaciones, específicamente el de los sistemas de gestión de la información, que contribuyen a que las empresas puedan ser más productivas, tengan mayor calidad y que sus procesos internos se ejecuten en menos tiempo.

Se hizo un estudio de los sistemas homólogos existentes, y se pudo asegurar que era necesario implementar un sistema nuevo para el Departamento de Cuadros del CCPCC, pues ninguno de los ya existentes modela los requisitos que demanda el cliente en este caso, tales son: procesos de solicitud, presentación de la información y evaluación de los cuadros de dirección, y los que presentan funcionalidades similares no están validados funcionalmente en una institución.

Para guiar el proceso de desarrollo del sistema web para el Departamento de Cuadros se adoptó como metodología de desarrollo de software a AUP-UCI debido a que es la metodología que se emplea en el centro CISED. El lenguaje de modelado previsto es UML 2.0 y como herramientas CASE Visual Paradigm en su versión 8.0 y ER/Studio en su versión 8.0. Por otra parte, Python es el lenguaje de programación escogido mientras que PyCharm 4.5.4 es seleccionado como IDE de desarrollo. Se utilizará a PostgreSQL 9.4 como Sistema Gestor de Bases de Datos. Para llevar a cabo la implementación del sistema web se utilizará el framework Django en su versión 1.7.

Capítulo 2: Análisis y diseño de la solución

Introducción

En este capítulo se evidencian las principales características de la solución propuesta. Se hace un levantamiento de los requisitos funcionales y no funcionales necesarios para lograr que el sistema web funcione correctamente. Además, se construyen los artefactos correspondientes al análisis y diseño acorde con la metodología seleccionada en el capítulo anterior.

2.1. Modelo de dominio

Un modelo de dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. También se le denomina modelo conceptual, modelo de objetos del dominio y modelo de objetos de análisis. Al utilizar la notación UML, un modelo de dominio se representa con un conjunto de diagramas de clases en los que no se define ninguna operación. Su principal objetivo es definir las interrelaciones de los objetos más importantes representados mediante clases. Además, desempeña un papel clave en la comprensión del entorno actual (Larman, 2003).

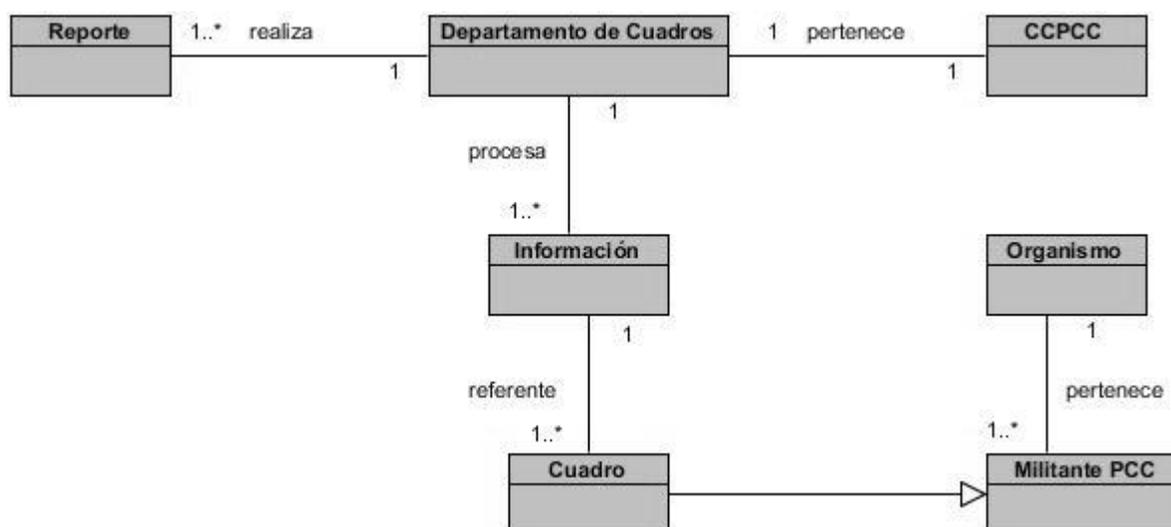


Figura 2. Modelo de dominio. Elaboración propia

2.1.1. Descripción de los conceptos del dominio

CCPCC: entidad que representa al Comité Central del PCC, es el encargado de manejar toda la información de los militantes del PCC proveniente de cada organismo del país.

Departamento de cuadros: entidad que representa uno de los departamentos del CCPCC, su función es llevar el control de todos los cuadros de dirección de los organismos del país y darles un seguimiento.

Información: entidad que representa la información que el Departamento de Cuadros procesa.

Cuadro: entidad que representa a las personas que son miembros de PCC y que poseen un cargo de dirección.

Militante PCC: entidad que representa a los militantes del PCC de todos los organismos del país.

Organismo: entidad que representa a todos los organismos o entidades del país que desempeñan una determinada labor.

Reportes: entidad que representa a los reportes que son realizados una vez que el Departamento de Cuadros procesa la información referente a los cuadros.

2.2. Descripción de la propuesta de solución

El Sistema web para la gestión de la información en el Departamento de Cuadros del CCPCC constituye un subsistema del sistema para el control de militantes del PCC, este estará compuesto por 5 módulos a petición del cliente: Datos Personales, Nomencladores, Evaluación y Caracterización, Movimientos e Investigaciones y Reportes. En la presente investigación se implementarán cada uno de ellos.

El módulo Datos Personales se encarga de gestionar todos los datos personales relacionados con los cuadros del PCC de todo el país, los cuales se pueden agrupar de la siguiente forma: datos personales, laborales, referentes a las organizaciones u organismos a los que pertenece, a su trayectoria laboral y militar, los lugares de residencia, los estudios realizados, la preparación recibida y, además, los datos personales de las personas relacionadas con ellos.

El módulo Nomencladores tiene la función de mantener el sistema lo más extensible posible, de manera tal que se pueda aumentar el contenido por defecto del sistema actual sin que haya necesidad de adquirir un sistema nuevo. Este módulo ofrece la posibilidad de gestionar todos los nomencladores.

El módulo Evaluación y Caracterización permite gestionar la información referente a los reconocimientos, sanciones y evaluaciones de los cuadros del PCC, para así tener un mayor control y poder darles seguimiento.

El módulo Movimientos e Investigaciones ofrece la posibilidad de gestionar los movimientos que se le hacen a los cuadros y las investigaciones. Los posibles movimientos que se pueden realizar son promoción, oficializar el cargo, reincorporación, traslado y liberación, y con respecto a las investigaciones permite llevar un control de todos los anónimos y denuncias que han tenido lugar.

El módulo Reportes brinda el servicio de generar reportes y exportarlos en formato PDF.

2.3. Requisitos del sistema

En el proceso de ingeniería de software, los requisitos se utilizan como datos de entrada en la etapa de diseño del producto que se desea lograr. Estos representan qué debe hacer el sistema, pero no cómo hacerlo y, además, incorporan las cualidades o propiedades que el mismo debe poseer (Sommerville, 2007).

2.3.1. Requisitos funcionales (RF)

Los requisitos funcionales definen el comportamiento interno de un software, son condiciones que el sistema ha de cumplir. Estos muestran las funcionalidades que deben satisfacerse para cumplir con las especificaciones de software (Sommerville, 2007).

Tabla 1. Requisitos funcionales del Módulo Datos Personales. Elaboración propia.

Módulo Datos Personales	
RF_1: Insertar datos personales del cuadro	RF_3: Eliminar datos personales del cuadro
RF_2: Modificar datos personales del cuadro	RF_4: Obtener datos personales del cuadro

RF_5: Insertar enfermedades del cuadro	RF_25: Insertar datos referentes a los estudios realizados por el cuadro a partir de la enseñanza media
RF_6: Modificar enfermedades del cuadro	RF_26: Modificar datos de los estudios realizados por el cuadro a partir de la enseñanza media
RF_7: Eliminar enfermedades del cuadro	RF_27: Eliminar datos de los estudios realizados por el cuadro a partir de la enseñanza media
RF_8: Listar enfermedades del cuadro	RF_28: Listar los estudios realizados por el cuadro a partir de la enseñanza media
RF_9: Insertar datos laborales del cuadro	RF_29: Insertar datos de los estudios actuales del cuadro
RF_10: Modificar datos laborales del cuadro	RF_30: Modificar datos de los estudios actuales del cuadro
RF_11: Eliminar datos laborales del cuadro	RF_31: Eliminar datos de los estudios actuales del cuadro
RF_12: Obtener datos laborales del cuadro	RF_32: Listar los estudios que realiza actualmente el cuadro
RF_13: Insertar datos sobre las organizaciones a las que pertenece el cuadro	RF_33: Insertar datos de la preparación recibida por el cuadro
RF_14: Modificar datos de las organizaciones a las que pertenece el cuadro	RF_34: Modificar datos de la preparación recibida por el cuadro
RF_15: Eliminar datos de las organizaciones a las que pertenece el cuadro	RF_35: Eliminar datos de la preparación recibida por el cuadro
RF_16: Listar organizaciones a las que pertenece un cuadro	RF_36: Listar escuelas de preparación cursadas por el cuadro
RF_17: Insertar datos a la trayectoria laboral del cuadro	RF_37: Insertar datos a trayectoria militar del cuadro
RF_18: Modificar datos de la trayectoria laboral de cuadro	RF_38: Modificar datos de trayectoria militar del cuadro
RF_19: Eliminar datos de la trayectoria laboral del cuadro	RF_39: Eliminar datos de la trayectoria militar del cuadro
RF_20: Obtener trayectoria laboral	
RF_21: Insertar datos de lugares de residencia del cuadro	
RF_22: Modificar datos de lugares de residencia del cuadro	
RF_23: Eliminar datos de lugares de residencia del cuadro	
RF_24: Listar lugares de residencia	

RF_40: Obtener trayectoria militar del cuadro	RF_49: Insertar datos del arma de fuego del cuadro
RF_41: Insertar datos de la estancia del cuadro en el exterior	RF_50: Modificar datos del arma de fuego del cuadro
RF_42: Modificar datos de la estancia del cuadro en el exterior	RF_51: Eliminar datos del arma de fuego del cuadro
RF_43: Eliminar datos de la estancia del cuadro en el exterior	RF_52: Obtener los datos del arma de fuego del cuadro
RF_44: Listar datos de la estancia del cuadro en el exterior	RF_53: Insertar datos de la persona relacionada con el cuadro
RF_45: Insertar datos del vehículo del cuadro	RF_54: Modificar datos de la persona relacionada con el cuadro
RF_46: Modificar datos del vehículo del cuadro	RF_55: Eliminar datos de la persona relacionada con el cuadro
RF_47: Eliminar datos del vehículo del cuadro	RF_56: Obtener los datos de la persona relacionada con el cuadro
RF_48: Obtener los datos del vehículo del cuadro	

Tabla 2. Requisitos funcionales del Módulo Evaluaciones y Caracterización. Elaboración propia.

Módulo Evaluaciones y Caracterización

RF_57: Realizar evaluación del cuadro	RF_63: Eliminar reconocimiento de un cuadro
RF_58: Modificar evaluación del cuadro	RF_64: Listar reconocimientos del cuadro
RF_59: Eliminar evaluación del cuadro	RF_65: Insertar sanción a un cuadro
RF_60: Obtener evaluación del cuadro	RF_66: Modificar sanción de un cuadro
RF_61: Insertar reconocimiento a un cuadro	RF_67: Eliminar sanción de un cuadro
RF_62: Modificar reconocimiento de un cuadro	RF_68: Listar sanciones del cuadro

Tabla 3. Requisitos funcionales del Módulo Nomencladores. Elaboración propia.

Módulo Nomencladores	
RF_69: Insertar organismo de dirección	RF_97: Insertar tipo de curso
RF_70: Modificar organismo de dirección	RF_98: Modificar tipo de curso
RF_71: Eliminar organismo de dirección	RF_99: Eliminar tipo de curso
RF_72: Listar organismos de dirección	RF_100: Listar tipos de curso
RF_73: Insertar propuesta de movimiento	RF_101: Insertar curso
RF_74: Modificar propuesta de movimiento	RF_102: Modificar curso
RF_75: Eliminar propuesta de movimiento	RF_103: Eliminar curso
RF_76: Listar propuestas de movimiento	RF_104: Listar cursos
RF_77: Insertar tipo de propuesta de movimiento	RF_105: Insertar tipo de escuela
RF_78: Modificar tipo de propuesta de movimiento	RF_106: Modificar tipo de escuela
RF_79: Eliminar tipo de propuesta de movimiento	RF_107: Eliminar tipo de escuela
RF_80: Listar tipos de propuesta de movimiento	RF_108: Listar tipos de escuelas
RF_81: Insertar tipo de investigación	RF_109: Insertar escuela
RF_82: Modificar tipo de investigación	RF_110: Modificar escuela
RF_83: Eliminar tipo de investigación	RF_111: Eliminar escuela
RF_84: Listar tipos de investigaciones	RF_112: Listar escuelas
RF_85: Insertar país	RF_113: Insertar tipo de reconocimiento
RF_86: Modificar país	RF_114: Modificar tipo de reconocimiento
RF_87: Eliminar país	RF_115: Eliminar tipo de reconocimiento
RF_88: Listar países	RF_116: Listar tipos de reconocimientos
RF_89: Insertar idioma	RF_117: Insertar reconocimiento
RF_90: Modificar idioma	RF_118: Modificar reconocimiento
RF_91: Eliminar idioma	RF_119: Eliminar reconocimiento
RF_92: Listar idiomas	RF_120: Listar reconocimientos
RF_93: Insertar tipo de estudio	RF_121: Insertar causa de baja militar
RF_94: Modificar tipo de estudio	RF_122: Modificar causa de baja militar
RF_95: Eliminar tipo de estudio	RF_123: Eliminar causa de baja militar
RF_96: Listar tipos de estudios actuales	

RF_124: Listar causas de baja militar	RF_133: Insertar ciudadanía
RF_125: Insertar causa de cambio de trabajo	RF_134: Modificar ciudadanía
RF_126: Modificar causa de cambio de trabajo	RF_135: Eliminar ciudadanía
RF_127: Eliminar causa de cambio de trabajo	RF_136: Listar ciudadanías
RF_128: Listar causas de cambio de trabajo	RF_137: Insertar nacionalidad
RF_129: Insertar tipo de sanción	RF_138: Modificar nacionalidad
RF_130: Modificar tipo de sanción	RF_139: Eliminar nacionalidad
RF_131: Eliminar tipo de sanción	RF_140: Listar nacionalidades
RF_132: Listar tipos de sanciones	

Tabla 4. Requisitos funcionales del Módulo Movimientos e Investigaciones. Elaboración propia.

Módulo Movimientos e Investigaciones	
RF_141: Insertar movimiento de un cuadro	RF_145: Insertar investigación de un cuadro
RF_142: Modificar movimiento de un cuadro	RF_146: Modificar investigación de un cuadro
RF_143: Eliminar movimiento de un cuadro	RF_147: Eliminar investigación de un cuadro
RF_144: Listar movimientos de un cuadro	RF_148: Listar investigaciones de un cuadro

Tabla 5. Requisitos funcionales del Módulo Reportes. Elaboración propia.

Módulo Reportes
RF_149: Generar reportes de los cuadros
RF_150: Exportar reportes de los cuadros

2.3.2. Requisitos no funcionales (RnF)

De acuerdo con Sommerville (2007), los requisitos no funcionales son aquellos requisitos que no describen información a guardar, ni funciones a realizar, sino que son propiedades que hacen al producto atractivo, usable, rápido o confiable. Además, se conocen como un conjunto de características de calidad, que es necesario tener en cuenta al diseñar e implementar el software.

- RnF_1:** El sistema debe ser una aplicación web.
- RnF_2:** El sistema debe cumplir con un conjunto de patrones de diseño seleccionados por el equipo de desarrollo.
- RnF_3:** El código del sistema debe cumplir con estándares de codificación definidos por el equipo de desarrollo.
- RnF_4:** El sistema debe ser implementado utilizando herramientas, lenguajes y tecnologías libres.
- RnF_5:** El sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras.
- RnF_6:** El sitio contará con una estructura sencilla y uniforme.
- RnF_7:** El sistema debe estar disponible en idioma español.
- RnF_8:** Del lado del servidor la computadora empleada debe ser Pentium 4 a 1GHz o superior, mínimo 4Gb de RAM⁴ y 40Gb o superior de disco duro.
- RnF_9:** Del lado del cliente la computadora empleada debe ser Pentium 4 a 1GHz o superior, mínimo 512Mb de RAM y 40Gb o superior de disco duro.
- RnF_10:** Del lado del servidor debe instalarse el intérprete de Python en su versión 2.7.5, el marco de trabajo Django en su versión 1.7, la librería wkhtmltopdf en su versión 0.12.3.2 y el SGBD PostgreSQL en su versión 9.4.
- RnF_11:** Del lado del cliente debe instalarse como navegador web Firefox en su versión 35 o superior o Chrome en su versión 32 o superior.

2.4. Historias de Usuario (HU)

Las historias de usuario son una técnica utilizada en las metodologías de desarrollo ágiles para especificar los requisitos de software. Se describe brevemente y en un lenguaje natural, las características que el sistema debe poseer. Cada historia de usuario debe ser lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en un corto período de tiempo (Canós, Panadés y Letelier, 2006).

⁴ RAM del inglés *Random Access Memory*

A continuación, se presentan algunas de las historias de usuario definidas para el desarrollo de la propuesta de solución planteada en la presente investigación, el resto de las historias de usuario se pueden encontrar en el Anexo 2.

Tabla 6. HU Insertar datos personales del cuadro. Elaboración propia.

Historia de usuario	
Número: HU_1	Nombre del requisito: Insertar datos personales del cuadro
Programador: Rachel Morales Izquierdo	Iteración Asignada: 2
Prioridad: Muy Alta	Tiempo Estimado: 2
Tiempo Real: 1	
Descripción: El sistema permitirá insertar datos específicos del cuadro.	
Observaciones: Debe existir previamente un militante para poder crearlo a cuadro e insertarle los datos personales.	
Prototipo de interfaz:	
<p>Datos Personales</p> <p>Lugar de Nacimiento: <input type="text"/></p> <p>Provincia de Nacimiento: <input type="text"/></p> <p>Ciudadanía: <input type="text"/></p> <p>Color de Ojos: <input type="text"/></p> <p>Color de Pelo: <input type="text"/></p> <p>Estatura: <input type="text"/></p> <p>Peso (Kgs): <input type="text"/></p> <p>Teléfono Personal: <input type="text"/></p> <p>Correo Personal: <input type="text"/></p> <p>Idiomas: <input type="text"/></p> <p>Dominio básico de informática: <input type="checkbox"/></p> <p>Militancia política: <input type="text"/></p> <p>Fecha ingreso: <input type="text" value="09/05/2016"/></p> <p>Estado de Salud: <input type="text" value="Bueno"/></p> <p>Ubicación en tiempo de guerra: <input type="text"/></p> <p style="text-align: right;">ADICIONAR</p>	

Tabla 7. HU Realizar evaluación del cuadro. Elaboración propia.

Historia de usuario	
Número: HU_57	Nombre del requisito: Realizar evaluación del cuadro
Programador: Rachel Morales Izquierdo	Iteración Asignada: 3
Prioridad: Alta	Tiempo Estimado: 2
Tiempo Real:	
Descripción: El sistema permitirá realizar la evaluación correspondiente a un cuadro de acuerdo al valor asignado a los indicadores en el sistema y a su vez, insertará la evaluación obtenida.	
Observaciones: Se debe de haber insertado previamente en el sistema los posibles valores de cada indicador para poder emitir la evaluación	
Prototipo de interfaz:	

Evaluación

<p>Período que se evalúa</p> <p>Desde: <input type="text" value="09/05/2016"/></p> <p>Hasta: <input type="text" value="09/05/2016"/></p> <p>Principales aspectos a evaluar</p> <p>Actitud mantenida en el ejercicio de sus funciones: <input type="text"/></p> <p>Resultados de su desempeño: <input type="text"/></p> <p>Capacidad de dirección demostrada: <input type="text"/></p>	<p>Influencia y exigencia hacia los subordinados: <input type="text"/></p> <p>Cualidades personales: <input type="text"/></p> <p>Capacidad intelectual: <input type="text"/></p> <p>Evaluación final</p> <p><input type="radio"/></p> <p>Conclusiones de la evaluación: <input type="text"/></p>	<p>Texto de la evaluación</p> <p>Señalamientos: <input type="text"/></p> <p>Recomendaciones: <input type="text"/></p> <p>EVALUADOR</p> <p>Nombre y Apellidos: <input type="text"/></p> <p>Cargo: <input type="text"/></p> <p>Fecha: <input type="text" value="09/05/2016"/></p>
---	---	---

[ADICIONAR](#)

2.5. Planificación

En este punto se efectúa una planificación del sistema por etapas, donde se aplicarán diferentes iteraciones, y por cada iteración el cliente ha de recibir una versión nueva (Núñez, 2010). A continuación, se define el plan de entrega y el plan de iteraciones que presidirán el proceso de desarrollo.

2.5.1. Plan de entrega

Una vez definidas las historias de usuario es conveniente proceder a la planificación de la fase de ejecución. Se establecerán para ello 4 iteraciones, en las cuales se codificarán las 150 historias de usuario obtenidas.

Tabla 8. Plan de entrega. Elaboración propia.

<i>Entregable</i>	<i>Iteración</i>	<i>Fin de la iteración</i>
Módulo Nomencladores	1	Febrero 2016
Módulo Datos Personales	2	Febrero 2016
Módulo Evaluaciones y Caracterización	3	Marzo 2016
Módulo Movimientos e Investigaciones	3	Marzo 2016
Módulo Reportes	4	Abril 2016

2.5.2. Plan de iteraciones

El plan de iteraciones es una planificación donde el cliente y los desarrolladores acuerdan el tiempo que debe demorar cada historia de usuario en ser desarrollada. El cliente escoge, según sus necesidades, cuáles son las historias que deben estar terminadas al concluir cada iteración del plan (Beck y Andres, 2004).

La prioridad de las historias de usuarios fue establecida de acuerdo a la importancia que tenían estas en el negocio y al nivel de dependencia entre las distintas funcionalidades. La implementación del sistema propuesto tendrá una duración de 12 semanas. La tabla que sigue a continuación muestra el plan elaborado para llevar a cabo su desarrollo.

Tabla 9. Plan de iteraciones. Elaboración propia.

<i>Iteración</i>	<i>No. HU a implementar</i>	<i>Descripción de la iteración</i>	<i>Semanas estimadas</i>
1	Desde HU_69 hasta HU_140	En esta iteración se implementarán las historias de usuario que tengan una prioridad en el negocio <i>Muy Alta</i>	2
2	Desde HU_1 hasta HU_56	En esta iteración se implementarán las historias de usuario que tengan una prioridad en el negocio <i>Muy Alta</i>	2
3	Desde HU_57 hasta HU_68, desde HU_141 hasta HU_148	En esta iteración se implementarán las historias de usuario que tengan una prioridad en el negocio <i>Alta</i>	6
4	HU_149 HU_150	En esta iteración se implementarán las historias de usuario que tengan una prioridad en el negocio <i>Alta</i>	2

2.6. Diseño de la propuesta de solución

La actividad de diseño de software se refiere al establecimiento de las estructuras de datos y la arquitectura general del software, de manera que se satisfagan los requerimientos del software. El proceso de diseño traduce los requisitos en una representación de software (Pressman, 2010).

2.6.1. Patrones de diseño utilizados en la solución

Los patrones de diseño figuran una estructura recurrente de componentes que se comunican entre sí para resolver un problema general de diseño que se repite en sistemas orientados a objetos. En él se describe el problema, la solución, cuándo se aplica la solución y las consecuencias que el uso del mismo pueda traer (Gamma, Vlissides, Johnson y Helms, 1994).

Patrones GoF

Los patrones GoF (por sus siglas en inglés, *Gang of Four*) agrupan los patrones de acuerdo a su propósito: creación, estructura y comportamiento.

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

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

Patrones GRASP

Los patrones GRASP (por sus siglas en inglés, *General Responsibility Assignment Software Patterns*) describen los principios fundamentales de la asignación de responsabilidades a objetos. El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente el software orientado a objetos (Grosso, 2011).

Los patrones GRASP utilizados en la solución propuesta fueron:

Bajo acoplamiento: Plantea que se debe poder reutilizar las funcionalidades de las distintas clases, con un nivel de dependencia mínima. Este patrón se evidencia en todas las aplicaciones web que funcionen sobre Django, pues cada pieza de las aplicaciones tiene un propósito clave, que puede modificarse sin afectar otras piezas. Por ejemplo, se puede cambiar la URL de cierta parte de la aplicación sin tener que afectar la implementación subyacente o se puede modificar el HTML de una página sin tener que tocar el código Python que la renderiza.

Alta cohesión: Asigna responsabilidades de manera tal que la cohesión siga siendo alta, o sea que las funcionalidades de las clases estén altamente relacionadas de forma tal que exista una colaboración entre ellas para compartir el esfuerzo y no caiga todo el peso sobre una única clase. Usar este patrón simplifica el mantenimiento y favorece el bajo acoplamiento. Este patrón se utiliza en todas las clases. Un ejemplo se evidencia en las funcionalidades correspondientes al gestionar evaluaciones del cuadro, en este caso se emplean diversas clases para representar cada una de las funcionalidades, `EvaluacionesCreateView()` y `EvaluacionesUpdateView()` por mencionar algunas.

Controlador: Permite manejar todos los eventos del sistema, al servir de intermediario entre las interfaces y el algoritmo que las implementa. Este patrón se evidencia en `cuadro.views`, cuando se listan los datos personales de los cuadros a través de la función `ListarDatosPersonales(request)`. Dicha función recibe la solicitud enviada por el usuario, consulta la base de datos y renderiza la respuesta en la plantilla correspondiente.

2.6.2. Arquitectura de software

La arquitectura de software es la estructura de un sistema o bien la forma en que va a estar organizado este; incluye los elementos que lo conforman, sus propiedades visibles desde el exterior y las relaciones que existen entre ellos (Pressman, 2010).

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

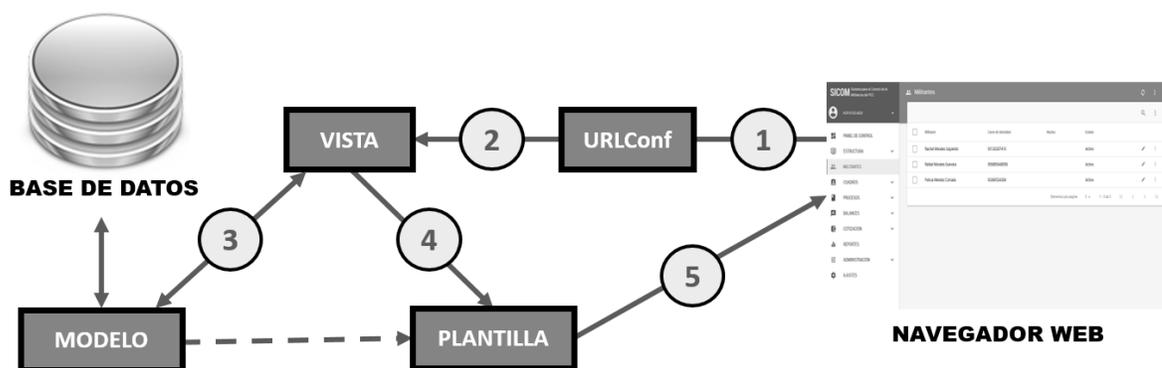


Figura 3. Funcionamiento del MTV de Django. (Infante, 2012)

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

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

Plantilla: Define la interfaz de las páginas web, o sea, cómo se van a mostrar los datos al usuario. También posee algunas etiquetas propias del framework Django.

URLConf: Posee todas las configuraciones de las URLs del sistema, a través de este archivo se puede controlar el despliegue de las vistas, acción conocida como mapeo de URLs.

2.6.3. Diagrama de clases del diseño

A diferencia del modelo de dominio, un diagrama de clases de diseño muestra definiciones de entidades software más que conceptos del mundo real (Larman, 2003). A pesar de que la metodología empleada en la presente investigación no requiere la realización del diagrama de clases del diseño, se muestra este para un mejor entendimiento de la relación existente entre las historias de usuario.

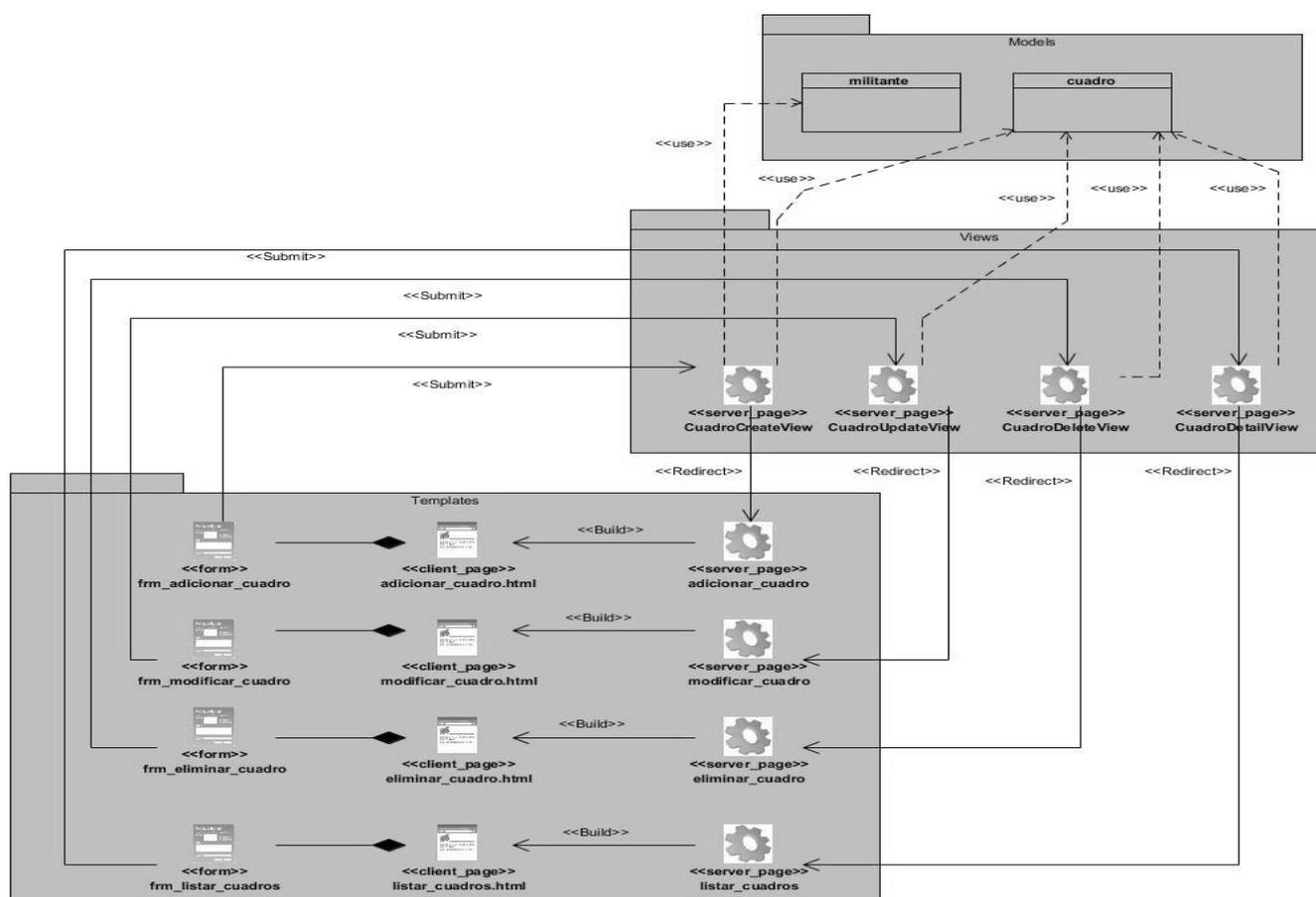


Figura 4. Diagrama de clases del diseño. Elaboración propia.

2.6.4. Modelo de datos

Un modelo de datos es una definición lógica, independiente y abstracta de los objetos, relaciones y demás que en conjunto constituyen la máquina abstracta con la que interactúan los usuarios. Los objetos nos permiten modelar la estructura de los datos (Date, 2001). A continuación, un fragmento del modelo de datos del sistema propuesto, el modelo en su totalidad se puede encontrar en el Anexo 3.

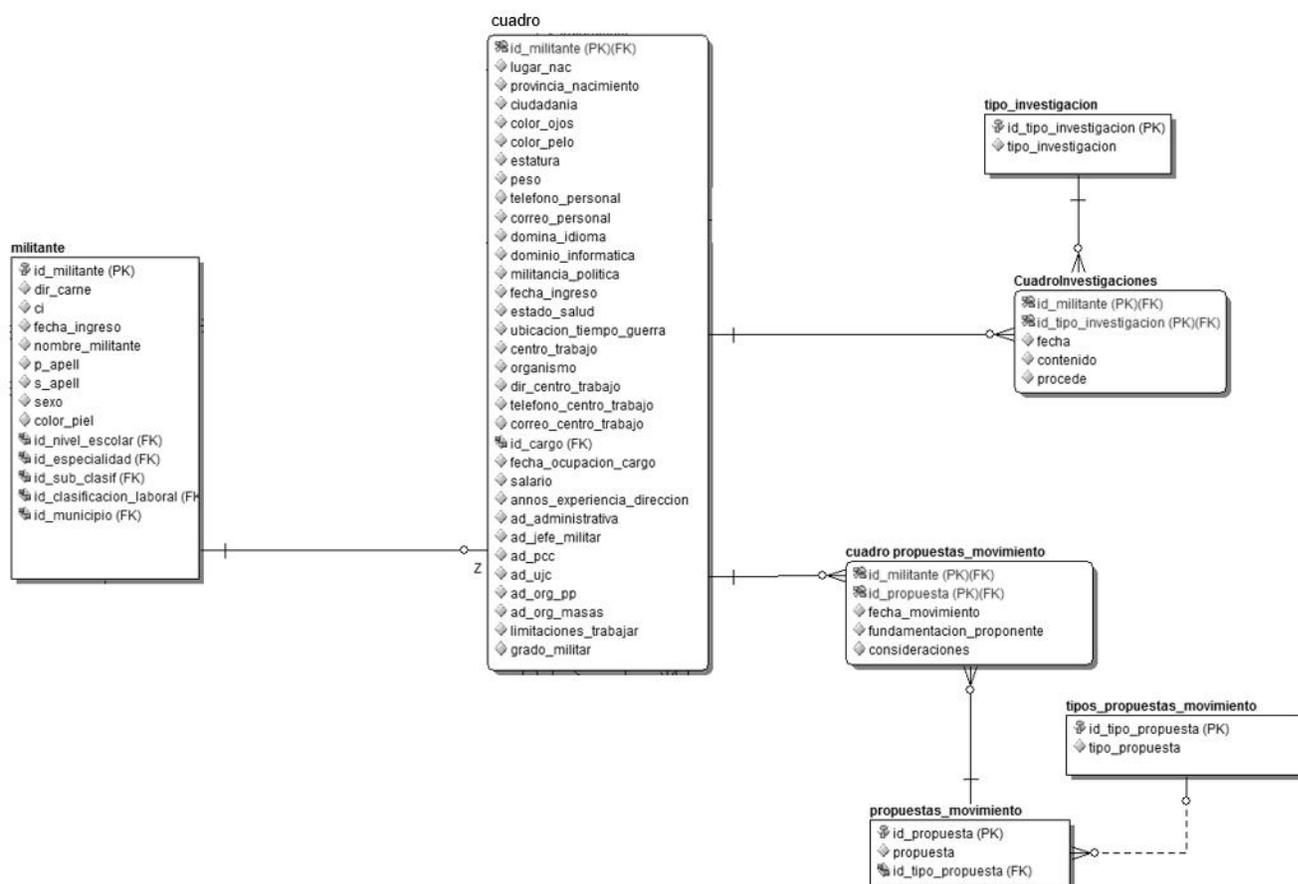


Figura 5. Fragmento del modelo de datos que representa las entidades del módulo “Movimientos e Investigaciones”. Elaboración propia.

Conclusiones del capítulo

Una vez culminadas las actividades de planificación y diseño de la propuesta de solución se obtuvo una visión más clara y detallada del sistema que se desea desarrollar. Los requerimientos funcionales y no funcionales obtenidos a partir del proceso de identificación de requisitos, constituyeron elementos claves en la construcción de la propuesta de solución. La utilización de los patrones de diseño permitió identificar aspectos importantes de la estructura del diseño del sistema web propuesto, lo que garantizó una mayor organización e hizo el código más legible. Los artefactos generados constituyeron no solo la base del

sistema, sino que también permitieron una mejor comprensión del mismo. Además, adoptar la arquitectura de software Modelo-Vista-Plantilla propuesta por el framework de desarrollo utilizado, abonó una propicia organización del sistema a implementar.

Capítulo 3: Implementación y prueba de la propuesta de solución

Introducción

En el presente capítulo se muestran los artefactos correspondientes a las etapas de implementación y prueba del sistema, así como los estándares de codificación que debe seguir el equipo de desarrollo para implementar el software. De acuerdo a la metodología que se utiliza, se especifican, de los tipos de pruebas que esta plantea, el que será empleado para validar el sistema. También se hará uso de pruebas de integración con el objetivo de que los resultados sean los esperados y que el sistema funcione correctamente.

3.1 Diagrama de componentes

El diagrama de componentes tiene como propósito mostrar las relaciones estructurales entre los componentes de un sistema, sus dependencias y localización. Las dependencias en este tipo de diagrama indican que un componente utiliza los servicios ofrecidos por otro componente. Los componentes son considerados unidades autónomas encapsuladas dentro de un sistema que pueden ser librerías, ejecutables, binarios y códigos fuente (Bell, 2004).

3.1.1 Descripción del diagrama de componentes

Plantilla: Paquete que agrupa a las plantillas encargadas de presentar la información al usuario.

Vista: Paquete que agrupa a todos los componentes que interactúan con el paquete de clases Modelo; estos componentes permiten trabajar con algunas utilidades sobre los formularios y la renderización de la información en las plantillas apropiadas.

Modelo: Paquete que agrupa las clases que representan el dominio de entidades de la base de datos y que permiten la interacción directa con el paquete Vista.

Django ORM: Componente encargado de mapear, crear, acceder y modificar la información que se encuentra en la base de datos.

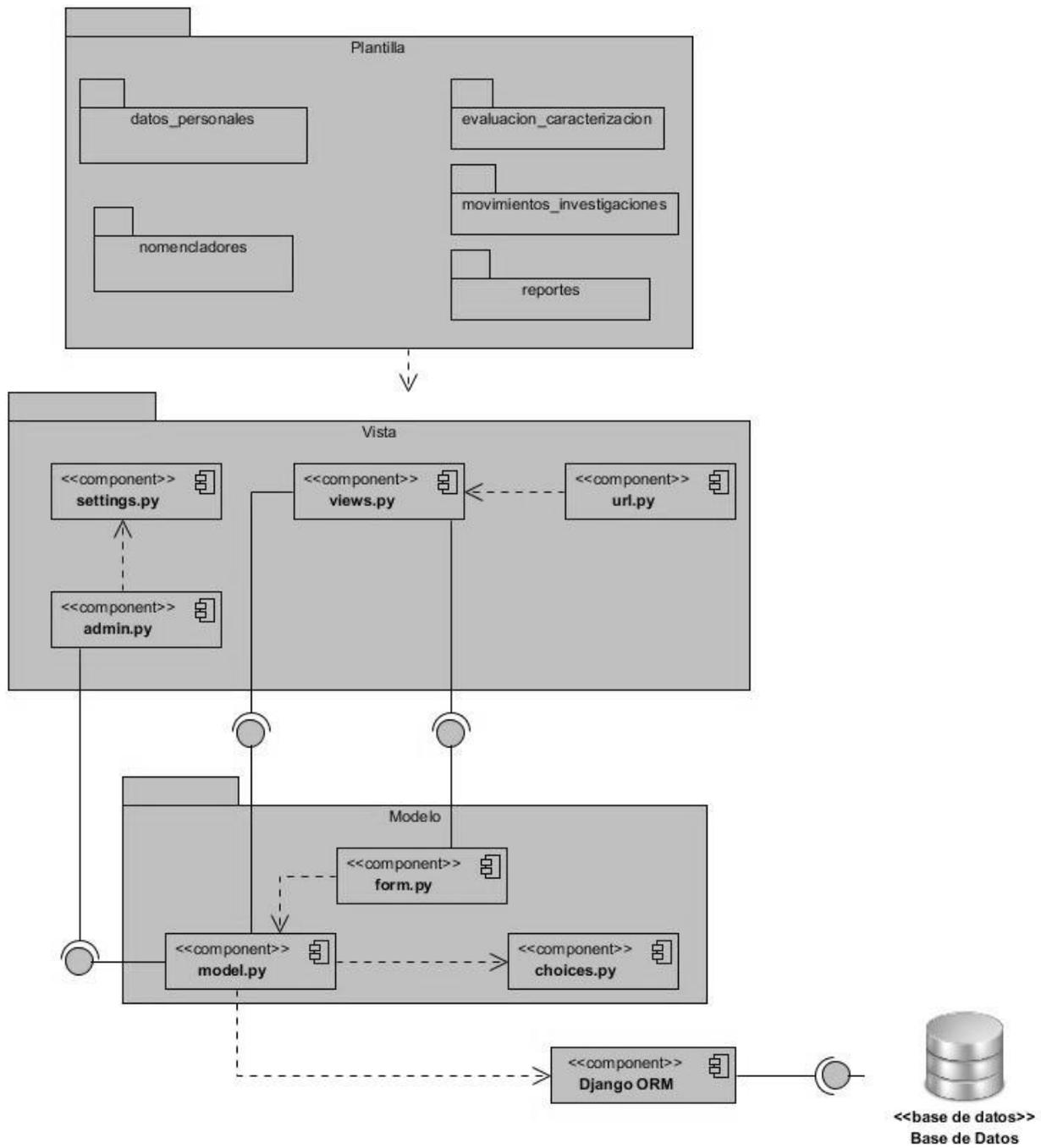


Figura 6. Diagrama de componentes. Elaboración propia.

3.2 Diagrama de despliegue

Un diagrama de despliegue modela la arquitectura de un sistema en tiempo de ejecución; muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos (sparxsystems, 2013). Las relaciones que existen entre nodos representan los protocolos de comunicación que se utilizan para acceder a cada uno.

A continuación, se muestra el diagrama de despliegue definido para la solución propuesta que cuenta con 3 nodos principales. El nodo PC Cliente que requiere de un navegador web, el nodo Servidor Web donde debe estar instalado el Sistema web para la gestión de la información del Departamento de Cuadros y el nodo SGBD en el cual debe estar instalado el Sistema Gestor de Bases de Datos PostgreSQL con la base de datos de Infoest. El nodo PC Cliente representa las estaciones de trabajo de los usuarios que se conectan al sistema, las mismas realizan peticiones al Servidor Web mediante el protocolo HTTPS. El nodo Servidor Web, a su vez, estará conectado al nodo SGBD mediante el protocolo TCP/IP.

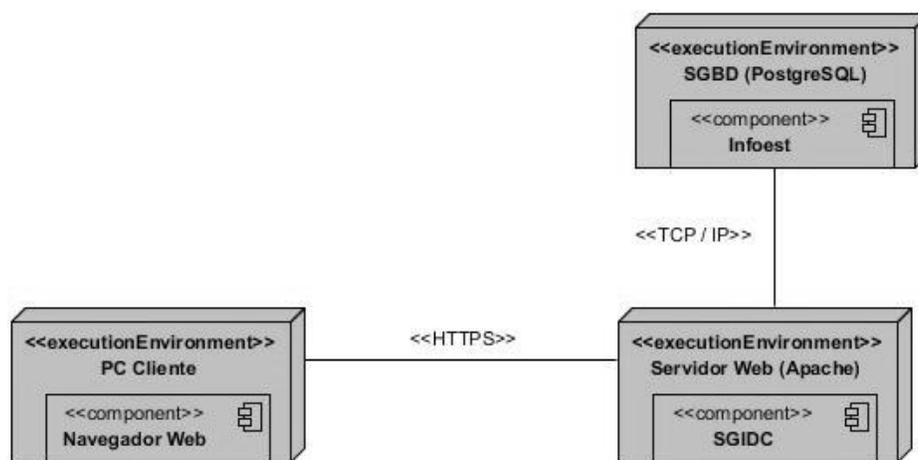


Figura 7. Diagrama de despliegue. Elaboración propia.

3.3 Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación del código fuente de un software. Es preciso definir una serie de pautas que lleven por objetivo uniformar la estructura del código de manera tal que este sea lo más legible posible, como si un único programador hubiese escrito todo el

código de una sola vez (Microsoft, 2015). Las pautas fundamentales definidas para la implementación son las siguientes:

- El tamaño máximo de las líneas de código debe ser de cien a ciento veinte caracteres aproximadamente, de manera tal que se garantice la completa visibilidad de las líneas de código sin necesidad de realizar desplazamiento horizontal.

```
class TrayectoriaMilitar(models.Model):
    organismo = models.CharField(max_length=20, choices=ORGANOS_MILITARES, verbose_name="Órgano militar")
    anno_ingreso = models.IntegerField(max_length=4, choices=YEAR_CHOICES,
                                      default=datetime.datetime.now().year, verbose_name="Año ingreso")
    anno_baja = models.IntegerField(max_length=4, choices=YEAR_CHOICES,
                                   default=datetime.datetime.now().year, verbose_name="Año baja")
    causa_baja = models.ForeignKey(CausaBajaMilitar, verbose_name="Causa de baja", related_name='causa_baja')
    cuadro = models.ForeignKey(Cuadros, verbose_name="Cuadro")
```

Figura 8. Fragmento de código fuente de “views.py”. Elaboración propia.

- Los nombres de las clases y las funciones adoptarán la notación UpperCamelCase y no se utilizará el guion bajo como delimitador entre palabras.

```
class EvaluacionesCreateView(SuccessMessageMixin, CreateView):
    template_name = 'evaluacion_caracterizacion/evaluaciones/adicionar_evaluacion.html'
    form_class = EvaluacionForm
    model = Evaluacion
```

Figura 9. Fragmento de código fuente de “views.py”. Elaboración propia.

- Las sentencias de importación no tienen líneas en blanco entre ellas.

```
from django.core import validators, validators
from django.core.validators import MaxValueValidator
from django.db import models
from django.template.defaultfilters import capfirst
from django.utils.text import get_text_list
from procesos.models import Militante
from cuadro.choices import *
from nomencladores.models import *
from phonenumber_field.modelfields import PhoneNumberField
```

Figura 10. Fragmento de código fuente de “models.py”. Elaboración propia.

- Los nombres de los atributos, variables y parámetros tendrán todas las letras en minúsculas y usarán el guion bajo como delimitador entre palabras.

```
class CuadroInvestigaciones(models.Model):
    cuadro = models.ForeignKey(Cuadros, verbose_name="Cuadro", null=True)
    tipo_investigacion = models.ForeignKey(TipoInvestigacion, verbose_name="Tipo de investigación")
    fecha = models.DateField(auto_now=False, auto_now_add=False, verbose_name="Fecha del anónimo")
    contenido = models.TextField(verbose_name="Contenido del anónimo")
    procede = models.CharField(max_length=100, choices=SINO, verbose_name="Procede a investigación")

    class Meta:
        verbose_name = "Investigación"
        verbose_name_plural = "Investigaciones"
```

Figura 11. Fragmento de código fuente de “models.py”. Elaboración propia.

3.4 Pruebas de software

Las pruebas de software son un elemento crítico para la garantía de calidad del software, y representan una revisión final de las especificaciones del diseño y de la codificación. Son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa (Pressman, 2002).

La metodología que guía el proceso de desarrollo de la presente investigación propone realizar pruebas internas, de liberación y de aceptación. Sin embargo, para la solución propuesta solo se formalizarán las pruebas internas, puesto que se encargan de verificar el resultado de la implementación al probar cada construcción (Rodríguez, 2015). Las otras pruebas que propone la metodología requieren de una entidad certificadora y una aprobación por parte del cliente, elementos que no se contemplan dentro de los requerimientos del presente trabajo. Además, como la solución propuesta debe integrarse con el sistema de militantes, se hará uso de pruebas de integración.

3.4.1 Estrategias de prueba

Una estrategia de prueba del software integra los métodos de diseño de casos de prueba en una serie bien planteada de pasos que va a converger en la eficaz construcción del software. Dicha estrategia debe ser lo suficientemente flexible como para promover un enfoque personalizado, y al mismo tiempo lo

adecuadamente rígido como para originar una planeación razonable y un seguimiento administrativo del avance del producto (Pressman, 2005).

Pressman plantea las siguientes estrategias de prueba:

- Pruebas de unidad
- Pruebas de integración
- Pruebas de validación
- Pruebas de sistema

A la solución que se propone se le realizará cada una de las pruebas antes definidas.

3.4.1. Pruebas de unidad

Las pruebas de unidad descomponen las funciones del programa en comportamientos comprobables discretos que se pueden probar como unidades individuales. Están destinadas a verificar las unidades más pequeñas del software. Se aplican a las funcionalidades para verificar que los flujos de control y datos están cubiertos y funcionan tal como se espera. El objetivo de estas pruebas es ejecutar un código fuente al llamar directamente a los métodos de una clase pasándole a estos los parámetros adecuados (Microsoft, 2015). La prueba de unidad siempre está orientada a caja blanca.

Laurie (como se citó en Correa y Machado, 2012) plantea que las pruebas de **caja blanca** se centran en los detalles procedimentales del software, por lo que su diseño está en gran medida ligado al código fuente. Para realizar este tipo de prueba, el probador escoge diferentes valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa y asegurarse de que se devuelven los valores de salida adecuados.

El entorno de desarrollo PyCharm da la posibilidad de realizar pruebas de unidad a los métodos implementados. A continuación, se muestran los casos de prueba aplicados a las funcionalidades “Realizar evaluación del cuadro” y “Listar reconocimientos del cuadro” del módulo Evaluaciones y Caracterización

Tabla 10. Caso de prueba unitaria “RealizarEvaluacionesTest”. Elaboración propia.

Caso de prueba unitaria

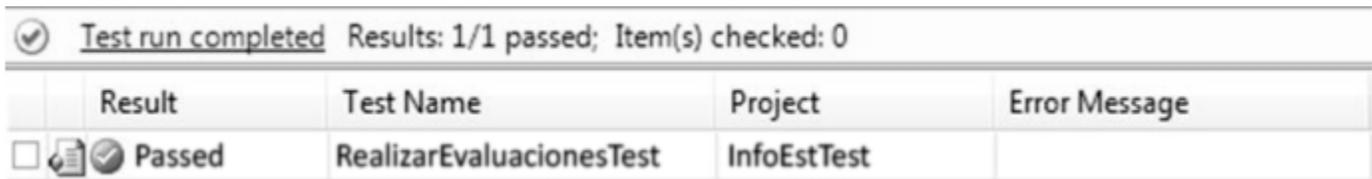
Nombre de la prueba: RealizarEvaluacionesTest

Descripción: Permite comprobar que el elemento pasado como parámetro se guarda satisfactoriamente en la base de datos.

Entradas: *cuadro* cuadro, *date* desde, *date* hasta, *integer* actitud_mantenida, *integer* resultados_desempeno, *integer* capacidad_dir, *integer* influencia_exigencia, *integer* cualidades_personales, *integer* capacidad_intelectual, *string* señalamientos, *string* recomendaciones, *string* conclusión, *string* nombre, *string* cargo, *date* fecha

Criterio de aceptación: Se verifica directamente en la base de datos que el elemento haya sido adicionado satisfactoriamente.

Resultado:



Test run completed Results: 1/1 passed; Item(s) checked: 0

	Result	Test Name	Project	Error Message
<input type="checkbox"/>  	Passed	RealizarEvaluacionesTest	InfoEstTest	

Tabla 11. Caso de prueba unitaria “ListarReconocimientosTest”. Elaboración propia.

Caso de prueba unitaria

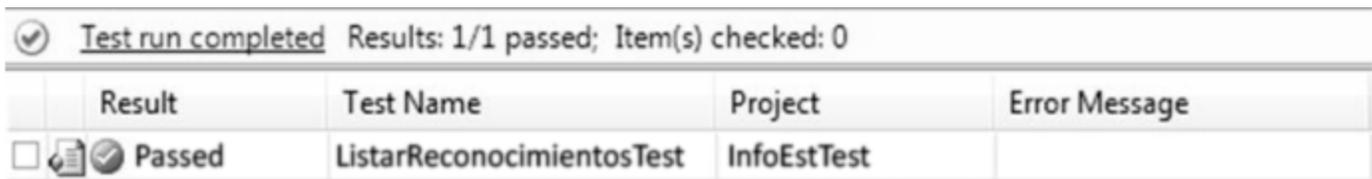
Nombre de la prueba: ListarReconocimientosTest

Descripción: Permite obtener un listado con todos los reconocimientos que tiene un determinado cuadro.

Entradas: *integer* id_cuadro

Criterio de aceptación: El tamaño del listado devuelto es igual al esperado.

Resultado:



Test run completed Results: 1/1 passed; Item(s) checked: 0

	Result	Test Name	Project	Error Message
<input type="checkbox"/>  	Passed	ListarReconocimientosTest	InfoEstTest	

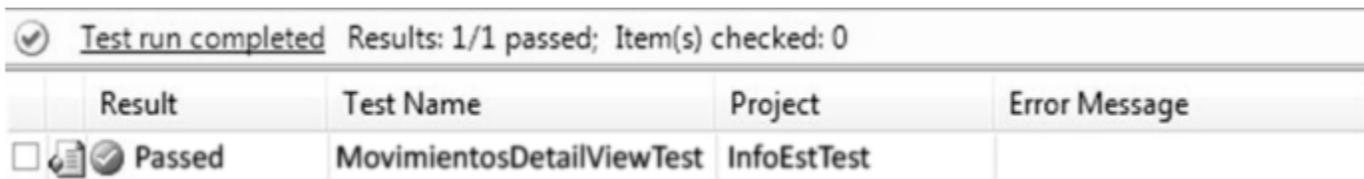
Resultados de las pruebas de unidad

Las pruebas unitarias se aplicaron a funcionalidades de considerable importancia una vez culminada su implementación, por lo que, de la aplicación de estas pruebas no se tiene un control en cuanto a la cantidad de veces que fueron ejecutadas en las iteraciones de implementación, pero se detectaron un total de diez no conformidades, las cuales fueron resueltas en el momento de su detección.

3.4.2. Pruebas de integración

Las pruebas de integración están diseñadas para probar la interacción entre los distintos componentes de un sistema. Su objetivo es tomar los componentes probados en unidad y construir una estructura acorde a lo que propone el diseño (US, 2013). A continuación, se muestra un caso de prueba que responde a la prueba de integración aplicada a la funcionalidad “Generar reportes de los cuadros” del módulo Reportes:

Tabla 12. Caso de prueba de integración “MovimientosDetailViewTest”. Elaboración propia.

Caso de prueba de integración				
Nombre de la prueba: MovimientosDetailViewTest				
Descripción: Permite obtener una plantilla con un conjunto específico de datos del cuadro.				
Entradas: integer id_cuadro				
Criterio de aceptación: Se verifica directamente que la información que contiene la plantilla coincida con los datos almacenados en la base de datos.				
Resultado:				
				

Resultados de las pruebas de integración

Las pruebas de integración se aplicaron a las funcionalidades que de una forma u otra requerían parámetros que eran propios del sistema para el control de militantes del PCC, por lo que estas pruebas se llevaron a

cabo tras cada iteración de implementación. Durante su ejecución se encontraron un total de doce no conformidades, todas fueron resueltas en el momento de su detección. (Ver figura 12)



Figura 12. Resultado de las pruebas de integración. Elaboración propia.

3.4.3. Pruebas de validación

Las pruebas de validación consisten en realizar acciones visibles para el usuario y en la salida que el sistema tiene ante cada una de estas acciones. La validación del software se consigue mediante una serie de pruebas de **caja negra** que demuestran la conformidad con los requisitos funcionales. Las pruebas de caja negra se realizan sobre la interfaz del software, sin tener en cuenta el funcionamiento interno del sistema, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas (Pressman, 2002).

A continuación, los casos de prueba correspondientes a las funcionalidades realizar, modificar, eliminar y mostrar evaluaciones del cuadro:

Tabla 13. Caso de prueba Realizar Evaluación. Elaboración propia.

Nombre HU: Realizar evaluación del cuadro			
Descripción: El sistema permitirá realizar la evaluación correspondiente a un cuadro de acuerdo al valor asignado a los indicadores en el sistema.			
Texto Válido:			
Desde	(válido)	Cualidades personales	(válido)

Hasta	(válido)	Capacidad intelectual	(válido)
Actitud mantenida en el ejercicio de sus funciones	(válido)	Señalamientos	(válido)
Resultados de su desempeño	(válido)	Recomendaciones	(válido)
Capacidad de dirección demostrada	(válido)	Conclusiones de la evaluación	(válido)
Influencia y exigencia hacia los subordinados	(válido)	Nombre y Apellidos	(válido)
		Cargo	(válido)
		Fecha	(válido)

Respuesta del Sistema: El sistema realiza la evaluación del cuadro, la adiciona y muestra una notificación de creación exitosa.

Texto Inválido

Desde	(inválido)	Cualidades personales	(inválido)
Hasta	(inválido)	Capacidad intelectual	(inválido)
Actitud mantenida en el ejercicio de sus funciones	(inválido)	Señalamientos	(inválido)
Resultados de su desempeño	(inválido)	Recomendaciones	(inválido)
Capacidad de dirección demostrada	(inválido)	Conclusiones de la evaluación	(inválido)
Influencia y exigencia hacia los subordinados	(inválido)	Nombre y Apellidos	(inválido)
		Cargo	(inválido)
		Fecha	(inválido)

Respuesta del Sistema: El sistema muestra un mensaje de error informando que se introdujeron datos incorrectos.

Flujo Central: Módulo Cuadros / Opción Evaluaciones y Caracterización / Acción Evaluaciones / Opción Adicionar

Tabla 14. Caso de prueba Modificar Evaluación. Elaboración propia.

Nombre HU: Modificar evaluación del cuadro

Descripción: El sistema permitirá modificar la evaluación correspondiente a un cuadro.

Texto Válido:

Desde	(válido)	Cualidades personales	(válido)
Hasta	(válido)	Capacidad intelectual	(válido)

Actitud mantenida en el ejercicio de sus funciones	(válido)	Señalamientos	(válido)
		Recomendaciones	(válido)
Resultados de su desempeño	(válido)	Conclusiones de la evaluación	(válido)
Capacidad de dirección demostrada	(válido)	Nombre y Apellidos	(válido)
Influencia y exigencia hacia los subordinados	(válido)	Cargo	(válido)
		Fecha	(válido)

Respuesta del Sistema: El sistema actualiza la evaluación del cuadro y muestra una notificación de modificación exitosa.

Texto Inválido

Desde	(inválido)	Cualidades personales	(inválido)
Hasta	(inválido)	Capacidad intelectual	(inválido)
Actitud mantenida en el ejercicio de sus funciones	(inválido)	Señalamientos	(inválido)
		Recomendaciones	(inválido)
Resultados de su desempeño	(inválido)	Conclusiones de la evaluación	(inválido)
Capacidad de dirección demostrada	(inválido)	Nombre y Apellidos	(inválido)
Influencia y exigencia hacia los subordinados	(inválido)	Cargo	(inválido)
		Fecha	(inválido)

Respuesta del Sistema: El sistema muestra un mensaje de error informando que se introdujeron datos incorrectos.

Flujo Central: Módulo Cuadros / Opción Evaluaciones y Caracterización / Acción Evaluaciones / Opción Editar evaluación

Tabla 15. Caso de prueba Eliminar Evaluación. Elaboración propia.

Nombre HU: Eliminar evaluación del cuadro

Descripción: El sistema permitirá eliminar la evaluación correspondiente a un cuadro.

Texto: Válido

Respuesta del Sistema: El sistema elimina la evaluación del cuadro y muestra una notificación de eliminación exitosa.

Flujo Central: Módulo Cuadros / Opción Evaluaciones y Caracterización / Acción Evaluaciones / Opción Eliminar evaluación

Tabla 16. Caso de prueba Mostrar evaluaciones del cuadro. Elaboración propia.

Nombre HU: Mostrar evaluaciones del cuadro

Descripción: El sistema permitirá mostrar un listado con todas las evaluaciones del cuadro.

Texto: Válido

Respuesta del Sistema: El sistema lista todas las evaluaciones del cuadro registradas en el mismo.

Flujo Central: Módulo Cuadros / Opción Evaluaciones y Caracterización / Acción Evaluaciones

Tabla 17. Descripción de las variables de entrada para los casos de prueba Realizar y Modificar evaluación. Elaboración propia.

#	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Desde	Campo de fecha	No	Puede ser cualquiera de las opciones que brinda el widget de la fecha, pero con las restricciones de que debe ser menor que el valor del campo hasta y menor o igual que el valor de la fecha actual.
2	Hasta	Campo de fecha	No	Puede ser cualquiera de las opciones que brinda el widget de la fecha, pero con las restricciones de que debe ser mayor que el valor del campo desde y menor o igual que el valor de la fecha actual.

3	Actitud mantenida en el ejercicio de sus funciones	Campo de número	No	Puede ser cualquier número entero positivo menor que 11
4	Resultados de su desempeño	Campo de número	No	Puede ser cualquier número entero positivo menor que 11
5	Capacidad de dirección demostrada	Campo de número	No	Puede ser cualquier número entero positivo menor que 11
6	Influencia y exigencia hacia los subordinados	Campo de número	No	Puede ser cualquier número entero positivo menor que 11
7	Cualidades personales	Campo de número	No	Puede ser cualquier número entero positivo menor que 11
8	Capacidad intelectual	Campo de número	No	Puede ser cualquier número entero positivo menor que 11
9	Señalamientos	Campo de texto	Si	Puede ser cualquier combinación de caracteres alfanuméricos incluyendo el espacio
10	Recomendaciones	Campo de texto	Si	Puede ser cualquier combinación de caracteres alfanuméricos incluyendo el espacio
11	Conclusiones de la evaluación	Campo de selección	No	Puede ser cualquiera de las opciones que muestra el campo de seleccionar
12	Nombre y Apellidos	Campo de texto	No	Puede ser cualquier combinación de caracteres alfabéticos incluyendo el espacio
13	Cargo	Campo de texto	No	Puede ser cualquiera de las opciones disponibles que carga por defecto el sistema

14	Fecha	Campo de fecha	No	Puede ser cualquiera de las opciones que brinda el widget de la fecha., pero con las restricciones de que debe ser mayor o igual que el valor del campo hasta y menor o igual que el valor de la fecha actual.
----	-------	----------------	----	--

Resultados de las pruebas de validación

Las pruebas de validación se ejecutaron tras concluir cada una de las cuatro iteraciones de implementación, de manera general en la primera iteración se detectaron veinte no conformidades, agrupadas en no conformidades de código, interfaz y ortografía, de ellas fueron solucionadas dieciséis. En la segunda, se detectaron ocho no conformidades, incluyendo las cuatro no resueltas en la iteración anterior, aquí se resolvieron cinco de ellas. En la tercera iteración se encontraron cuatro no conformidades, incluidas las tres no resueltas en la segunda iteración de pruebas, fueron solucionadas en esta instancia tres de ellas. Para una cuarta iteración, solo se encontró la no conformidad que quedó pendiente en la iteración anterior, la misma se solucionó satisfactoriamente. (Ver figura 13).

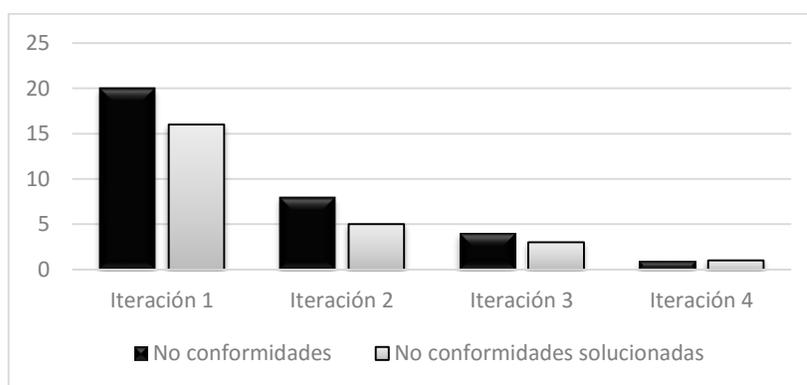


Figura 13. Resultado de las pruebas de validación. Elaboración propia.

3.4.4. Pruebas de sistema

Las pruebas de sistema engloban una serie de pruebas cuyo propósito fundamental es ejercitar el sistema de cómputo. Cada una de estas pruebas tiene un objetivo distinto, pero todas trabajan para verificar que se hayan integrado adecuadamente todos los elementos del sistema y que realizan las funciones correctas (Pressman, 2005). El tipo de prueba de sistema aplicada a la solución fue la prueba de rendimiento. Existen diversos tipos de pruebas de rendimiento, entre las más importantes, según Correa y Machado (2012), se encuentran:

Pruebas de carga: Es el tipo de prueba de rendimiento más sencillo. Las pruebas de carga generalmente se realizan para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperadas. Esta carga puede ser el número esperado de usuarios que utilizan el sistema de manera concurrente.

Pruebas de estrés: Este tipo de prueba se realiza para determinar la solidez del sistema en los momentos de carga extrema, permite a los administradores determinar si el sistema rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

Resultados de las pruebas de sistema

Para llevar a cabo estas pruebas, se creó un entorno con las condiciones mínimas necesarias para la ejecución del sistema. Dichas pruebas fueron ejecutadas sobre las funcionalidades que mayor demanda pueden tener durante el funcionamiento del mismo. La Figura 14 muestra los resultados obtenidos al realizar esta prueba a la funcionalidad “Realizar evaluación del cuadro”.

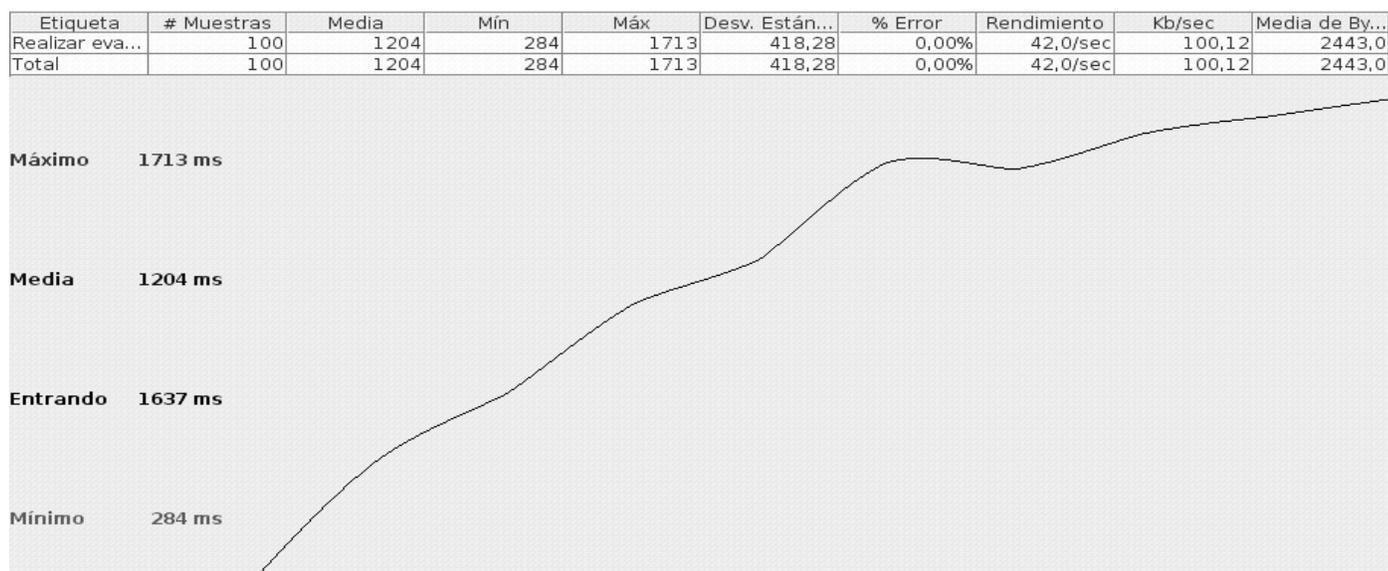


Figura 14. Resultado de las pruebas de rendimiento. Elaboración propia.

Conclusiones del capítulo

A lo largo del proceso de implementación el uso de los estándares de codificación definidos contribuyó a obtener una adecuada uniformidad y legibilidad en el código fuente. El diagrama de componentes dio la posibilidad de ver las dependencias entre los componentes del sistema, además reflejó, de manera más detallada, la ubicación física de los componentes de acuerdo a la arquitectura empleada. La aplicación de pruebas de unidad, integración, validación y rendimiento a la solución desarrollada permitió encontrar errores que afectaban el funcionamiento del sistema, lo que posibilitó corregirlos a tiempo para que el mismo cumpliera totalmente con los requisitos funcionales definidos en la etapa de análisis. Las pruebas realizadas fueron presentadas al cliente como evidencia, lo cual compone la prueba final de que el subsistema fue aceptado por los mismos al catalogar los resultados de forma positiva para el cumplimiento de sus necesidades y peticiones formuladas.

Conclusiones

La investigación desarrollada y los resultados obtenidos permiten arribar a las siguientes conclusiones:

- El estudio de los principales aspectos relacionados con los sistemas de gestión de información permitió sentar las bases para el posterior desarrollo del sistema web.
- A partir del análisis y diseño del sistema web se obtuvieron los artefactos necesarios para guiar el proceso de desarrollo del mismo.
- Como resultado de la implementación se obtuvo el Sistema web para la gestión de la información del Departamento de Cuadros del CCPCC, que cumple con los 150 requerimientos funcionales identificados en la fase de ejecución.
- Las pruebas internas realizadas permitieron probar el correcto funcionamiento del sistema y los resultados satisfactorios de las mismas otorgaron validez a la presente investigación, siendo esto la base para que el sistema pueda favorecer los procesos que tienen lugar en el Departamento de Cuadros del Comité Central del Partido Comunista de Cuba.

Recomendaciones

Se recomienda a los interesados en mejorar o continuar esta investigación:

- Incorporar funcionalidades que permitan la personalización de los reportes, pues las implementadas en esta investigación solo generan reportes específicos.

Bibliografía

- Aja, L. (2002, 10 de julio). *Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones*. Acimed. Recuperado de http://bvs.sld.cu/revistas/aci/vol10_5_02/aci04502.htm
- Arias, Y., Blanco, R. y Bagarotti, Y. (2013). Metodología para el desarrollo e implantación de sistemas de información geográfica. *Serie Científica de la Universidad de las Ciencias Informáticas*. 6(7).
- Arjonilla, D., Sixto, J., Medina, G. y José, A. (2002). *La Gestión de los sistemas de información en la empresa*. Madrid, España: Ediciones Pirámides.
- Beck, Kent y Andres, Cynthia. 2004. *Extreme Programming Explained: Embrace Change 2nd Edition*. Estados Unidos: Addison-Wesley.
- Bell, D. (2004). *UML basics: The component diagram*. Recuperado de: <https://www.ibm.com/developerworks/rational/library/dec04/bell/>
- Bertino, E. y Martino, L. (1993). *Object-Oriented Database System: Concepts and Architectures*. Massachusetts, Estados Unidos: Addison-Wesley.
- Bracho, D. (30 de octubre de 2011). *Beneficios de los Sistemas de Información*. Recuperado de <http://es.slideshare.net/sylem123/beneficios-de-los-sistemas-de-informacion>.
- Canós, J., Letelier, P. y Panadés, M. (2006). *Metodologías Ágiles en el Desarrollo de Software*. Recuperado de: <http://ima.udg.edu/Docencia/07-08/3105200728/TodoAgil.pdf>
- CCM. (2015). *Lenguajes de programación*. Recuperado de <http://es.ccm.net/contents/304-lenguajes-de-programacion>
- CEA. (2010). *Sistemas de Información Geográfica*. España. <http://sig.cea.es/SIG>
- Consejo de Estado. (2011). Decreto Ley 281. Del sistema de información del gobierno. *Gaceta Oficial de la República*, CIX (10), 29-33.
- Correa, J. & Machado, Y. *Subsistema de aprovisionamiento de usuarios para el Sistema de Administración de Identidades* (Tesis de pregrado). Universidad de las Ciencias Informáticas, Cuba
- Danysoft. (2006). *Modelado de bases de datos*. Recuperado de <http://www.danysoft.com/free/model2.pdf>
- Date, C. (2001). *Introducción a los Sistemas de bases de datos*. México: Pearson Educación de México

- Dayley, B. (2006). *Python Phrasebook: Essential Code and Commands*. Recuperado de <http://shodan.me/books/Programming/Python/Python%20Phrasebook%20-%20Essential%20Code%20and%20Commands%20%282006%29.pdf>
- Diamon, J. (2006). *Armas, gérmenes y acero: breve historia de la humanidad en los últimos trece mil años*. Colombia: Random House Mondadori.
- Días, Z. & González, A. (2009). *Sistema Gestor de Trabajos de Diploma* (Tesis de pregrado). Universidad de las Ciencias Informáticas, Habana, Cuba.
- Duque, A. (28 de febrero de 2014). *METODOLOGÍAS DE DESARROLLO AUP*. Recuperado de <https://prezi.com/0w76rs1poj dq/metodologias-de-desarrollo-aup/>
- Empresamia. (2013). *¿Qué es un servidor web?* <http://empresamia.com/crear-empresa/crear/item/644-que-es-un-servidor-web>
- Galiano, G. (2015). *ESTUDIO COMPARATIVO DE SERVIDORES MULTIMEDIA* (Tesis de pregrado). Universidad Nacional de Educación a Distancia, Alicante.
- Gamma, E., Vlissides, J., Johnson, R. y RICHARD HELM. (1994). *Design Patterns: elements of Reusable Object-Oriented Software*. Pearson Education. ISBN 0321700694, 9780321700698. Recuperado de: <https://msdn.microsoft.com/es-es/library/bb972242.aspx>
- García, Y. y Pérez, Y. (2009). *Aplicación de BPM en las Etapas de Iniciación del PSI y Concepción del Proceso de Desarrollo de Software del DDS* (Tesis de pregrado). Universidad de las Ciencias Informáticas, Cuba.
- Gonzalez, C., & Hug, C. (2011). *ConML• Conceptual Modelling Language Technical Specification• version 1.1*.
- Gorita, I. y Kuyama, S. (2002). *Gestión de la Información en las Organizaciones del Sistema de las Naciones Unidas: Sistemas de Información para la Gestión*. https://www.unju.org/en/reports-notes/Documents/JIU_REP_2002_9_Spanish.pdf
- Grosso, A. (2011). *Patrones GRASP. Prácticas de software*. Recuperado de: <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>
- Guerra, Y. & Matos, E. (2013). *Sistema para la verificación de personas en línea a través de huellas dactilares*. (Tesis de pregrado). Universidad de las Ciencias Informáticas, Habana, Cuba.
- Guevara, E. (1970). *Obras. 1957-1967*. La Habana: Casa de las Américas.

- Hamlett, K. & Media, D. (2015). *The Features of an Information Management System*. Houston Chronicle. Recuperado de http://smallbusiness.chron.com/features-information-management-system-2114.html?__hstc=51389238.a4e81186687d88b1757551c9c909fea4.1447992550188.1447992550188.1448420049898.2&__hssc=51389238.1.1448420049898&__hsfp=388920196
- Infante, S. (2012). *Curso Django para perfeccionistas con deadlines*. Recuperado de https://www.academia.edu/9510572/maestrosdelweb_curso_django
- INTECO. (2009). *Ingeniería del Software: Metodologías y ciclos de vida*.
- JetBrains. (2015). *The Most Intelligent Python IDE*. <https://www.jetbrains.com/pycharm/?fromMenu>
- Kabir, M. (2002). *La biblia de Servidor Apache 2*. Recuperado de <https://yexia.files.wordpress.com/2010/09/mohammed-j-kabir-la-biblia-del-servidor-apache-21.pdf>
- Larman, C. (2003). *UML y patrones*. Segunda Edición. Recuperado de <http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf>
- Letelier, P. y Penadés, C. (2006). *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. C&TA, 05 (26).
- López, R., Martínez, F., & López, M. (2009). *Diseño de un sistema de información web para la gestión cultural* (Tesis de pregrado). Facultad de Comunicación y Documentación de la Universidad de Murcia, Campus Universitario de Espinardo. Murcia.
- Medina, J. (2005). *Evaluación del impacto de los sistemas de información en el desempeño individual del usuario: aplicación en instituciones universitarias* (Tesis doctoral). Escuela Técnica Superior de Ingenieros de Telecomunicación, Madrid.
- Mendoza, I. (2011). *Definición de un Framework para aplicaciones Web con navegación sensible a concerns* (Tesis de maestría). Universidad Nacional de La Plata, Argentina.
- Microsoft. (2015). *Revisiones de código y estándares de codificación*. España. <https://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>
- Moreno, E. (2011). *Herramienta para el Servicio de Información Administrativa del Gobierno de Cantabria* (Tesis de pregrado). Universidad Carlos III de Madrid. Madrid.
- Noda, Y. y Campanioni, J. (2015). *Capa de Servicios Web para el Sistema Integrado de Gestión Estadística (SIGE)* (Tesis de pregrado). Universidad de las Ciencias Informáticas, Cuba.

- Núñez, J. (2010). *Usabilidad en metodologías Ágiles* (Tesis de maestría). Universidad Politécnica de Madrid. Facultad de Informática, España.
- PCC. (1996). *Código de Ética de los Cuadros del Estado Cubano*. Recuperado de http://www.tsp.cu/sites/default/files/codigo_etica_cuadros_estado_cuba.pdf
- _____. (2011). *Historia del Partido Comunista de Cuba*. Recuperado de http://www.pcc.cu/i_historia.php
- _____. (28 y 29 de enero de 2012). *Primera Conferencia Nacional del Partido Comunista de Cuba*. Granma. Recuperado de <http://www.granma.cu/granmad/secciones/1ra-conferencia-pcc/objetivos.html>
- Peralta, C. & Durán, D. (2015). *Módulos de edición de plantillas y recepción de órdenes de impresión para el Sistema de Personalización de Documentos de Identidad basado en tecnologías libres* (Tesis de pregrado). Universidad de las Ciencias Informáticas, Cuba.
- Postgres-es. (2010). http://postgresql.org.es/sobre_postgresql
- PostgreSQL. (2014). http://www.postgresql.org/about/press/presskit94/es/#original_release
- Pressman, R. (2002). *Ingeniería de Software: Un Enfoque Práctico*. Quinta Edición. Nueva York, Estados Unidos: McGraw-Hill
- _____. (2005). *Ingeniería de Software: Un Enfoque Práctico*. Sexta Edición. Nueva York, Estados Unidos: McGraw-Hill / Interamericana de México
- _____. (2010). *Software Engineering: A practitioner's Approach*. Seventh Edition. New York, Estados Unidos: McGraw-Hill
- Roblejo, I. (2012). *Procedimiento para la validación de los requisitos no funcionales en los Sistemas de Gestión de Información* (Tesis de pregrado). Universidad de las Ciencias Informáticas, Cuba.
- Rodríguez, T. (2015). *Metodología de desarrollo para la Actividad productiva de la UCI*. <http://excriba.prod.uci.cu/page/context/shared/document-details?nodeRef=workspace://SpacesStore/a622adab-eac5-4fb3-ba08-a266767fff5f>
- Rojas, E. A. (1978). *Tesis y Resoluciones. Primer Congreso del Partido Comunista de Cuba*. Ciudad de la Habana, Cuba: Editorial Ciencias Sociales.
- Rossum, G. (2009). *El tutorial de Python*. <http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf>
- Rumbaugh, J., Jacobson, I., & Booch, G. (2004). *Unified Modeling Language Reference Manual*, The. Pearson Higher Education.

- Saavedra García, María L. y Tapia Sánchez, Blanca. (2013). *El uso de las tecnologías de información y comunicación TIC en las micro, pequeñas y medianas empresas (MIPyME) industriales mexicanas*. *Enl@ce: Revista Venezolana de Información, Tecnología y Conocimiento* 10(1), enero-abril, 85-104.
- Sánchez, L. (2015). *PIMCD2013-python. Publicación 1.0*. Universidad de Complutense de Madrid. España.
- Scrum.org. (2015). *What is Scrum?* Scrum.org. <https://www.scrum.org/resources/what-is-scrum>
- ServiciosTIC. (2015). *Las T.I.C.* <http://www.serviciostic.com/las-tic.html>
- Sommerville, I. (2007). *Software Engineering*. Recuperado de: http://eva.uci.cu/mod/resource/view.php?id=9269&subdir=/Sommerville_8va_edicion
- SparxSystems. (2013). *Diagrama de Despliegue UML2*. Sparx Systems. Recuperado de: http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html
- Torres, L. (mar.-abr., 2015). La gestión de información y la gestión del conocimiento. *SciELO*, 19 (2).
- Trigas, M. (2012). *Metodología Scrum. Desarrollo detallado de la fase de aprobación de un proyecto informático mediante el uso de metodologías ágiles*. <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>
- UCI. (2010). *Proyecto Estratégico*. Universidad de las Ciencias Informáticas.
- US. (2013). *Pruebas de integración*. Universidad de Sevilla.
- Vidal, M. y Araña, A. (jul.-sep. 2012). Gestión de la información y el conocimiento. *SciELO*, 26(3).