

Universidad de las Ciencias Informáticas

Facultad 1



Título: “Implementación de los Submódulos de reporte y proyección de movimiento de los cuadros de la Unión de Jóvenes Comunistas Nacional”.

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

Autores: Richard Rivero Pérez

Reinald Cabrera Llanes

Tutores: Ing. Catherine Muñoz Velázquez

Ing. Mairelys Boeras Velázquez

Ciudad de La Habana, Cuba

“Año 52 de la Revolución”



Si los jóvenes fallan, todo fallará
Fidel Castro Ruz

- ✚ A Dios por estar siempre conmigo, por ser bueno y misericordioso.
- ✚ Existe un conjunto de personas que son claves en mi vida y sin ellos este trabajo no sería lo que es. Mi madre, Odalys Pérez Castellero mamita del alma. Mi padre Lázaro García Giménez que aunque ya no viva conmigo lo quiero mucho y para mí siempre será el mismo. Mis hermanos Gerardo García, Meilin y Merlin Rivero. Mis tíos Odaisa Pérez y Javier Suarez. Mis abuelos Isabel y Avelino a los cuales debo mucho. A todos ellos mi más sincero agradecimiento. Agradezco también a Ada Margarita Zequeiras y su esposo Rafael Marzo por ser tan especiales.
- ✚ A mi chiquitica Sarah Aristegui Zequeiras quien es la dueña de mi corazón, por su amor y comprensión, su apoyo y dedicación.
- ✚ A mis hermanos Villalobos, los inseparables Vitico y Raidel
- ✚ Agradezco a mis compañeros de universidad, también mis amigos, los cuales han estado a mi lado en buenos y malos momentos. Entre ellos están mi queridísima Yula, la mulata del sabor, Daily Núñez Fernández, Rolando Bairon Santisteban uno de los 7, Carlos Pérez Mateu, Reinald Cabrera, Dariel el niche, Ángel Erllys Boloy el bolo, Dannier el delmillito, Lester Borrego, Orlando Castillo, Yasin Hernández y todos con los que he compartido aula o apartamento en estos 5 años. Agradezco también a los que no están, A Ramón el neirin, a Rubén el peludo, Yunier Peña mi compañero de trabajo y tantos otros que no están en este minuto en mi cerebro pero siempre estarán en mi corazón. Dios los bendiga y espero nos podamos ver con cierta frecuencia.

Agradecimientos

- ✚ Al piquete del churre, mis hermanitas y sus amistades por tan buenos momentos que pasamos juntos.
- ✚ Agradezco a mi compañero de tesis, el cual ya mencioné, por su paciencia y comprensión.
- ✚ A mis hermanos de la iglesia por su apoyo y oraciones.
- ✚ A mi tutora, por su gran trabajo y dedicación.
- ✚ A Julio Isaza mi primer profesor de programación y todos los que me han impartido clases.
- ✚ A mis compañeros del secretariado de la FEU, Yidian, los 2 Carlos, José Augusto, Yosbany y todos los demás.
- ✚ A esta universidad por servir de escenario para mi desarrollo en todas las esferas de la vida.
- ✚ A la Revolución y sus líderes por llevar a cabo un sistema como el nuestro y por supuesto una universidad como esta.

Richard.

- ✚ A mi mamá, la mujer de mi de vida, la que ha sido capaz de educarme y convertirme en el hombre que hoy soy, la que llora cuando está mucho tiempo sin verme, la que ha hecho tantas cosas por mí que necesitaría una vida entera para escribirlo; te quiero con todas las fuerzas de mi corazón pupa.
- ✚ A mi papá mi ídolo como hombre y como padre el mejor de todos, nunca me había gustado parecerme tato a nadie pero me encanta ser idéntico a ti como me dicen todos que lo soy a ti también te quiero con todo mi corazón y lo sabes aunque seamos a veces guajiros cerraos y no nos digamos que nos queremos.
- ✚ A mi hermana la niña de mis ojos, no sé qué sería si no tuviera una hermana como tú con tus malcriadeces y todo eso pero no importa sabes que te quiero como a nadie y que voy a estar aquí siempre para apoyarte, te quiero.
- ✚ A mi mami, gracias mamita por aguantar todas mis malcriadeces, por ayudarme tato sin pedirme nada a cambio, por quererme como sé que me quieres, por enderezarme cuando más lo necesitaba, te quiero mucho mi mamita.
- ✚ A toda mi familia, mis abuelos Papito, Cori, Ricardo, Nidia, mi tío Robe que nunca he recibido un no de su parte, mis tías Nillita y TiTi (la china), Omar, Tata mi hermano aunque no lo seamos, mis primos Tolo, Daiselis, Yuderkis, el loco de Raudel, los quiero mucho a todos, gracias por estar ahí cuando los necesito.
- ✚ A el Yuro (Abelitin) gracias mi hermano por ser un gran amigo y de los buenos de los que no quedan, a Pachi otro que no se queda detrás gracias por prestarme la bicicleta sabiendo que soy un desastre mi hermano, a Calero no

podía faltar uno que nunca pero nunca ha puesto un pero para nada que yo quisiera hacer, al Chumi primo gracias por aguantar todas las jodederas mías, gracias a todos mis socios del barrio los llevo presente siempre.

✚ A Lester, El Niche, Richard, Boloy a Rolo, Orlando, Yasin, Aliusca, Yula gracias por estar ahí cuando muchos decían que yo no me iba graduar y hablaban tato de mi, gracias también a esas personas que me sirvieron de motivación para seguir adelante, gracias a Rolo y a Vi por incluirme en el comedor obrero y cocinar para todos, muchas gracias los voy a extrañar muchísimo.

✚ A nuestra tutora por su gran ayuda y por aguantar todas las molestias causadas.

Rey.

A mi familia por haber confiado en mí, a mi chicuela del alma y a todos mis amigos.

Richard

A mis padres Mirian y Ricardo por hacer de mí quien soy, a mi hermana por todo su amor, a toda mi familia por creer en mí, a mi novia por ser fuente de inspiración, por brindarme su confianza y por quererme tanto. A ustedes que hicieron realidad mis sueños.

Rey.

DECLARACIÓN DE AUTORÍA

Se declara que Richard Rivero Pérez y Reinald Cabrera LLanes son los únicos autores de la investigación titulada: **“Implementación de los Submódulos de reporte y proyección de movimiento de los cuadros de la Unión de Jóvenes Comunistas Nacional”** y se autoriza a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmamos la presente a los **4** días del mes de **junio** del año **2010**.

Reinald Cabrera LLanes

Richard Rivero Pérez

Firma del Autor

Firma del Autor

Catherine Muñoz Velázquez

Firma del Tutor

La Unión de Jóvenes Comunistas (UJC) es la organización política de la juventud cubana. Es completamente independiente, el ingreso a la misma tiene carácter selectivo y voluntario, está estructurada a lo largo de todo el país bajo una Política de Cuadros¹. Entre sus objetivos principales, tiene el de asegurar la unidad de los jóvenes cubanos, movilizarlos en torno a la Revolución Socialista y contribuir a su educación.

Actualmente la UJC Nacional gestiona la información relacionada con sus cuadros de forma manual, provocándole problemas de almacenamiento y pérdida de la misma debido al deterioro de los documentos. No posee una centralización de la información, por lo que la recopilación y actualización de un conjunto de datos es realizado a través de llamadas telefónicas o realizando viajes a las provincias y municipios del país; situación que trae como consecuencia que estos no se tengan en tiempo y con la calidad requerida, gastos económicos en combustible, en pasajes para viajar y en llamadas telefónicas.

En la presente investigación se realiza la Implementación de los Submódulos de reporte y proyección de movimiento de los cuadros de la UJC Nacional, para lograr garantizar la seguridad, la proyección de movimiento y la generación de reporte de información referente a los cuadros de esta organización, en esta se utilizó como metodología de desarrollo a RUP, como lenguaje de programación JAVA, UML como lenguaje de modelado, NetBeans como entorno de desarrollo, como herramienta CASE a Visual Paradigm, como generador de reportes a DynamicJasper y como gestor de base de datos a Access.

Palabras claves

Cuadros, Submódulos, gestión de la información, Unión de Jóvenes Comunistas.

¹ Política de Cuadros: es el grupo de acciones que permiten preparar a una o varias personas para que ante la sustitución o liberación de un cargo, el proceso no se detenga. Es la forma en que se educa y desarrolla a los miembros de las organizaciones, como dirigentes revolucionarios. Es la estructura que permite garantizar una acertada dirección del proceso revolucionario bajo cualquier circunstancia.

Índice

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	1
1.1 Introducción	1
1.2 Estado del Arte	1
1.2.1 Sistemas de reportes a nivel internacional	2
1.2.2 Sistemas de reportes a nivel nacional	4
1.2.3 Sistemas de reportes en la Universidad de las Ciencias Informáticas (UCI)	6
1.2.4 Ideas a utilizar	7
1.3 Metodologías y herramientas de desarrollo de software	8
1.3.1 Metodología	8
1.3.2 Lenguaje de Modelado	11
1.3.3 Herramienta CASE	12
1.4 Herramientas y Tecnologías para el Desarrollo de la Aplicación	13
1.4.1 Herramientas para la generación de reportes	13
1.4.2 Lenguaje de programación	16
1.4.3 Entorno de Desarrollo	17
1.4.4 Sistema Gestor de base de datos	18
1.5 Conclusiones	21
CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA	22
2.1 Introducción	22
2.2 Valoración crítica del diseño propuesto por el analista.	22
2.2.3 Cambios de gestor de base de datos	22
2.2.4 Estructura de la base de datos	23
2.3 Análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser rehusados. Estrategias de integración.	37
2.4 Paradigma de programación	38
2.4.1 Características asociadas a la POO	38
2.5 Requisitos funcionales	39

2.6 Requisitos no funcionales	45
2.7 Diagramas de casos de uso del sistema. (Ver descripción en el Anexo 17)	47
2.8 Diagrama de Componentes	50
2.9 Diagrama de despliegue	51
2.10 Conclusiones.....	51
CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA	52
3.1 Introducción.....	52
3.2 Automatización de pruebas	52
3.3 Descripción Preliminar del Modelo de Pruebas	52
3.4 Pruebas de caja negra.....	53
3.4.1 Partición equivalente	53
3.6 Aplicación de pruebas de caja negra	54
3.7 Conclusiones	69
Conclusiones Generales.....	70
Recomendaciones	71
Glosario de Términos.....	72
Bibliografía	73
Referencias Bibliográficas	75

Índice de Tabla

Tabla 1: Operacionalización de variables-----	4
Tabla 2.1: Descripción Vivienda -----	23
Tabla 2.2: Descripción Reserva Cuadro -----	23
Tabla 2.3: Descripción Datos Interés -----	24
Tabla 2.4: Descripción Núcleo Familiar -----	24
Tabla 2.5: Estudios Militante-----	24
Tabla 2.6: Descripción Movilizaciones -----	25
Tabla 2.7: Descripción Servicio Militar Activo-----	25
Tabla 2.8: Descripción Dirección Particular -----	25
Tabla 2.9: Descripción Sanción Admin Jurídicas -----	26
Tabla 2.10: Descripción Trayectoria Estudiantil-----	26
Tabla 2.11: Descripción Miembro Comité -----	26
Tabla 2.12: Descripción Trayectoria Militar-----	27
Tabla 2.13: Descripción Familiar Extranjero -----	27
Tabla 2.14: Descripción Estado Salud -----	27
Tabla 2.15: Descripción del nomenclador Sector -----	28
Tabla 2.16: Descripción del nomenclador Rama-----	28
Tabla 2.17: Descripción del nomenclador Países -----	28
Tabla 2.18: Descripción del nomenclador Motivo Visita-----	29
Tabla 2.19: Descripción del nomenclador Tipo Mérito-----	29
Tabla 2.20: Descripción del nomenclador Año -----	29
Tabla 2.21: Descripción del nomenclador Estado Vivienda-----	29
Tabla 2.22: Descripción del nomenclador Tipo Militancia-----	30
Tabla 2.23: Descripción del nomenclador Parentesco-----	30
Tabla 2.24: Descripción del nomenclador Provincia -----	30
Tabla 2.25: Descripción del nomenclador Municipio-----	30
Tabla 2.26: Descripción del nomenclador Cargo Actual-----	31
Tabla 2.27: Descripción del nomenclador Tipos Estudios -----	31

Tabla 2.28: Descripción del nomenclador Categoría Militante-----	31
Tabla 2.29: Descripción del nomenclador Escuelas Políticas-----	31
Tabla 2.30: Descripción del nomenclador Tipo Estructura Superior -----	32
Tabla 2.31: Descripción del nomenclador Tipo Causa -----	32
Tabla 2.32: Descripción del nomenclador Sub Tipo Causa -----	32
Tabla 2.33: Descripción del nomenclador Nombre Organización-----	33
Tabla 2.34: Descripción del nomenclador Estados Militantes -----	33
Tabla 2.35: Descripción del nomenclador Nivel Cultural -----	33
Tabla 2.36: Descripción del nomenclador Tipo Miembro -----	33
Tabla 2.37: Descripción del nomenclador Composición Étnica -----	34
Tabla 2.38: Descripción del nomenclador Cargo DB -----	34
Tabla 2.39: Descripción del nomenclador Grado Militar-----	34
Tabla 2.40: Descripción del nomenclador Clasificador Ocupacional Cuadro -----	35
Tabla 2.41: Descripción del nomenclador Clasificador Ocupacional Reserva -----	35
Tabla 2.42: Descripción del nomenclador Labor -----	35
Tabla 2.43: Descripción del nomenclador Vía -----	36
Tabla 2.44: Descripción del nomenclador Motivo-----	36
Tabla 2.45: Descripción del nomenclador Estado -----	36
Tabla 2.46: Descripción del nomenclador Procedencia Laboral -----	37
Tabla 2.47: Descripción del nomenclador Sanción Cuadro -----	37
Tabla 2.50: Descripción del Diagrama de Componentes -----	51
Tabla 3.1: Descripción del Caso de Prueba: Caso de Uso Autenticar Usuario -----	55
Tabla 3.2: Descripción del Caso de Prueba: Caso de Uso Cambiar Contraseña -----	55
Tabla 3.3: Descripción del Caso de Prueba: Caso de Uso Buscar Usuario -----	56
Tabla 3.4: Descripción del Caso de Prueba: Caso de Uso Gestionar Usuario -----	58
Tabla 3.5: Descripción del Caso de Prueba: Caso de Uso Asignar Cargo -----	59
Tabla 3.6: Descripción del Caso de Prueba: Caso de Uso Gestionar Estructura -----	61
Tabla 3.7: Descripción del Caso de Prueba: Caso de Uso Gestionar Cargo-----	64
Tabla 3.8: Descripción del Caso de Prueba: Caso de Uso Realizar Movimiento -----	65

Tabla 3.9: Descripción del Caso de Prueba: Caso de Uso Gestionar Reserva-----	67
Tabla 3.10: Descripción del Caso de Prueba: Caso de Uso Gestionar Planificación de Movimiento -----	68

Índice de Figuras

Figura 1: Estructura de RUP-----	8
Figura 2: Diagrama de Casos de Uso Módulo Administración-----	47
Figura 3: Diagrama de Casos de Uso Módulo Proyección de Movimientos-----	48
Figura 4: Diagrama de Casos de uso. Modulo Reporte -----	49
Figura 5: Diagrama de Componentes de los Submódulos de reporte y proyección de movimiento de la UJC Nacional -----	50
Figura 6: Diagrama de Despliegue. Submódulos de reportes y proyección de movimientos de la UJC Nacional-----	51
Figura 8: Prueba de caja negra -----	53

INTRODUCCIÓN

La Unión de Jóvenes Comunistas (UJC) se constituyó el 4 de abril de 1962. Esta organización surge como resultado del proceso de unidad del movimiento juvenil cubano, de la victoria definitiva del socialismo tras los combates de Playa Girón y el desarrollo hacia formas superiores de organización de la Asociación de Jóvenes Rebeldes, su predecesora. La UJC fue creada con el fin de asegurar la unidad de los jóvenes cubanos, movilizarlos en torno a la Revolución Socialista y contribuir a su educación política. Estos objetivos deben ser alcanzados no sólo entre su membrecía, sino también con toda la juventud, a la cual representa y vela por sus intereses.

La UJC es cantera y reserva del Partido Comunista, pues tiene entre sus principales responsabilidades la formación ideológica en sus filas de los futuros militantes de la Organización partidista, además de contribuir a la educación de las nuevas generaciones. Trabaja con las organizaciones de masas juveniles, y con los niños, a través de la Unión de Pioneros de Cuba, para contribuir a la educación de las nuevas generaciones como constructores conscientes del socialismo esta protagoniza y dirige los más de 150 Programas de la Revolución que contribuyen a la formación de las nuevas generaciones y a una mayor educación de la sociedad.

Cuba actualmente, está inmersa en un proceso de informatización de la sociedad, con el objetivo de integrarse plenamente a la infraestructura global de la información, para avanzar en el plano social, hacer más eficiente la economía, y elevar la calidad de vida del pueblo por esto son innumerables las organizaciones e instituciones que hoy se integran a la aplicación de las Tecnologías de la Informática y las Comunicaciones (TIC).

La UJC es un ejemplo de las organizaciones que llevan a cabo esta integración. El ingreso a la misma es de carácter voluntario y selectivo. Su objetivo principal está en la continua preparación cultural, política e ideológica de los jóvenes cubanos, en función de seguir consolidando las conquistas de la Revolución, así como la formación integral de las nuevas generaciones. La organización está estructurada a lo largo de todo el país a través de una Política de Cuadros y presenta una estructura nacional compuesta por 14 Comités Provinciales, 169 Comités Municipales, incluyendo el Comité Municipal de la Isla de la Juventud, así como 8 Comités Distritales, de estos últimos 4 se ubican en el municipio Camagüey y 4 en el municipio Santiago de Cuba. Adicionalmente la Organización por su complejidad cuenta con 2 Comités Especiales: El Comité Especial UJC y el Comité Especial del Exterior. La UJC cuenta con Direcciones Juveniles en las

Fuerzas Armadas Revolucionarias (FAR) y el Ministerio del Interior (MININT) que se subordinan a las Direcciones Políticas del PCC de ambos mandos, y que tienen distribuidos los militantes a lo largo y ancho del país.

Actualmente la UJC Nacional no cuenta con un sistema informático para gestionar la información relacionada con sus cuadros, presentando grandes problemas en cuanto a almacenamiento y pérdida de la información, debido al deterioro de los documentos, no existe una centralización de la información de manera que se pueda acceder a ella en todo momento y desde cualquier parte, lo que puede incidir notablemente en demoras a la hora de reunir información. La recopilación y actualización de un conjunto de datos en ocasiones es realizado por teléfono y puede tomar varios días. La otra forma de realizar este proceso es a través de viajes a las provincias y municipios del país, influyendo en que no se tenga la información en tiempo y con la calidad requerida, trayendo como consecuencia gastos económicos en combustible para la transportación, gastos monetarios en pasajes y llamadas telefónicas. Por lo antes planteado se decidió el desarrollo de una aplicación que permita la obtención de la información y todo lo referente a la proyección de movimiento de los cuadros de la UJC Nacional y así mantenerla de forma organizada, posibilitando un mejor manejo de la misma.

Por consiguiente, el **problema científico** queda formulado de la siguiente forma: ¿Cómo facilitar la obtención de la información y la proyección de movimientos de los cuadros de la UJC Nacional?

Para resolver el problema planteado se propone como **objetivo general**: Implementar los Submódulos de reporte y proyección de movimiento de los cuadros de la UJC Nacional.

De ahí se derivan los siguientes **objetivos específicos**:

- ❖ Realizar análisis crítico del diseño de los Submódulos de reporte y proyección de movimiento.
- ❖ Diseñar casos pruebas.

El **objeto de estudio** de este trabajo son los procesos de obtención de la información y proyección de movimiento de los cuadros de la UJC Nacional. El **campo de acción** queda enmarcado en los Submódulos de reporte y proyección de movimiento de los cuadros de la UJC Nacional.

Para lograr el cumplimiento de los objetivos trazados se proponen las siguientes **tareas científicas**:

- Confección del marco teórico-conceptual de la investigación partiendo de una búsqueda y revisión bibliográfica.

- Profundización de los componentes que integran un sistema de gestión de información.
- Estudio de los aspectos teóricos sobre diferentes plataformas de gestión existentes en el mundo.
- Selección de las características específicas del sistema para su fácil empleo
- Estudio de herramientas para la automatización de las pruebas.
- Estudio de los artefactos entregados por los analistas de los submódulos.
- Estudio de las librerías que faciliten el trabajo con reportes de tablas.
- Implementación de los Submódulos de reporte y proyección de movimiento de los cuadros de la UJC Nacional.

Se plantea como **hipótesis**: Si se desarrollan los Submódulos de reporte y proyección de movimiento de los cuadros de la UJC Nacional entonces se logrará un mejor control y manejo de la información de los mismos.

Dada la hipótesis anterior se plantean las siguientes **variables de la investigación**:

Operacionalización de las variables:

Variable independiente: Submódulos de reporte y proyección de movimiento.

Variable dependiente: Control y manejo de la información

Variables	Dimensiones	Indicadores	Sub- Indicadores	Unidad de Medida
Submódulos de reporte y proyección de movimiento.	Apariencia o interfaz externa	Utilización	Simple	1
			Complejo	2
	Rendimiento	Tiempo de respuesta	Mínimo	1
			Medio	2
			Alto	3

	Seguridad	Confidencialidad	Alta	3
			Media	2
			Baja	1
		Integridad	Alta	3
			Media	2
			Baja	1
		Disponibilidad	Alta	3
			Media	2
			Baja	1
Control y manejo de la información.	Rendimiento	Complejidad	Alta	3
			Media	2
			Baja	1
		Organización de la información.	Buena	2
			Mala	1

Tabla 1: Operacionalización de variables

Terminado el desarrollo de este sistema se tendrá como **posibles resultados:**

Un software que posibilite la administración del sistema, la proyección de movimiento de los cuadros de la UJC Nacional y la obtención de reportes de información de acuerdo con un criterio previamente seleccionado por el usuario.

Para llevar a cabo las tareas planteadas se emplearon métodos teóricos y empíricos de la investigación científica. Los **métodos teóricos** utilizados para cumplir con las tareas a desarrollar son:

↳ **Análisis Histórico-Lógico:** Permite estudiar de forma razonada la trayectoria histórica real de los fenómenos, su evolución y desarrollo. Este método permitió realizar un estudio de los sistemas de gestión de información relacionados con los reportes y proyección de movimiento de los cuadros de la UJC Nacional, así como de las tecnologías que existen actualmente para decidir cuáles emplear en el desarrollo de la aplicación.

↳ **Analítico-Sintético:** Son dos procesos inherentes al pensamiento, operaciones lógicas importantes; que permiten, buscar la esencia de los fenómenos, los rasgos que los caracterizan y los distinguen. Su objetivo en una investigación es analizar las teorías, documentos, permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio. Este método permitió realizar el estudio de documentos relacionados con los cuadros de la UJC Nacional, extraer lo esencial de cada uno de ellos y partir del análisis de toda la información obtenida organizarla y sintetizarla para elaborar una estructura adecuada.

Los métodos empíricos utilizados son:

Modelación: Hizo posible que se estudiara la realidad mediante diversos modelos y diagramas que ayudaron a comprender el proceso en su totalidad. Ejemplos estos son los diagramas de casos de usos, los diagramas de clases del análisis, el modelo de diseño.

Observación: Se realizó una exhaustiva observación para ver que sucedía en realidad con los procesos que realiza la UJC Nacional con respecto a todos sus cuadros llegando así a un mejor entendimiento de los mismos.

Estructura del Documento

Capítulo1: En este capítulo se explican conceptos fundamentales que ayudan a comprender mejor el trabajo. Se hace un estudio de sistemas similares en el ámbito nacional e internacional de lo que se quiere implementar, así como, de las características de las principales herramientas, tecnologías y metodología a utilizar en el desarrollo de los submódulos.

Capítulo2: En este capítulo se realiza la crítica del análisis y diseño propuesto con anterioridad, se analizan posibles implementaciones, componentes o módulos ya existentes que puedan ser rehusados, se definen requisitos funcionales y no funcionales que se tienen en cuenta para el desarrollo de la aplicación.

Capítulo3: En este capítulo se realiza la validación del sistema mediante las pruebas de caja negra. Se genera la documentación respectiva de las mismas dándose una pequeña explicación de sus características y se evalúan los resultados finales de las pruebas y los resultados obtenidos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

La utilización de los sistemas de reportes en cualquier lugar del mundo forma parte de la evolución lógica en los diferentes sistemas informáticos. Muchos de estos sistemas implementan generadores de reportes, mientras que otros son los encargados de la obtención de reportes específicos en dependencia de la organización a la que pertenecen y de las necesidades específicas y el interés del cliente. Le permiten al usuario obtener información de manera rápida y organizada de acuerdo con la información almacenada en las bases de datos y en muchas ocasiones sirven como herramientas para la toma de decisiones de acuerdo con el análisis de la información que se haya obtenido de los datos que se almacenen en las bases de datos.

En el capítulo se realiza un estudio de diferentes sistemas informáticos que se encargan de la obtención de reportes tanto a nivel internacional como nacional incluyendo a la Universidad de las Ciencias Informáticas. Además, se realiza la fundamentación del uso de las herramientas y lenguajes de programación que serán utilizados para el desarrollo del sistema.

1.2 Estado del Arte

Sistema informático: es el conjunto de partes interrelacionadas, hardware, software y de Recurso Humano. Un sistema informático típico emplea una computadora que usa dispositivos programables para capturar, almacenar y procesar datos.

Sistemas de Gestión: es una estructura probada para la gestión y mejora constante de las políticas, los procedimientos y procesos de la organización. Un sistema de gestión ayuda a lograr los objetivos de la organización mediante una serie de estrategias, que incluyen la optimización de procesos, el enfoque centrado en la gestión y el pensamiento disciplinado.

La implementación de un sistema de gestión eficaz puede ayudar a:

- Mejorar la efectividad operativa.
- Reducir costos.
- Aumentar la satisfacción de clientes y partes interesadas.
- Lograr mejoras continuas.

- Potenciar la innovación.
- Aportar claridad.

El uso de un sistema de gestión probado permite renovar constantemente el objetivo, las estrategias, las operaciones y niveles de servicio.

Sistemas de información: están concebidos para capturar, actualizar, integrar, consultar y analizar información pertinente a la organización a la cual pertenecen.

Un sistema de información es la composición de elementos que operan unidos en orden de capturar, procesar, almacenar y distribuir información. Esta información se utiliza generalmente para tomar decisiones, la coordinación, el control y el análisis en una organización. Frecuentemente, el propósito básico del sistema es la gestión de la información.

Los sistemas de información:

- Gestionan los datos que almacenan.
- Gestionan los usuarios que utilizan la información, que además pueden agregarla.
- Poseen una interfaz en correspondencia con la información que contienen.
- Integran no sólo datos e información, sino también programas y otros sistemas de información.

1.2.1 Sistemas de reportes a nivel internacional

Nefrosoft

Es una aplicación informática para la gestión de una Unidad de Hemodiálisis. Realizada en el sistema operativo Microsoft Windows, utilizando la base de datos Microsoft Access. Compatible con Windows 95 / 98 / Me / NT4 / 2000 / XP. Permite la gestión clínica de un centro de hemodiálisis totalmente personalizable por el usuario, contiene ayuda de contexto, posibilidad de exportación de datos y además permite la generación de informes en formato .doc, envío por fax y correo electrónico.

Nefrosoft permite realizar el registro de los datos clínicos generados en la práctica médica diaria, de forma fácil y rápida. Permite trabajar con historias clínicas, datos administrativos, y todos los datos de la evolución clínica: analíticas, exploraciones, tratamientos, informes y listados. (1)

Sistema de Análisis Financiero y Rentable (SAFYR).

SAFYR es una aplicación que tiene como objetivo principal la administración de portafolios de inversiones de títulos de valores. Es una aplicación muy amigable, ágil, segura y lo más importante es su facilidad de manejo y operación. La interfaz gráfica y la automatización que se diseñó para la captura de datos hacen posible todas estas ventajas. Está desarrollado en Microsoft Visual Studio .Net ASP.NET (Páginas Activas de Servidor .NET), con motor de bases de datos en Microsoft SQL Server utilizando una arquitectura cliente servidor. Cumple con todos los requerimientos de trabajo en un entorno multiusuario, esto significa que varios usuarios desde diferentes máquinas o equipos pueden crear, modificar, ver y administrar datos simultáneamente sin generar ningún tipo de conflicto.

SAFYR cuenta con un módulo de reportes el cual se divide en las siguientes categorías:

- Listado de Cancelaciones.
- Listado del PYG.
- Listado de Valoración de Portafolio.
- Saldos por Clientes.
- Saldos de Clientes por Oficina.
- Impresión de Recibos de Caja.
- Impresión de Cheques.
- Listado de Transacciones Pasivas (Reporte de Aperturas, Adiciones, Retiros y Cancelaciones).
- Reporte de los Cierres Diarios.
- Extractos de Clientes.

Cada una de las categorías descritas anteriormente a su vez está compuesta por una variedad de informes o reportes. Destacar que el SAFYR incluye algunos tipos de reportes dentro de otros módulos, por lo que se considera que los reportes presentes en este sistema, no se encuentran centralizados. (2)

Software Integrado de Gestión Académica Web (SIGA)

Es un sistema modular para la administración académica y curricular, diseñado especialmente para instituciones de educación superior, que integra tanto datos como procesos en una solución completa eliminando barreras de espacio y tiempo.

SIGA es el sistema ideal para instituciones de educación superior que quieren posicionarse y re-organizar el manejo de la información para optimizar y mejorar los recursos de flujo de información al interior de la entidad.

Está compuesto por subsistemas que permiten la operación de los diferentes procesos académicos y estos presentan a su vez un Módulo de Listados y Reportes de información de acuerdo con el subsistema seleccionado dentro de los cuales se encuentran.

- Matrícula Académica.
- Registro Académico.
- Gestión Profesoral.
- Programación Académica.
- Gestión de Egresados.

1.2.2 Sistemas de reportes a nivel nacional

Cuba no se ha quedado apartada del resto del mundo respecto a los sistemas de reportes, por lo que ha desarrollado software que incluyen dentro de sus funcionalidades la obtención de reportes de información.

Sistema para la Gestión Académica (GESTACAD).

Sistema desarrollado en la Universidad de Matanzas “Camilo Cienfuegos”, se encarga fundamentalmente del procesamiento de la información académica, facilitando a los distintos usuarios obtener y actualizar información.

Capítulo 1: Fundamentación Teórica

Presenta seis módulos:

- Módulo de Administración: permite la gestión de las tablas del sistema vía Web así como agregar nuevas consultas al sitio oficial y establecer los distintos niveles de acceso a estas.
- Módulo Web para las Secretarías Docentes: son las encargadas de la Gestión de Estudiante, permite hasta el momento la realización de acciones generales comunes en una Secretaría Docente así como la obtención de reportes oficiales.
- Módulo Web para los Jefes de Departamentos docentes: incluye acciones relativas como la asignación de la carga docente y el control sobre los profesores del departamento.
- Módulo Web para la Gestión de la Matrícula.
- Módulo Web para los Profesores: estos pueden llevar el control docente de sus estudiantes, el control de las evaluaciones así como reportes relativos a su carga docente.
- Sitio Web: con reportes en línea con la utilidad del registro docente para los profesores además de la búsqueda de estudiantes la cual devuelve, además de algunos datos personales del estudiante, su ubicación según el horario docente detallando aula, asignatura y tipo de clases que está recibiendo además de su estado si se ha pasado asistencia en el turno de clase.

Este módulo brinda reportes como son:

- Reportes dinámicos de información: da la posibilidad al usuario de seleccionar los campos de datos que desea obtener en el reporte así como el título de este y las condiciones que debe cumplir la información a mostrar.
- Reporte de notas por asignatura y grupo(examen final, extraordinario, especial, premio)
- Reporte de los resultados académicos de un estudiante en toda su carrera.

SAICRE

Sistema Automatizado Informativo para la Comunidad de Residencia Estudiantil de la Universidad de Pinar del Río (UPR) surge ante la necesidad de gestionar la información del becado de la comunidad referida y de sus indicadores para lograr su control y obtener con rapidez consultas y reportes necesitados, gestión que se realizaba de forma manual. Este producto informático resalta no sólo por los

beneficios que trae consigo su uso al humanizar el trabajo y ahorrar recursos, sino también, por las cuestiones novedosas que presenta en sus procesos de actualización, de reportes, así como las herramientas brindadas para garantizar su seguridad. El sistema registra la información de los becados y sus indicadores, a partir de estas funcionalidades se obtienen reportes por criterios según los intereses del usuario.

1.2.3 Sistemas de reportes en la Universidad de las Ciencias Informáticas (UCI)

Sistema de Gestión Académico (Akademos).

Sistema de Gestión Académico que se utiliza en la UCI cuenta con siete módulos:

- Matrícula: permite la obtención de estudiantes y estructura.
- Estudiante: permite a los estudiantes estar informados sobre su desempeño docente.
- Registro: permite llevar el control del desarrollo docente, notas y asistencia.
- Profesores: permite gestionar todo lo referente a los profesores del claustro.
- Plan de Estudio: permite la definición de los planes de estudio, asignatura, disciplina y perfiles.
- Reporte: permite el diseño y generación de reportes referentes a la información contenida en las hojas de matrícula y prematrícula.
- Seguridad: todo lo referente a la asignación de permisos.

Permite exportar los reportes a diferentes formatos e imprimirlos, permitiendo su configuración. Contiene un asistente para realizar reportes de evaluaciones ya sean de tipo resumen o listado. Permite enviar reportes por correo electrónico, así como la configuración de estos para que se envíen un día determinado. El administrador del módulo puede configurar permisos según el rol que tenga el usuario.

Sistema de Corte, Sistema de Asistencia y Sistema de Evaluaciones

Estos sistemas son creados por la dirección de Informatización de la UCI en el año 2003-2004 con el objetivo de garantizar la gestión académica de la universidad y satisfacer las necesidades de información de sus usuarios.

Los Sistemas del Corte y de Asistencia permiten brindar reportes que pueden ser consultados a nivel de facultad, año, grupo, provincia, municipio, asignatura, sexo, raza, vía de ingreso y procedencia.

Capítulo 1: Fundamentación Teórica

El Sistema del Corte muestra la cantidad de estudiantes que hay para cada uno de los tipos de evaluaciones (Bien, Regular y Mal), y a partir de este reporte se puede obtener el listado de estos estudiantes con sus evaluaciones por asignaturas y su evaluación final.

El Sistema de Asistencia cuenta con estos mismos reportes, sólo que en lugar de las evaluaciones por asignaturas, muestra la cantidad de ausencias por asignaturas.

Por su parte el Sistema de Evaluaciones pone a disposición del usuario los siguientes reportes:

- Cantidad de estudiantes sin mundial por facultad.
- Cantidad de estudiantes sin mundial por provincia.
- Listado de estudiantes por grupo y facultad que no tienen evaluaciones insertadas en las asignaturas obligatorias.
- Listado de estudiantes con 5 en todas las asignaturas.
- Listado de estudiantes con 4 y/o 5 en todas las asignaturas.
- Listado de estudiantes con 1 asignatura suspensa.
- Listado de estudiantes con 2 asignaturas suspensas.
- Listado de estudiantes con 3 o más asignaturas suspensas.

En el caso de los tres últimos reportes también se muestran cuáles son las asignaturas que tienen suspensas.

1.2.4 Ideas a utilizar

Después de realizado el estudio de los diferentes sistemas informáticos existentes que incluyen módulos de reportes se decidió utilizar algunas de las ideas propuestas por estos sistemas. Ejemplos de estas ideas a utilizar son las del Módulo Reportes del Sistema de Gestión Académica Akademos; permite el diseño y generación de reportes referentes a la información contenida en las hojas de matrícula y pre-matrícula. Otras ideas a utilizar serían las brindadas por el sistema GESTACAD (Sistema para la Gestión Académica), que da la posibilidad al usuario de seleccionar los campos de datos que desea obtener en el reporte así como el título de este y las condiciones que debe cumplir la información a mostrar, también se podría utilizar ideas brindadas por los sistemas Nefrosoft y Akademos los mismos brindan la posibilidad

de imprimir, exportar a .xls, .pdf, .doc los datos del reporte generado y se desea que este sistema también le brinde al usuario la opción de imprimir, exportar a .xls y a .pdf el reporte obtenido.

1.3 Metodologías y herramientas de desarrollo de software

1.3.1 Metodología

RUP

Proceso Unificado de Desarrollo constituye una metodología tradicional para el análisis, implementación y documentación de sistemas orientados a objeto. Presenta a UML como lenguaje de representación visual. Define en su modelación como principales elementos a:

- Trabajadores (quién).
- Actividades (cómo).
- Artefactos (qué).
- Flujo de Actividades (cuándo).

RUP ha agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo, los 6 primeros son conocidos como flujos de ingeniería y los restantes como flujos de apoyo.

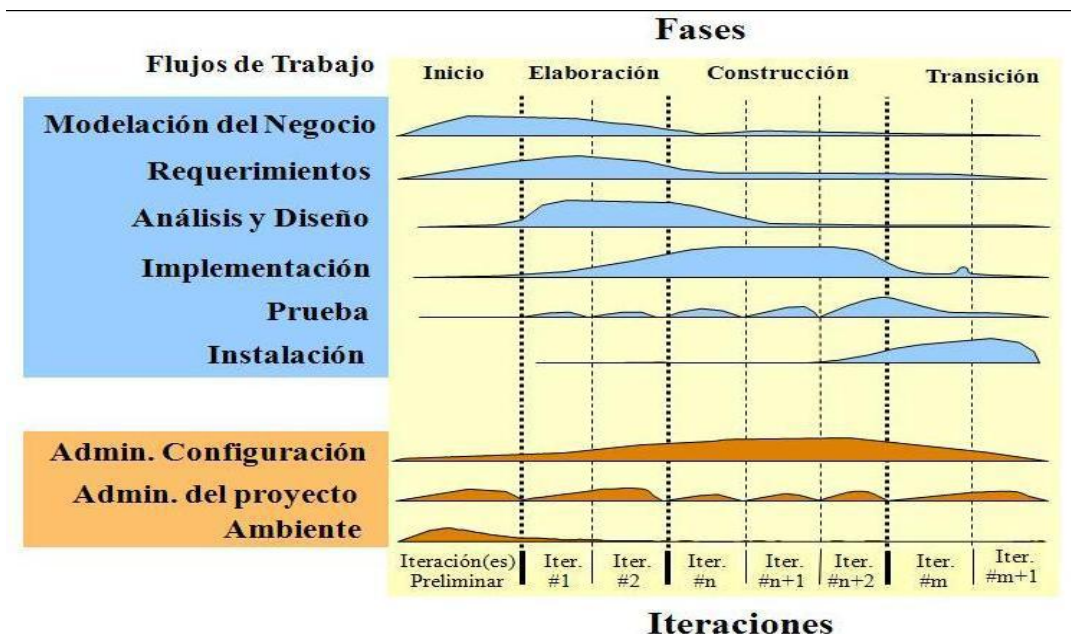


Figura 1: Estructura de RUP

Flujos de Trabajo

Flujos de Ingeniería:

- **Modelamiento del negocio:** comprende la estructura y la dinámica de la organización en la cual se va a implantar un sistema, comprender los problemas actuales de la organización e identifica las mejoras potenciales, asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización y deriva los requerimientos del sistema que va a soportar la organización.
- **Requerimientos:** establece y mantiene un acuerdo entre clientes y otros involucrados sobre lo que el sistema deberá hacer, provee a los desarrolladores un mejor entendimiento de los requerimientos del sistema, define el alcance del sistema, define una interfaz de usuario para el sistema, enfocada a las necesidades y metas del usuario, provee una base para estimar recursos y tiempo de desarrollo del sistema.
- **Análisis y Diseño:** proporciona una comprensión detallada de los requisitos, impone una estructura del sistema que debemos esforzarnos por conservar lo más fielmente posible cuando demos forma al sistema, por lo que indica con precisión lo que se debe programar.
- **Implementación:** define la organización del sistema en Subsistemas, se implementan los elementos de diseño, prueba los componentes desarrollados e integra los resultados en un sistema ejecutable.
- **Prueba:** encuentra y documenta los defectos que puedan afectar la calidad del software, valida que el software trabaje como fue diseñado, valida y prueba los requisitos que debe cumplir el software y valida que los requisitos fueron implementados correctamente.
- **Instalación:** produce una versión del producto y realiza actividades como empaque, instalación, asistencia a usuarios, para entregar el software a los usuarios finales.

Flujos de apoyo:

- **Administración de configuración y cambios:** describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a utilización y actualización concurrente de elementos, control de versiones.
- **Administración del proyecto:** involucra actividades con las que se busca obtener un producto que satisfaga las necesidades de los clientes.

- **Ambiente:** contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto, así como el procedimiento para implementar el proceso en una organización.

Fases:

- **Inicio:** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen a pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- **Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene uno o varias versiones del producto que han pasado las pruebas. Se ponen estas versiones a consideración de un subconjunto de usuarios.
- **Transición:** La versión ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

Características de RUP

- **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
- **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.
- **Iterativo e Incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y

diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración.

Tiene como principales ventajas:

- Coste del riesgo a un solo incremento
- Reduce el riesgo de no sacar el producto en el calendario previsto
- Acelera el ritmo de desarrollo
- Se adapta mejor a las necesidades del cliente. (3)

1.3.2 Lenguaje de Modelado

UML

Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software, está compuesto por diversos elementos gráficos que se combinan para conformar diagramas, ofrece un estándar para describir un plano del sistema, incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos, es importante recalcar que no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

El modelo gráfico de UML tiene un vocabulario en el que se identifican:

- Elementos: son abstracciones que constituyen los bloques básicos de construcción.
- Relaciones: son aquellas que ligan los elementos.
- Diagramas: son la representación gráfica de un conjunto de elementos y visualizan un sistema desde diferentes perspectivas.

Posee como principales ventajas:

- Mayor rigor en la especificación.
- Permite realizar una verificación y validación del modelo realizado.

- Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño, de más alto nivel, de la estructura de un proyecto. (4)

1.3.3 Herramienta CASE

Visual Paradigm

Visual Paradigm es una herramienta CASE² que utiliza UML como lenguaje de modelado. Fue desarrollada para el diseño de software desarrollados con el paradigma de programación orientado a objeto, trata reducir la duración del ciclo de desarrollo brindando ayuda tanto a arquitectos, analistas, diseñadores y desarrolladores, busca automatizar tareas tediosas que pueden distraer a los desarrolladores. Es una herramienta colaborativa porque soporta a varios usuarios trabajando en un mismo proyecto, genera la documentación del proyecto automáticamente en varios formatos como son Web o .pdf. Presenta la posibilidad de generar código a partir de los diagramas, para plataformas como .Net, Java y PHP, así como permite la obtención de diagramas a partir de código. Visual Paradigm para UML es multiplataforma, lo cual le permite al usuario utilizar esta herramienta en varios sistemas operativos como Windows, Linux, Unix y otros; además se encuentra disponible en distintas versiones: Enterprise, Professional, Standard, Modeler, Personal y Community.

Características del Visual Paradigm:

- Muy fácil de instalar y actualizar.
- Posee compatibilidad entre sus ediciones y soporte de UML versión 2.0.
- Soporta las notaciones definidas por BPMN.

Estas características permiten que esta herramienta sea óptima para realizar el modelado del software. Visual Paradigm ofrece Disponibilidad de integrarse en los principales Entornos de Desarrollo Integrado (IDE).

Presenta varias ventajas entre las que se destacan:

² Case: *Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadoras.

- Poderosa herramienta de generación de .pdf/.html a partir de diagramas UML.
- Sincronización entre el código fuente y el modelo en tiempo real.
- Soporte para toda la notación UML. (5)

1.4 Herramientas y Tecnologías para el Desarrollo de la Aplicación

1.4.1 Herramientas para la generación de reportes

JasperReports

JasperReports es una herramienta de código libre para generar reportes. Está escrita en Java y se puede utilizar en una gran variedad de aplicaciones de Java o aplicaciones Web, para generar contenido dinámico. Al utilizar JasperReports, es necesario añadirlo a las aplicaciones por medio de la inclusión de su librería, a la ruta de clase de la aplicación. Es posible utilizar JasperReports para aplicaciones de escritorio. Se pueden exportar los reportes a una multitud de formatos, como .pdf, .rtf, .html, .xml, .odt. La utilización de esta herramienta permite mejorar la gestión de la información, mediante la creación y gestión de informes. Con una solución de este tipo, se dispone en el tiempo deseado de los datos indispensables para la gestión eficaz de un departamento.

Para generar un reporte con JasperReports se emplean una serie de pasos más bien orientados al programador y no al cliente, ya que tiene que trabajar con distintos tipos de ficheros los cuales se muestran a continuación:

- Generar un fichero .jrxml en el que se configura cómo se quiere el informe
- Compilar el fichero .jrxml para obtener un fichero .jasper
- Rellenar los datos del informe. Esto generará un fichero .jrprint
- Exportar el fichero .jrprint al formato que se necesite .Esto generará el fichero en cuestión. (6)

IReport

Esta herramienta es un constructor/diseñador de informes, visual, poderoso, intuitivo y fácil de usar para JasperReports, y está escrito en Java. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, subinformes. Los datos para imprimir pueden ser recuperados por varias vías, incluso múltiples uniones JDBC, TableModels, JavaBeans, XML.

Las características más relevantes de IReport son:

- 100% escrito en Java y además es de código abierto y gratuito.
- Maneja el 98% de las etiquetas de JasperReports.
- Permite diseñar con sus propias herramientas: rectángulos, líneas, elipses, campos de textos, cartas, y subreportes.
- Soporta internacionalización nativamente.
- Buscador de la estructura del documento.
- Recopilador y exportador integrados.
- Incluye asistentes para crear automáticamente informes.
- Tiene asistentes para generar los subreportes.
- Tiene asistentes para las plantillas.
- Facilidad de instalación.(7)

DynamicJasper

DJ es un proyecto de código abierto que permite al desarrollador crear rápidamente una gran variedad de reportes a través de una intuitiva API escrita en Java. Esta es un conjunto de funciones y procedimientos o métodos, en la programación orientada a objetos que provee un conjunto de clases utilitarias para efectuar toda clase de tareas necesarias dentro de un programa. La API maneja todo lo relacionado con la diagramación y posicionamiento de los elementos del reporte haciendo el proceso de diseño fácil y automático.

DynamicJasper tiene como objetivo abarcar el 99% de los reportes que se basan en columnas. La API permite agregar variables en las cabeceras y pie de las columnas y grupos con operaciones tales como suma, contar, entre otras. Se puede definir en tiempo de ejecución el orden de aparición de las columnas, los grupos, las variables, los estilos, estilos condicionales y sub-reportes.

Características:

DJ usa directamente la API de JasperReports, por lo que no hay necesidad de archivos jrxml. Sin embargo, se pueden seguir utilizando como plantilla (marca de agua, logo de la compañía, entre otras) y permitir a DJ hacer la disposición de los elementos del reporte.

- Columnas Dinámicas: Se pueden definir las columnas del reporte y su orden de aparición en tiempo de ejecución. Cada una de ellas puede tener su propio estilo fuente de la cabecera de los datos, colores, bordes. El ancho de las columnas se acomoda automáticamente hasta alcanzar el ancho de la hoja.
- Grupo: Se pueden crear grupos dinámicamente usando una columna como criterio, o expresiones más complejas programadas en Java. Cada grupo tendrá su cabecera y pie, los cuales pueden tener variables mostrando subtotales de operaciones de las columnas.
- Formato condicional: DJ provee una manera sencilla de agregar formato condicional a los elementos de una columna. Se pueden utilizar expresiones sencillas escritas en Java para definir cuando utilizar un estilo en particular.
- Opciones de reportes dinámicos configurables en tiempo de ejecución tales como tamaño y orientación de página, márgenes y columnas. Los elementos siempre se acomodan a las nuevas condiciones.
- Exportar a los formatos más populares no es un problema, ya que DJ se basa en JasperReports y puede exportar los mismos formatos que este exporta. Aunque DJ agrega valor al poder exportar a .xls sin formateo de los datos, sin salto de página, sin celdas vacías en el medio, lo cual es ideal para usuarios que utilizarán las hojas de cálculo para hacer otras operaciones. Esto es difícil de lograr simplemente utilizando editores como IReport.
- Subreportes: Los sub reportes también pueden ser dinámicos. Se puede utilizar el DJ para concatenar varios reportes en uno solo.
- Auto texto: DJ tiene incorporado auto textos tales como “*Página 1 de 10*”, “Creado el 10 de Octubre de 2007”, o mensajes definidos por el usuario. Los mismos se pueden ubicar en la cabecera o pie de página del reporte.
- Imágenes en el encabezado: Se pueden definir hasta 3 imágenes para el encabezado de la primera página (izquierda, centro, derecha) y otras 3 distintas para el resto de las páginas. Ideal para agregar en una línea de código el logo de la organización.
- Gráficos: Agregar reportes se ha vuelto una tarea realmente sencilla a través de DJ. Solo hay que indicarle que columnas utilizar y el tipo de gráfico.

- Columnas Calculadas: La API de DJ provee una forma amigable de crear una columna calculada, las cuales pueden tener fila a fila una operación entre los elementos de la misma de otras columnas. (8)

1.4.2 Lenguaje de programación

JAVA

Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. Tiene un modelo de objetos simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Java es un lenguaje multiplataforma que permite realizar aplicaciones para todo tipo de entornos, para Web, dispositivos móviles, aplicaciones de escritorio, servidor, entre otros.

Consta de varias versiones o ediciones:

- Java empresarial se usa como se puede usar PHP, para crear aplicaciones web de gestión vinculadas a bases de datos
- JSE (Java para aplicaciones de escritorio) se usa para crear aplicaciones ejecutables que funcionarán en cualquier sistema operativo: Windows, Mac, Linux. Esta edición compite con .net de Microsoft.
- JME (Java para pequeños dispositivos) se usa para crear aplicaciones que correrán en dispositivos del tipo: móviles, PDAs, robótica, entre otros.

Características de Java:

- Robusto: realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución.
- Seguro: características como los punteros o el casting se eliminan para prevenir el acceso ilegal a la memoria.
- Portable: la portabilidad básica por ser de arquitectura independiente, implementa otros estándares de portabilidad para facilitar el desarrollo.
- Interpretado: el intérprete Java puede ejecutar directamente el código objeto.

- Simple: Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos.
- Orientado a objetos: Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo.
- Distribuido: Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales. (9)\

1.4.3 Entorno de Desarrollo

NetBeans

NetBeans es un entorno de desarrollo integrado de código abierto y de distribución gratuita que proporciona herramientas muy cómodas y de fácil uso para el desarrollo de aplicaciones sobre la plataforma Java. Es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas, se refiere a un entorno de desarrollo para aplicaciones, tanto de escritorio como aplicaciones empresariales en varias capas o programas para todo tipo de dispositivos móviles.

Entre las características que proporciona este entorno está el desarrollo de aplicaciones multiplataforma sobre: Mac OS, Windows, Linux. Incluye además:

- Soporte para Java, C, C++, XML y lenguajes HTML.
- Incluye control de versiones y compilación avanzada.
- Posibilidad de utilizar otras versiones de compiladores, depuradores.
- Creación visual de componentes gráficos.
- Herramientas con asistentes para facilitar la escritura de código.
- Dispone de todo un entorno para crear documentación javadoc. (10)

1.4.4 Sistema Gestor de base de datos

Hoy en día, prácticamente todo software de gestión necesita tener acceso a su amplio volumen de información de una forma óptima y segura. Los sistemas gestores de bases de datos son los programas encargados de realizar ésta y muchas otras funciones; por lo que el presente sistema también debe hacer uso de ellos. Se han tomado en cuenta como posibles sistemas gestores de base de datos a utilizar:

DB4O

Es un novedoso motor de base de datos orientada a objetos. Sus siglas se corresponden con la expresión "DataBase 4 (for) Object", que a su vez es el nombre de la compañía que lo desarrolla: db4objects, Inc. Este es un software de código abierto, disponible bajo licencia GNU GPL ³y bajo una licencia comercial. DB4O está diseñado para ser embebido en clientes u otros componentes de software, de manera totalmente invisible para el usuario final, se puede emplear en aplicaciones de escritorio que no requieran de un motor de base datos grandes. Posibilita una mayor longevidad del software. Se puede utilizar para dispositivos móviles, médicos, en la industria del transporte, en aplicaciones Web y en sistemas en tiempo real.

DB4O presenta características interesantes:

- No hay diferencia de impedancia: los objetos se almacenan tal y como son.
- Manejo automático del esquema de base de datos.
- No hay que cambiar las clases para poder almacenarlas.
- Uniones de datos automatizadas.
- Se instala añadiendo un único fichero de librería de 250Kb .jar para Java.
- Un único fichero de base de datos.
- Versionado del esquema automático.
- S.O.D.A. (Simple Object Database Access), un API de consultas de código abierto.(11)

³ GNU GPL: Licencia Pública General de GNU o *GNU General Public License*, es una licencia creada por la Fundación del Software Libre a mediados de los 80, orientada a proteger la libre distribución, modificación y uso de software.

MySQL

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario, probablemente, el gestor de bases de datos más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación (C++, Java, PHP), además de su fácil instalación y configuración. Además, posee gran fortaleza en la gestión de usuarios y contraseñas manteniendo un muy buen nivel de seguridad en los datos. MySQL es muy utilizado en aplicaciones web, como Drupal o phpBB, en plataformas Linux/Windows-Apache-MySQL-PHP/Perl/Python y por herramientas de seguimiento de errores como Bugzilla. Es una base de datos muy rápida en la lectura, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. MySQL fue diseñada para ser un gestor de bases de datos con una gran rapidez de respuesta. Fue adquirido por Sun Microsystems, ahora está patrocinado por una empresa propietaria, que posee el copyright de la mayor parte del código.

PostgreSQL

Potente sistema gestor de base de datos relacional orientado a objetos de software libre, publicado bajo la licencia BSD, desarrollado por Grupo Global de Desarrollo de PostgreSQL. PostgreSQL tiene un activo desarrollo y una probada arquitectura que se ha ganado una sólida reputación de fiabilidad, es una excelente solución para la administración de la base de datos. Funciona en todos los principales sistemas operativos, como Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. La nueva versión 8.3 incluye una cantidad récord de características nuevas y mejoradas respecto a versiones anteriores. La velocidad de respuesta que ofrece este gestor con bases de datos relativamente pequeñas puede parecer un poco deficiente, aunque esta misma velocidad la mantiene al gestionar bases de datos realmente grandes. A su vez, posee una gran escalabilidad y es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta, acoplándose, fundamentalmente esta característica a las necesidades del sistema.

SQL Server

Microsoft SQL Server es un Sistema Gestor de base de datos relacional basado en el lenguaje Transact-SQL, y específicamente en *Sybase IQ*, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son *Oracle*, *Sybase ASE*, *PostgreSQL*, *Interbase*, *Firebird* o *MySQL*. Las características principales de Microsoft SQL Server son:

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida (MSDE), con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en sus versiones 2005 y 2008 pasan a ser el SQL Express Edition, que se distribuye en forma gratuita. En el manejo de SQL mediante líneas de comando se utiliza el SQLCMD. Para el desarrollo de aplicaciones más complejas (tres o más capas), *Microsoft SQL Server* incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor sólo está disponible para Sistemas Operativos Windows.

Microsoft Access

Es un sistema de gestión de base de datos relacional creado y modificado por Microsoft para uso personal en pequeñas organizaciones. Es un componente de la suite Microsoft Office, aunque no se incluye en el paquete "básico". Una posibilidad adicional es la de crear ficheros con bases de datos que pueden ser consultados por otros programas. Este programa permite manipular datos en forma de tablas (la cual es la unión de filas y columnas), realizar cálculos complejos con fórmulas y funciones, incluso dibujar distintos tipos de gráficas. Microsoft Office Access permite crear formularios para insertar y modificar datos fácilmente. También tiene un entorno gráfico para ver las relaciones entre las diferentes

tablas de la base de datos. Tiene un sistema de seguridad de cifrado bastante primitivo y puede ser la respuesta a proyectos de programación de pequeños y medianos tamaños.

Se decidió utilizar Microsoft Access debido a que la juventud nacional no cuenta con computadoras cuyas propiedades soporten un gestor de base de datos más potente y además fue definido por la dirección del proyecto.

1.5 Conclusiones

Se concluye que:

- Después de realizado un estudio sobre diferentes sistemas de gestión de información y obtención de reportes referente a los cuadros de la UJC Nacional, no existen antecedentes de sistemas a nivel internacional relacionados con la UJC ya que esta es una organización que solo radica en Cuba y en el ámbito nacional no se conoció ningún sistema con igual propósito por lo que es necesario implementar los Submódulos de reporte y proyección de movimiento de los cuadros de la UJC Nacional.
- Las herramientas a utilizar en el desarrollo del proyecto son: UML como lenguaje de modelado, Visual Paradigma como herramienta CASE, DynamicJasper como generador de reportes, Microsoft Access como gestor de base de datos, RUP como metodología de desarrollo, JAVA como el lenguaje de programación y NetBeans como entorno de desarrollo.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

2.1 Introducción

Para el desarrollo de un sistema es necesario comprender cómo se realizan cada uno de los procesos que van a ser automatizados y de esta manera lograr un entendimiento entre los desarrolladores y los clientes, para lograr todo esto se debe realizar el modelamiento del negocio con el objetivo de comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema, comprender los problemas actuales de la organización e identificar las mejoras potenciales, asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización y derivar los requerimientos del sistema que va a soportar la organización.

En este capítulo se realiza la crítica del análisis y diseño propuesto con anterioridad, se analizan posibles implementaciones, componentes o módulos ya existentes que puedan ser rehusados, se definen requisitos funcionales y no funcionales que se tienen en cuenta para el desarrollo de la aplicación. Además, se identifican y se describen los casos de uso del sistema.

2.2 Valoración crítica del diseño propuesto por el analista.

En la fase de análisis y diseño se realizó un excelente trabajo, las clases del diseño están bien representadas ya que desde él se pueden obtener las clases fundamentales que deben ser definidas para que el sistema funcione satisfactoriamente, así como los atributos y métodos que deben tener las mismas, dando una idea clara de lo que se debe implementar. Los requisitos funcionales descritos por los analistas del sistema facilitaron el entendimiento de los procesos a implementar permitiendo una buena comprensión del problema y posibilitando la identificación de las clases y las funcionalidades a desarrollar. Tomándose como base la descripción detallada de los requisitos funcionales, se puede establecer una estrategia de trabajo que permita la implementación de dichas funcionalidades.

2.2.3 Cambios de gestor de base de datos

A pesar de que DB4O es un novedoso motor de base de datos orientada a objetos, un software de código abierto, embebido en clientes u otros componentes de software, de manera totalmente invisible para el usuario final y se pueden emplear en aplicaciones de escritorio que no requieran de un motor de base de datos grandes, se vuelve muy complicado el trabajo con las consultas a la hora de combinar varias tablas y obtener reportes de datos de las mismas, por lo que después de haber realizado un estudio de gestores

Capítulo 2: Descripción y análisis de la solución propuesta.

de base de datos como son Access, Postgre, MySQL, entre otros, la UJC Nacional y la dirección del proyecto ha decidido la utilización de Access debido a que sus estaciones de trabajo no cuentan con las condiciones necesarias para la instalación de gestores que necesiten de mayor memoria y capacidad de almacenamiento y además Access es un sistema de gestión de base de datos relacional que reduce la duplicación en el ingreso de datos, permite ver las relaciones entre las diferentes tablas de la base de datos, permite crear pequeñas aplicaciones con unos pocos conocimientos de programación y además al ser relacional es capaz de manejar una o más tablas de datos a la vez.

2.2.4 Estructura de la base de datos

Después de realizado un análisis del diseño de la base de datos se decide que el propuesto no satisface las necesidades que se requieren para la implementación, debido a la ausencia de tablas necesaria para lograr un mejor acceso y control a los datos por lo que se crean nuevas tablas Vivienda, Datos Interés, Núcleo Familiar, Reserva Cuadro, Estudios Militante, Movilizaciones, Servicio Militar Activo, Dirección Particular, Sanción Admin Jurídicas, Trayectoria Estudiantil, Miembro Comité, Trayectoria Militar, Familiar Extranjero, Estado Salud. A continuación se verá la descripción de cada una de las nuevas tablas. ([Ver diseño de la base de datos en el Anexo 1](#))

Descripción de las nuevas tablas

Vivienda

Nombre: Tb_dVivienda.		
Descripción: Esta clase permite almacenar todos los datos referentes a la vivienda.		
Atributo	Tipo	Descripción
id_vivienda	int	Identificador de la vivienda.
problemas	String	Los problemas que presenta la vivienda.

Tabla 2.1: Descripción Vivienda

Reserva Cuadro

Nombre: Tb_dReserva_Cuadro.		
Descripción: Esta clase permite almacenar todos los datos referentes a las reservas de cuadro.		
Atributo	Tipo	Descripción
id_reserva	int	El identificador de la reserva.

Tabla 2.2: Descripción Reserva Cuadro

Capítulo 2: Descripción y análisis de la solución propuesta.

Datos de Interés.

Nombre: Tb_dDatos_Interes.		
Descripción: Esta clase permite almacenar todos los datos de interés de un militante.		
Atributo	Tipo	Descripción
id_datos	int	Identificador de datos.
datos	String	Los datos del militante.

Tabla 2.3: Descripción Datos Interés

Núcleo Familiar

Nombre: Tb_dNucleo_Familiar.		
Descripción: Esta clase permite almacenar todos los datos referentes al núcleo familiar.		
Atributo	Tipo	Descripción
CI	String	El carnet de identidad.
nombre	String	El nombre.
pmer_apellido	String	El primer apellido.
sgdo_apellido	String	El segundo apellido.

Tabla 2.4: Descripción Núcleo Familiar

Estudios Militante

Nombre: Tb_dEstudios_Militante.		
Descripción: Esta clase permite almacenar todos los datos de los estudios de los militantes.		
Atributo	Tipo	Descripción
id_estudios	Int	El identificador del estudio.
especialidad	string	La especialidad estudiada en su carrera.
graduado_pedagogico	string	Si es graduado pedagógico.
estudia	string	Si estudia actualmente.

Tabla 2.5: Estudios Militante

Capítulo 2: Descripción y análisis de la solución propuesta.

Movilizaciones

Nombre: Tb_dMovilizaciones		
Descripción: Esta clase permite almacenar todos los datos referentes a las movilizaciones.		
Atributo	Tipo	Descripción
id_movilización	int	El identificador de la movilización.
act_realizada	string	Las actividades realizadas.
anno	int	El año

Tabla 2.6: Descripción Movilizaciones

Servicio Militar Activo

Nombre: Tb_dServicio_Militar_Activo		
Descripción: Esta clase permite almacenar todos los datos referentes al servicio militar activo.		
Atributo	Tipo	Descripción
id_servicio	int	El identificador del servicio.
cumplimiento	string	Valoración del cumplimiento del servicio militar activo
unidad_militar	string	El nombre de la unidad militar.
causa_no_SMA	string	La causa de por qué no paso el servicio militar

Tabla 2.7: Descripción Servicio Militar Activo

Dirección Particular

Nombre: Tb_dDireccion_Particular		
Descripción: Esta clase permite almacenar todos los datos referentes a la dirección particular.		
Atributo	Tipo	Descripción
id_direccion	int	El identificador de la dirección.
calle	string	La calle.
no_casa	Int	El número de la casa.
apartamento	string	El apartamento.
entre_calles	string	La entre calle.
reparto	string	El nombre del reparto

Tabla 2.8: Descripción Dirección Particular

Capítulo 2: Descripción y análisis de la solución propuesta.

Sanción Admin Jurídicas

Nombre: Tb_dSancion_Admin_Juridicas		
Descripción: Esta clase permite almacenar todos los datos referentes a las sanciones administrativas jurídicas.		
Atributo	Tipo	Descripción
id_sancion	int	El identificador de la sanción.
tipo _ sanción	string	El tipo de sanción.
fecha _ sanción	date	La fecha de la sanción.
centro _ trabajo	string	El centro de trabajo actual
causas	string	Descripción abreviada de la causa.

Tabla 2.9: Descripción Sanción Admin Jurídicas

Trayectoria Estudiantil

Nombre: Tb_dTrayectoria_Estudiantil		
Descripción: Esta clase permite almacenar todos los datos referentes a la trayectoria estudiantil.		
Atributo	Tipo	Descripción
id_trayectoria	int	El identificador de la trayectoria estudiantil.
nombre _ CE	string	El nombre completo del centro de estudio.
fecha_ingreso	date	La fecha de ingreso
fecha_salida	date	La fecha de salida.

Tabla 2.10: Descripción Trayectoria Estudiantil

Miembro Comité

Nombre: Tb_dMiembro_Comite		
Descripción: Esta clase permite almacenar todos los datos referentes al miembro del comité.		
Atributo	Tipo	Descripción
id_miembro	int	El identificador del miembro.
es_miembro	string	Si es miembro.

Tabla 2.11: Descripción Miembro Comité

Capítulo 2: Descripción y análisis de la solución propuesta.

Trayectoria Militar

Nombre: Tb_dTrayectoria_Militar.		
Descripción: Esta clase permite almacenar todos los datos referentes a la trayectoria militar.		
Atributo	Tipo	Descripción
id_trayectoria	int	El identificador de la trayectoria.
especialidad	int	La especialidad.
fecha_ingreso	date	La fecha de ingreso.
fecha_salida	date	La fecha de salida.
nombre_UM	int	El nombre de la unidad militar.

Tabla 2.12: Descripción Trayectoria Militar

Familiar Extranjero

Nombre: Tb_dFamiliar_Extranjero.		
Descripción: Esta clase permite almacenar todos los datos referentes al país donde se encuentra el familiar extranjero.		
Atributo	Tipo	Descripción
pais	string	El nombre del país donde se encuentra.

Tabla 2.13: Descripción Familiar Extranjero

Estado Salud

Nombre: Tb_dEstado_Salud.		
Descripción: Esta clase permite almacenar todos los datos referentes al estado de salud.		
Atributo	Tipo	Descripción
id_estado_salud	int	El identificador del estado de salud.
grupo_sang	string	El grupo sanguíneo.
enf_padece	string	Las enfermedades que padece.
enf_padecidas	int	Cantidad de enfermedades padecidas.

Tabla 2.14: Descripción Estado Salud

También se hace uso de nomencladores los cuales preparan el sistema para que este pueda responder las demandas de los usuarios, además se utilizan con el fin de estandarizar la información almacenada y facilitar así la recuperación de la misma. Los nomencladores creados se mencionan a continuación

Capítulo 2: Descripción y análisis de la solución propuesta.

Sector, Rama, Países, Motivo Visita, Tipo Mérito, Año, Estado Vivienda, Tipo Militancia, Parentesco, Tipo Estructura, Provincia, Municipio, Cargo Actual, Tipos Estudios, Categoría Militante, Escuelas Políticas, Tipo Estructura Superior, Tipo Causa, Sub Tipo Causa, Nombre Organización, Estados Militantes, Nivel Cultural, Tipo Miembro, Composición Étnica, Cargo DB, Grado Militar, Clasificador Ocupacional Cuadro, Clasificador Ocupacional Reserva, Labor, Vía, Motivo, Estado, Procedencia Laboral, Sanción Cuadro, Causa Sanción.

Descripción de los nomencladores utilizados

Sector

Nombre: Tb_nSector.		
Descripción: Esta clase permite almacenar los datos de un sector.		
Atributo	Tipo	Descripción
id_sector	int	El identificador del sector.
nombre	string	El nombre del sector.

Tabla 2.15: Descripción del nomenclador Sector

Rama

Nombre: Tb_nRama.		
Descripción: Esta clase permite almacenar los datos referentes a la rama de trabajo.		
Atributo	Tipo	Descripción
id_rama	int	El identificador de la rama.
nombre	string	El nombre de la rama.

Tabla 2.16: Descripción del nomenclador Rama

Países

Nombre: Tb_nPaíses		
Descripción: Esta clase permite almacenar todos los datos referentes a los países.		
Atributo	Tipo	Descripción
id_pais	int	El identificador del país
nombre	string	El nombre del país.

Tabla 2.17: Descripción del nomenclador Países

Capítulo 2: Descripción y análisis de la solución propuesta.

Motivo Visita

Nombre: Tb_nMotivo_Visita.		
Descripción: Esta clase permite almacenar todos los datos referentes a los motivos de la visita.		
Atributo	Tipo	Descripción
id_motivo	int	El identificador del motivo.
motivo	string	El motivo de la visita.

Tabla 2.18: Descripción del nomenclador Motivo Visita

Tipo Mérito

Nombre: Tb_nTipo_Merito.		
Descripción: Esta clase permite almacenar todos los datos referentes a los tipos de méritos.		
Atributo	Tipo	Descripción
id_tipo_merito	int	Identificador del tipo de mérito.
tipo_merito	string	El tipo de mérito.

Tabla 2.19: Descripción del nomenclador Tipo Mérito

Año

Nombre: Tb_nAño.		
Descripción: Esta clase permite almacenar todos los datos referentes a los años		
Atributo	Tipo	Descripción
id_año	int	Identificador del año.
año	int	El año.

Tabla 2.20: Descripción del nomenclador Año

Estado Vivienda

Nombre: Tb_nEstado_Vivienda.		
Descripción: Esta clase permite almacenar todos los datos referentes al estado de la vivienda.		
Atributo	Tipo	Descripción
id_estado	int	Identificador de estado.
estado	string	El estado que presenta la vivienda.

Tabla 2.21: Descripción del nomenclador Estado Vivienda

Capítulo 2: Descripción y análisis de la solución propuesta.

Tipo Militancia

Nombre: Tb_nTipo_Militancia.		
Descripción: Esta clase permite almacenar todos los datos referentes al tipo de militancia.		
Atributo	Tipo	Descripción
id_militancia	int	El identificador de la militancia.
tipo_militancia	string	El tipo de militancia.

Tabla 2.22: Descripción del nomenclador Tipo Militancia

Parentesco

Nombre: Tb_nParentesco.		
Descripción: Esta clase permite almacenar todos los datos referentes al tipo de militancia.		
Atributo	Tipo	Descripción
id_parentesco	int	El identificador de la militancia.
nombre	string	El nombre del parentesco.

Tabla 2.23: Descripción del nomenclador Parentesco

Provincia

Nombre: Tb_nProvincia.		
Descripción: Esta clase permite almacenar todo lo referente a las provincias.		
Atributo	Tipo	Descripción
id_provincia	int	El identificador de la provincia.
nombre	string	El nombre de la provincia.

Tabla 2.24: Descripción del nomenclador Provincia

Municipio

Nombre: Tb_nMunicipio.		
Descripción: Esta clase permite almacenar todo lo referente a los municipios.		
Atributo	Tipo	Descripción
id_municipio	Int	El identificador del municipio.
nombre	string	El nombre del municipio.

Tabla 2.25: Descripción del nomenclador Municipio

Capítulo 2: Descripción y análisis de la solución propuesta.

Cargo Actual

Nombre: Tb_nCargo.		
Descripción: Esta clase permite almacenar todo lo referente a los municipios.		
Atributo	Tipo	Descripción
id_cargo	Int	El identificador del cargo.
nombre	string	El nombre del cargo.

Tabla 2.26: Descripción del nomenclador Cargo Actual

Tipos Estudios

Nombre: Tb_nTipos_Estudios.		
Descripción: Esta clase permite almacenar todos los datos referentes a los tipos de estudios.		
Atributo	Tipo	Descripción
id_tipo_est	int	El identificador del tipo de estudio.
nombre	string	El nombre del estudio.

Tabla 2.27: Descripción del nomenclador Tipos Estudios

Categoría Militante

Nombre: Tb_nCategoria_Militante		
Descripción: Esta clase permite almacenar todos los datos referentes a las categorías de los militantes.		
Atributo	Tipo	Descripción
id_categoria	int	El identificador de la categoría.
nombre	string	El nombre de la categoría.

Tabla 2.28: Descripción del nomenclador Categoría Militante

Escuelas Políticas

Nombre: Tb_nEscuelas_Políticas.		
Descripción: Esta clase permite almacenar todos los datos referentes a las escuelas políticas.		
Atributo	Tipo	Descripción
id_escuela	int	El identificador de la escuela.
nombre	string	El nombre de la escuela.

Tabla 2.29: Descripción del nomenclador Escuelas Políticas

Capítulo 2: Descripción y análisis de la solución propuesta.

Tipo Estructura Superior

Nombre: Tb_nTipo_Estructura_Superior.		
Descripción: Esta clase permite almacenar todos los datos referentes al tipo de estructura superior.		
Atributo	Tipo	Descripción
id_tipo_sup	int	El identificador del tipo de estructura superior.
estructura	string	El nombre de la estructura.

Tabla 2.30: Descripción del nomenclador Tipo Estructura Superior

Tipo Causa

Nombre: Tb_nTipoCausa.		
Descripción: Esta clase permite almacenar todos los datos referentes a las causas de la sanción.		
Atributo	Tipo	Descripción
id_causa	int	El identificador de la causa.
causa	string	Descripción abreviada de la causa.

Tabla 2.31: Descripción del nomenclador Tipo Causa

Sub Tipo Causa

Nombre: Tb_nSubTipoCausa.		
Descripción: Esta clase permite almacenar todos los datos referentes a otras causas no registradas.		
Atributo	Tipo	Descripción
id_subcausa	int	El identificador de la causa.
causa	string	Descripción abreviada de la causa.

Tabla 2.32: Descripción del nomenclador Sub Tipo Causa

Nombre Organización

Nombre: Tb_nNombre_Organizacion		
Descripción: Esta clase permite almacenar todos los datos referentes a los nombres de la organización existentes en el país.		
Atributo	Tipo	Descripción
id_org	int	El identificador de la organización.
nombre	string	El nombre de la organización.

Tabla 2.33: Descripción del nomenclador Nombre Organización

Estados Militantes

Nombre: Tb_nEstados_Militantes		
Descripción: Esta clase permite almacenar todos los datos referentes a los estados de los militantes.		
Atributo	Tipo	Descripción
id_estado	int	El identificador del estado.
nombre	string	El nombre del estado.

Tabla 2.34: Descripción del nomenclador Estados Militantes

Nivel Cultural

Nombre: Tb_nNivel_Cultural		
Descripción: Esta clase permite almacenar todos los datos referentes al nivel cultural		
Atributo	Tipo	Descripción
id_nivel	int	El identificador del nivel escolar.
nombre	string	El nombre del nivel escolar.

Tabla 2.35: Descripción del nomenclador Nivel Cultural

Tipo Miembro

Nombre: Tb_nTipo_Miembro		
Descripción: Esta clase permite almacenar todos los datos referentes a los tipos de miembro.		
Atributo	Tipo	Descripción
id_tipo_miembro	int	El identificador del tipo de miembro.
tipo_miembro	string	El tipo de miembro.

Tabla 2.36: Descripción del nomenclador Tipo Miembro

Capítulo 2: Descripción y análisis de la solución propuesta.

Composición Étnica

Nombre: Tb_nComposicion_Etnica		
Descripción: Esta clase permite almacenar todos los datos referentes a la composición étnica.		
Atributo	Tipo	Descripción
id_comp_etnica	int	El identificador de la composición étnica.
comp_etnica	string	La composición étnica.

Tabla 2.37: Descripción del nomenclador Composición Étnica

Cargo DB

Nombre: Tb_nCargo_DB		
Descripción: Esta clase permite almacenar todos los datos referentes al cargo de dirigente de base.		
Atributo	Tipo	Descripción
id_cargo	int	El identificador del cargo.
nombre	string	El nombre del cargo.

Tabla 2.38: Descripción del nomenclador Cargo DB

Grado Militar

Nombre: Tb_nGrado_Militar.		
Descripción: Esta clase permite almacenar todos los datos referentes al grado militar.		
Atributo	Tipo	Descripción
id_grado	int	El identificador del grado.
nombre	string	El nombre del grado.

Tabla 2.39: Descripción del nomenclador Grado Militar

Clasificador Ocupacional Cuadro

Nombre: Tb_nClasificador_Ocupacional_C.		
Descripción: Esta clase permite almacenar todos los datos referentes al clasificador ocupacional de cuadro.		
Atributo	Tipo	Descripción
id_clasificador_c	int	El identificador del clasificador ocupacional de cuadro.
nombre	string	El nombre del clasificador.

Tabla 2.40: Descripción del nomenclador Clasificador Ocupacional Cuadro

Clasificador Ocupacional Reserva

Nombre: Tb_nClasificador_Ocupacional_R		
Descripción: Esta clase permite almacenar todos los datos referentes al clasificador ocupacional de reserva.		
Atributo	Tipo	Descripción
id_clasificador_r	int	El identificador del clasificador ocupacional de reserva.
nombre	string	El nombre del clasificador.

Tabla 2.41: Descripción del nomenclador Clasificador Ocupacional Reserva

Labor

Nombre: Tb_nLabor.		
Descripción: Esta clase permite almacenar todos los datos referentes a la labor.		
Atributo	Tipo	Descripción
id_labor	int	El identificador de la labor.
nombre	string	El nombre de la labor.

Tabla 2.42: Descripción del nomenclador Labor

Capítulo 2: Descripción y análisis de la solución propuesta.

Vía

Nombre: Tb_nVia.		
Descripción: Esta clase permite almacenar todos los datos referentes a la vía por la que el familiar extranjero abandono el país.		
Atributo	Tipo	Descripción
id_via	int	El identificador de la vía
via	string	La vía que utilizó para viajar.

Tabla 2.43: Descripción del nomenclador Via

Motivo

Nombre: Tb_nMotivo.		
Descripción: Esta clase permite almacenar todos los datos referentes al motivo del abandono del país del familiar.		
Atributo	Tipo	Descripción
id_motivo	int	El identificador del motivo.
motivo	string	El motivo del abandono del familiar.

Tabla 2.44: Descripción del nomenclador Motivo

Estado

Nombre: Tb_nEstado.		
Descripción: Esta clase permite almacenar todos los datos referentes al estado de salud.		
Atributo	Tipo	Descripción
id_estado	int	El identificador del estado.
estado	string	Breve descripción de su estado de salud.

Tabla 2.45: Descripción del nomenclador Estado

Procedencia Laboral

Nombre: Tb_nProcedencia_Laboral.		
Descripción: Esta clase permite almacenar todos los datos referentes a la procedencia laboral.		
Atributo	Tipo	Descripción
id_procedencia	int	El identificador de la procedencia laboral.
procedencia	string	La procedencia laboral.

Tabla 2.46: Descripción del nomenclador Procedencia Laboral

Sanción Cuadro

Nombre: Tb_nSancion_Cuadro.		
Descripción: Esta clase permite almacenar todos los datos referentes a la sanción de cuadros.		
Atributo	Tipo	Descripción
id_sancion	int	El identificador de la sanción.
nombre	string	El nombre de la sanción.

Tabla 2.47: Descripción del nomenclador Sanción Cuadro

2.3 Análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser rehusados. Estrategias de integración.

Aquí se analizan los beneficios del reporteador DynamicJasper que por su importancia y usabilidad son necesarios para el correcto funcionamiento del submódulo de reportes. A continuación se describen sus principales características y como se integra con el IDE de desarrollo NetBeans.

DJ crea rápidamente una gran variedad de reportes a través de una intuitiva API escrita en Java. La API maneja todo lo relacionado con la diagramación y posicionamiento de los elementos del reporte haciendo el proceso de diseño fácil y automático, provee una forma amigable de crear una columna calculada, las cuales pueden tener fila a fila una operación entre los elementos de la misma de otras columnas además puede exportar documentos en diferentes formatos como son Excel, PDF, Open Office, Word, HTML, entre otros.

Capítulo 2: Descripción y análisis de la solución propuesta.

Este reporteador se integra fácilmente con el IDE de desarrollo a continuación se verán los pasos para realizar esta integración:

1. Se crea una carpeta en el proyecto con el nombre (lib).
2. Se copia en esta carpeta el DynamicJasper-3.0.6.jar posibilitando así que para donde quiera que se traslade la aplicación el reporteador formará parte de ella.
3. Selecciona la opción “Agregar Biblioteca”, se busca en la dirección donde se copió y se selecciona la opción “Añadir Biblioteca”.

2.4 Paradigma de programación

El paradigma de programación utilizado es el orientada a objeto (POO) es una de las formas más populares de programar y viene teniendo gran acogida en el desarrollo de proyectos de software desde los últimos años. Esta acogida se debe a sus grandes capacidades y ventajas respecto a otros paradigmas de la programación:

- Fomenta la reutilización y extensión del código.
- Permite crear sistemas más complejos.
- Relacionar el sistema al mundo real.
- Facilita la creación de programas visuales.
- Construcción de prototipos
- Agiliza el desarrollo de software
- Facilita el trabajo en equipo
- Facilita el mantenimiento del software

2.4.1 Características asociadas a la POO

Abstracción

La abstracción consiste en captar las características esenciales de un objeto, así como su comportamiento. En los lenguajes de programación orientada a objetos, el concepto de Clase es la representación y el mecanismo por el cual se gestionan las abstracciones.

Capítulo 2: Descripción y análisis de la solución propuesta.

Encapsulamiento

El encapsulamiento consiste en unir en la Clase las características y comportamientos, esto es, las variables y métodos. Es tener todo esto en una sola entidad. En los lenguajes estructurados esto era imposible. Es evidente que el encapsulamiento se logra gracias a la abstracción y el ocultamiento.

Ocultamiento

Es la capacidad de ocultar los detalles internos del comportamiento de una Clase y exponer sólo los detalles que sean necesarios para el resto del sistema.

El ocultamiento permite 2 cosas: restringir y controlar el uso de la Clase. Restringir porque habrá cierto comportamiento privado de la Clase que no podrá ser accedido por otras Clases y controlar porque daremos ciertos mecanismos para modificar el estado de nuestra Clase y es en estos mecanismos donde se validarán que algunas condiciones se cumplan. En Java el ocultamiento se logra usando las palabras reservadas: public, private y protected delante de las variables y métodos.

2.5 Requisitos funcionales

Módulo Administración

1 Buscar Usuario

El administrador podrá buscar el usuario utilizando el criterio de búsqueda: rol.

2 Gestionar Usuarios

2.1 Agregar un usuario

El sistema debe permitir insertar Nombre del usuario, contraseña, nombre completo del usuario, seleccionar el rol que va a tener el usuario.

Tipos de roles:

Funcionario municipal: tiene acceso a la gestión de la información relacionada con los cuadros de la UJC Municipal, gestionar la evaluación de los cuadros de la UJC Municipal, así como obtener los reportes de la UJC Municipal.

Funcionario provincial : tiene acceso a la gestión de la información relacionada con los cuadros de la UJC Provincial, gestionar la evaluación de los cuadros de la UJC Provincial, así como obtener los reportes de la UJC Provincial y Municipal.

Funcionario movimiento nacional: tiene acceso a la gestión de la información relacionada con los cuadros de la UJC Nacional y tiene acceso a obtener los reportes de la UJC Nacional, Provincial y Municipal.

Capítulo 2: Descripción y análisis de la solución propuesta.

Funcionario de control estadístico nacional: tiene acceso a obtener los reportes de la UJC Nacional, Provincial, Municipal y gestionar la evaluación de los cuadros de la UJC Nacional.

Administrador: tiene acceso a la gestión de los usuarios.

2.2 Modificar Datos de Usuario

El sistema debe permitir buscar el usuario, seleccionar el usuario y modificar sus datos (usuario, contraseña, nombre completo, rol).

2.3 Eliminar Usuario

El sistema debe permitir buscar el usuario, seleccionar el usuario y eliminarlo.

3 Autenticar Usuario

El sistema debe permitir el acceso al sistema, entrando usuario y contraseña. Se tendrá acceso sólo a las acciones que puede hacer según el rol.

4 Cambiar Contraseña

El sistema debe permitir al usuario cambiar su contraseña, entrando su usuario, contraseña actual, nueva contraseña y confirmando la nueva contraseña.

Módulo Proyección de Movimientos

5 Gestionar estructura organización

5.1 Crear estructura

El sistema debe permitir seleccionar:

- Tipo de Estructura.
- A quién se subordina la estructura.
- En el caso de los comités provinciales y municipales, permitirá seleccionar provincia y municipio.
- Comité Nacional UJC
- Comité Provincial UJC
- Comité Municipal UJC

5.2 Modificar estructura

El sistema debe permitir seleccionar una estructura y modificar sus datos a quién se subordina, provincia, municipio.

5.3 Listar estructura

El sistema debe mostrar el árbol de dependencia de todas las estructuras creadas.

Capítulo 2: Descripción y análisis de la solución propuesta.

6 Gestionar cargos

El sistema debe permitir gestionar los cargos de una estructura para los diferentes niveles de dirección.

6.1 Crear estructura de cargos

El sistema debe permitir realizar, la ubicación de la estructura a la que se le gestionarán los cargos.

- ↵ Seleccionar tipo de estructura.
- ↵ Seleccionar provincia.
- ↵ Seleccionar municipio.

Para la ubicación realizada permitirá seleccionar:

- ↵ Nombre del cargo.
- ↵ Cargo Superior.
- ↵ A quién pertenece.

6.2 Modificar estructura cargos

El sistema debe permitir seleccionar una determinada estructura y modificar los datos de los cargos nombre del cargo, cargo superior, a quién pertenece.

6.3 Listar estructura cargos

El sistema debe permitir mostrar el árbol de dependencia de todos los cargos de la estructura seleccionada.

7 Asignar cargos

El sistema debe mostrar un listado con el nombre, apellidos y el cargo del Cuadro UJC, en la medida que se realice la asignación. Para asignar los cargos el sistema permitirá:

- ↵ Seleccionar estructura.
- ↵ Seleccionar cargo.
- ↵ Insertar primer nombre.
- ↵ Insertar segundo nombre.
- ↵ Insertar primer apellido.
- ↵ Insertar segundo apellido.
- ↵ Total de Comités de Bases que atiende Para Instructor y Funcionarios.
- ↵ Zona de Trabajo Para Instructor y Funcionarios.

Capítulo 2: Descripción y análisis de la solución propuesta.

8 Gestionar reserva para la proyección de movimiento.

8.1 Asignar reserva.

El sistema debe permitir buscar el militante y sus reservas. Se podrán planificar hasta dos reservas para cada cargo.

El sistema debe permitir buscar el militante y eliminar las reservas planificadas que seleccione.

9 Gestionar planificación de movimiento

9.1 Planificar movimiento.

El sistema debe permitir buscar el militante y mostrar un listado con los datos de sus reservas planificadas. Permitir seleccionar si será movido (sí o no), la causa de por qué será movido, la reserva que lo sustituirá y la fecha (mes) prevista para el movimiento.

9.2 Modificar planificación.

El sistema debe permitir buscar el militante y de modificar la planificación de movimiento realizada será movido (sí o no), la causa de por qué será movido, la reserva que lo sustituirá y la fecha (mes) prevista para el movimiento.

10 Realizar movimiento

El sistema debe permitir buscar el militante y mostrar la planificación de movimientos realizada. Permitir realizar el movimiento, actualizando la asignación de cargos de la estructura.

Módulo Reportes

11 Crear reportes

El sistema debe permitir obtener reportes de tablas tienen plantillas específicas y reportes de documentos los cuales tienen datos específicos.

11.1 Obtener reportes de tablas (RT)

Los reportes de tablas tienen plantillas específicas. El sistema debe dar la posibilidad de escoger el período del que se desea obtener el reporte. Para los reportes de tablas que se relacionan a continuación se indicarán como campos de selección:

Período del reporte: El sistema debe dar la posibilidad al usuario, de seleccionar el período en el cual desea obtener el reporte.

↳ Período Acumulativo mes inicio, mes final, año

↳ Mensual mes, año.

Capítulo 2: Descripción y análisis de la solución propuesta.

Estructura: La aplicación debe permitir al usuario seleccionar, la determinada estructura de la que quiere obtener la información.

11.2 Obtener reportes documentos (RD)

El sistema permitirá obtener reportes de documentos. Los documentos tienen datos específicos a mostrar e insertar.

12 Obtener RT según los movimientos de cuadros por causas

Recoge la información sobre los movimientos de los Cuadros de la UJC y las causas del movimiento. ([Ver formato del reporte en el Anexo 2](#)).

13 Obtener RT según la procedencia laboral, composición étnica, militancia y edad de los cuadros

Recoge la información relacionada con la procedencia laboral, composición étnica, militancia y edad de los cuadros. ([Ver formato del reporte en el Anexo 3](#)).

14 Obtener RT según la antigüedad como cuadro y la antigüedad en el cargo de todos los cuadros

Recoge la información relacionada con la antigüedad como cuadro y en el cargo de todos los cuadros del país. ([Ver formato del reporte en el Anexo 4](#))

15 Obtener RT según la información cultural y política de los cuadros de la UJC

Recoge la información cultural y política de los cuadros de la UJC. ([Ver formato del reporte en el Anexo 5](#))

16 Obtener RT según la información de las plazas movidas por cargos

Recoge la información de las plazas movidas por cargos. ([Ver formato del reporte en el Anexo 6](#))

17 Obtener RT según el listado de los movimientos realizados en el mes en curso

Recoge el listado de los movimientos realizados en el mes en curso. ([Ver formato del reporte en el Anexo 7](#))

18 Obtener RT según el listado de las plazas vacantes y los cuadros sin trabajar

Recoge el listado de las plazas vacantes y los cuadros sin trabajar con los datos correspondientes. ([Ver formato del reporte en el Anexo 8](#))

19 Obtener RT según el cumplimiento de la proyección de movimientos en el año

Recoge el cumplimiento de la proyección de movimientos en el año. ([Ver formato del reporte en el Anexo 9](#))

20 Obtener RT proyección de movimientos según plantilla de cuadros

Contendrá los datos de los Cuadros de la UJC que ocupan los diferentes cargos en las diferentes estructuras y los datos de las reservas propuestas. ([Ver formato del reporte en el Anexo 10](#))

Capítulo 2: Descripción y análisis de la solución propuesta.

21 Obtener RT realización de movimiento con fecha prevista

Este reporte contiene los datos del cuadro que va a ser movido y de sus sustitutos en la diferente estructura. ([Ver formato del reporte en el Anexo 11](#))

22 Obtener RT de cuadros que pasan al PCC

Contiene los datos de los Cuadros de la UJC que pasarán al PCC en un año previsto. ([Ver formato del reporte en el Anexo 12](#))

23 Obtener RT de cuadros que pasaron al PCC

Contiene los datos de los Cuadros de la UJC que han pasado al PCC en un determinado año. ([Ver formato del reporte en el Anexo 13](#))

24 Obtener RD Certificado de Evaluación de Cuadro

El sistema debe permitir obtener los siguientes datos, que se encuentran almacenados en el expediente del militante, para crear el Certificado de Evaluación de Cuadro. Antes de imprimirse o exportarse, el sistema permitirá incorporarlo a un listado de documentos que contendrá el Cuadro UJC. ([Ver formato del reporte en el Anexo 14](#))

25 Obtener RD Plantilla de Cargos

El sistema debe permitir obtener la Plantilla de Cargos en los diferentes niveles de dirección, después de realizar la gestión de la plantilla. ([Ver formato del reporte en el Anexo 15](#))

26 Obtener RD Expediente de Datos Personales

El sistema debe permitir obtener el Expediente de Datos Personales tanto para una Reserva UJC, como para un Cuadro UJC. ([Ver formato del reporte en el Anexo 16](#))

27 Mostrar reportes

El sistema debe permitir visualizar los reportes antes de exportarlos o imprimirlos.

28 Exportar reportes.

El sistema debe permitir exportar los reportes luego de visualizado a formato .pdf.

29 Imprimir reportes

El sistema debe permitir imprimir los reportes cuando el usuario lo solicite.

Capítulo 2: Descripción y análisis de la solución propuesta.

2.6 Requisitos no funcionales

Apariencia o interfaz externa.

- El diseño de la interfaz será muy sencillo y fácil con pocos campos a seleccionar para proporcionar una buena interacción el usuario y el sistema y que además facilite el tiempo de respuesta debido a que se gestiona una gran cantidad de información
- La información que se debe brindar sólo debe estar relacionada con el tema a tratar para no conllevar al entretenimiento innecesario del usuario.
- Los botones utilizados deberán corresponderse con la función que realizan.

Usabilidad.

- Una vez terminado el software se deberá impartir a las personas relacionadas con los reportes y la proyección de movimiento de los cuadros de la UJC Nacional una capacitación de cómo se debe trabajar con la aplicación
- Las personas que tramiten con estos procesos deberán tener conocimientos básicos de computación.
- El software posee una excelente facilidad de uso por lo que el trabajo con los reportes y la proyección de movimiento será cómodo y factible.

Rendimiento.

La velocidad en el tiempo de respuesta y procesamiento de la información en el momento de generar el reporte y realizar cualquier movimiento debe ser mínima para lograr un producto eficiente, cumpliendo con la gestión de la información que se necesita.

- Se necesita un mayor rendimiento físico en el instante de exportar e imprimir el reporte debido al volumen de información.
- El sistema deberá garantizar la consistencia y disponibilidad de la información en todo momento requiriéndose un tiempo de recuperación mínimo.
- El sistema debe tener la posibilidad de usarse en computadoras que no tengan mucho rendimiento debido a que la UJC no posee equipamiento tecnológico muy avanzado.

Soporte.

- Realizar pruebas a la aplicación cuando se haya concluido.

Capítulo 2: Descripción y análisis de la solución propuesta.

- Proporcionar a la UJC Nacional todos los servicios de instalación y configuración de la aplicación.
- Proporcionarle mantenimiento en un espacio de tiempo de 3 meses.

Portabilidad.

- El sistema es multiplataforma.

Seguridad.

- Confidencialidad:
 - Definir el nivel de acceso según el rol que tendrá cada usuario de los diferentes departamentos para su interacción con el software.
- Disponibilidad:
 - Para acceder al sistema el usuario debe estar autenticado previamente.
- Integridad:
 - Una vez generado el reporte solo se exportará al formato .pdf para que no se modifique la información brindada.
- Como algoritmo de reducción criptográfica se utilizó MD5. La codificación del MD5 es de 128 bits representada típicamente como un número de 32 dígitos hexadecimal. Es un algoritmo asimétrico y ampliamente utilizado en el mundo de la seguridad.

Confiabilidad.

- El sistema de BD debe contar con soporte para la recuperación en el caso de fallos y errores.
- El software debe ser capaz de mantener la integridad de los datos.

Software

- Se requiere Sistema Operativo Windows 2000 o Superior.
- Máquina Virtual de Java.
- Microsoft Access 2003.

Hardware

- Se necesita como mínimo 256 de RAM.
- Se requiere además de una impresora que será utilizada para imprimir los reportes.

Capítulo 2: Descripción y análisis de la solución propuesta.

Restricciones en el diseño y la implementación

- Los nombres de los métodos deben ser lo más sencillo posible para un mejor entendimiento entre el equipo de desarrolladores.
- Los nombres de las clases y métodos deben empezar con letra mayúscula y no debe ser usado el guión bajo.

2.7 Diagramas de casos de uso del sistema. (Ver descripción en el Anexo 17)

Módulo Administración

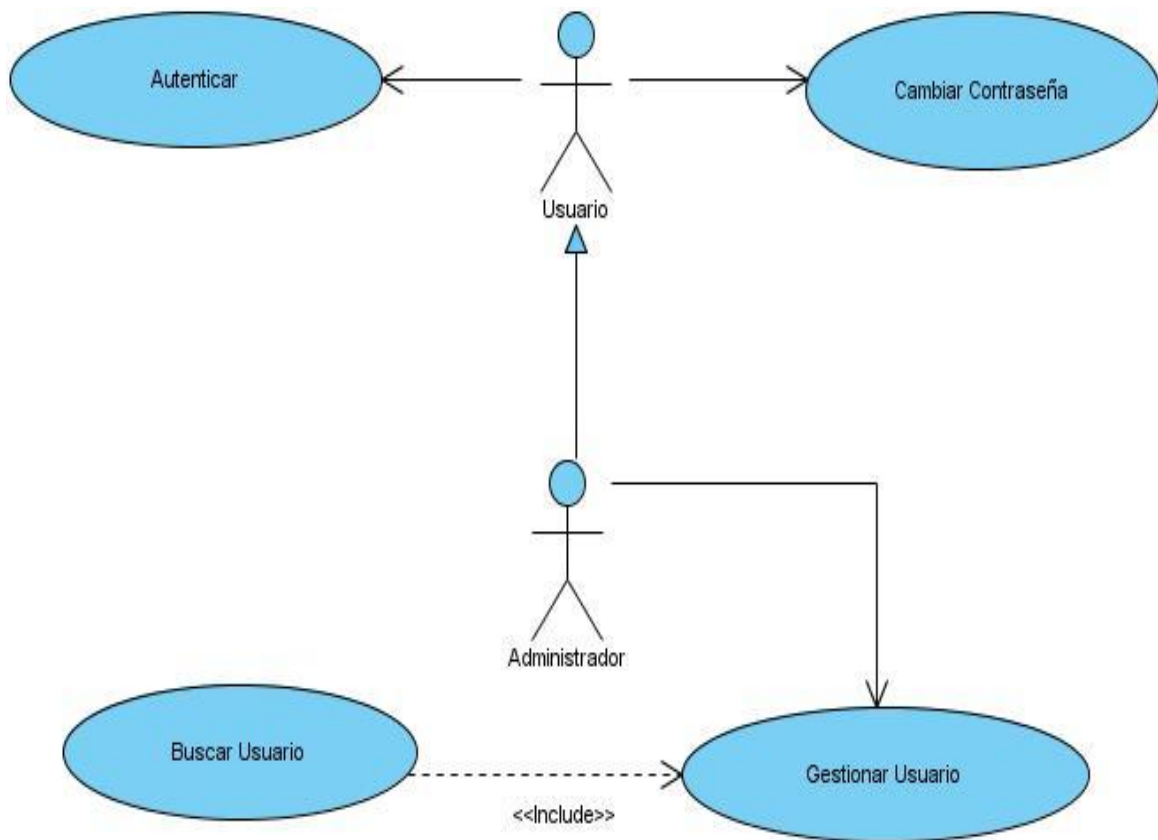


Figura 2: Diagrama de Casos de Uso Módulo Administración

Módulo Proyección de Movimientos.

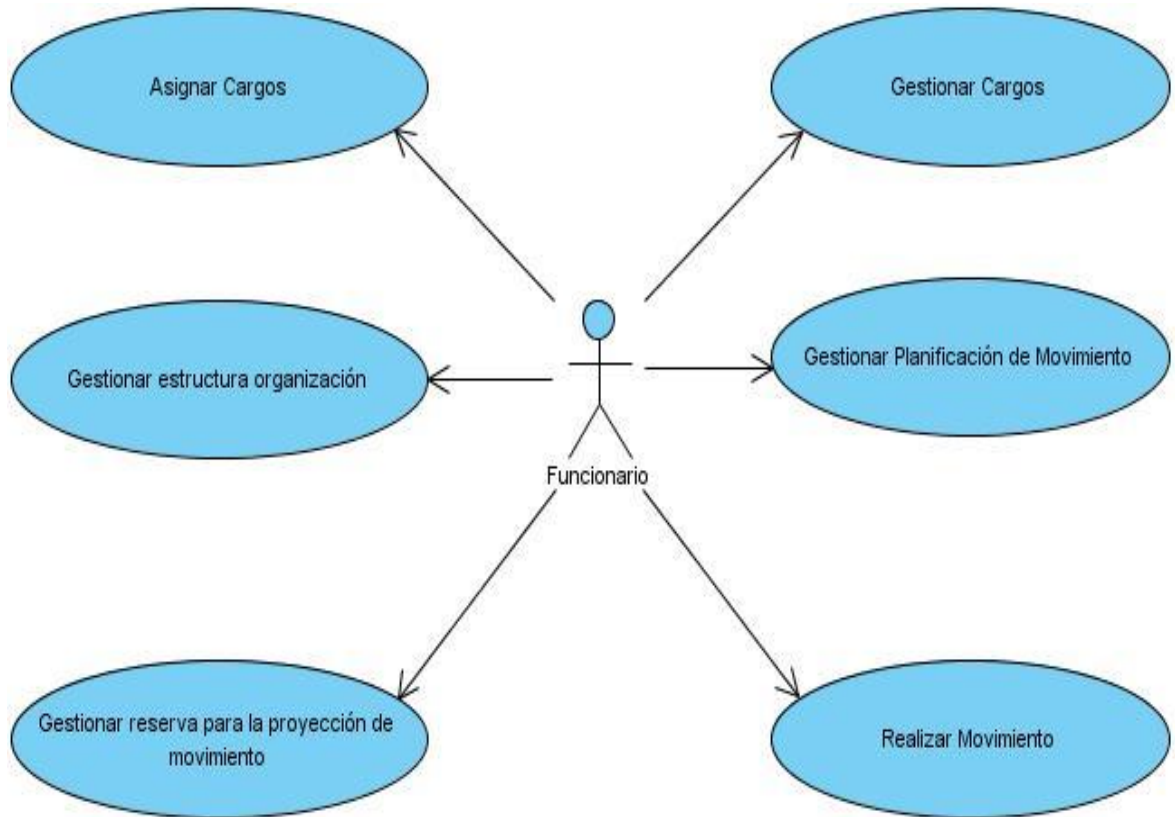


Figura 3: Diagrama de Casos de Uso Módulo Proyección de Movimientos

Módulo Reportes.

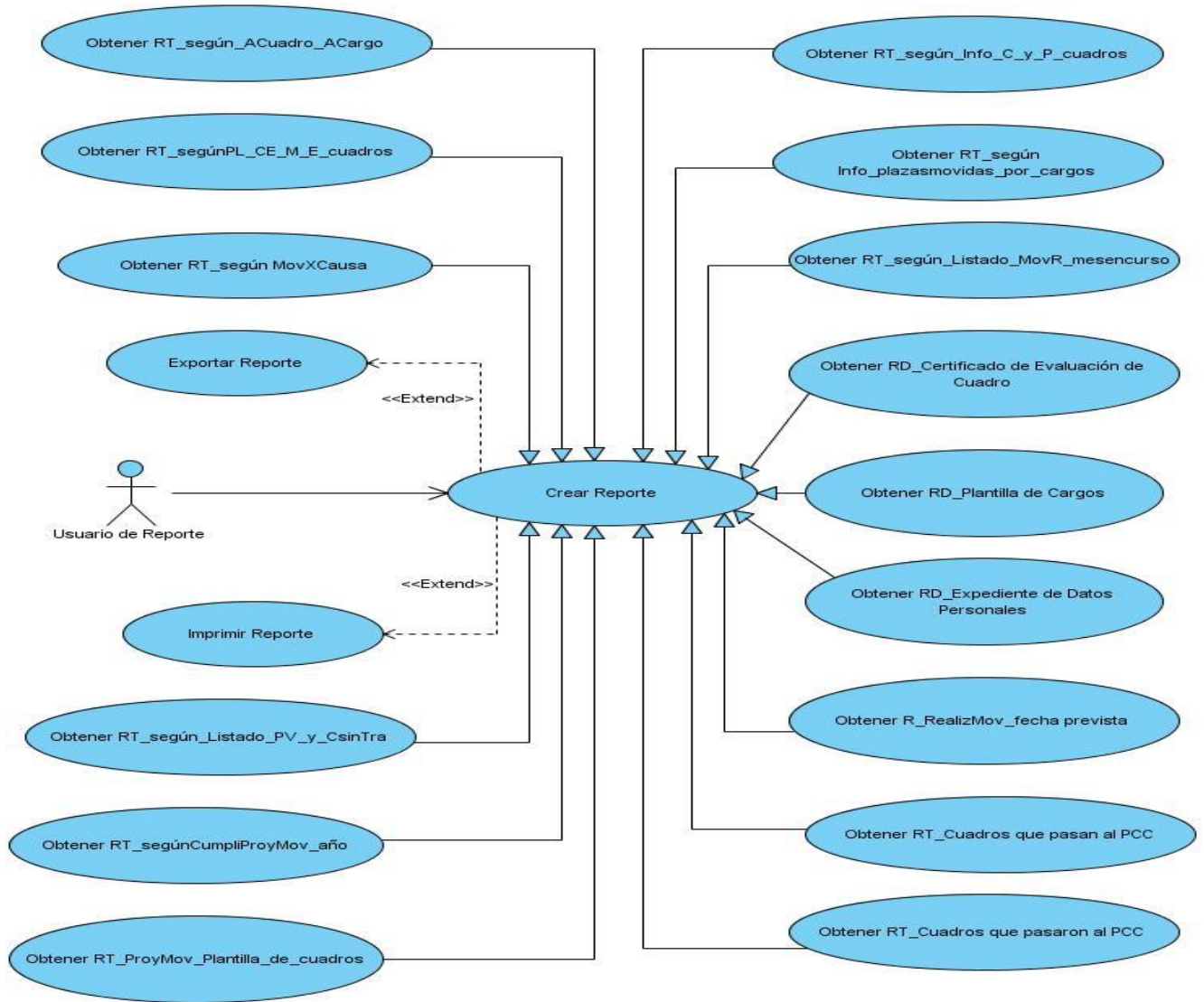


Figura 4: Diagrama de Casos de uso. Modulo Reporte

2.8 Diagrama de Componentes

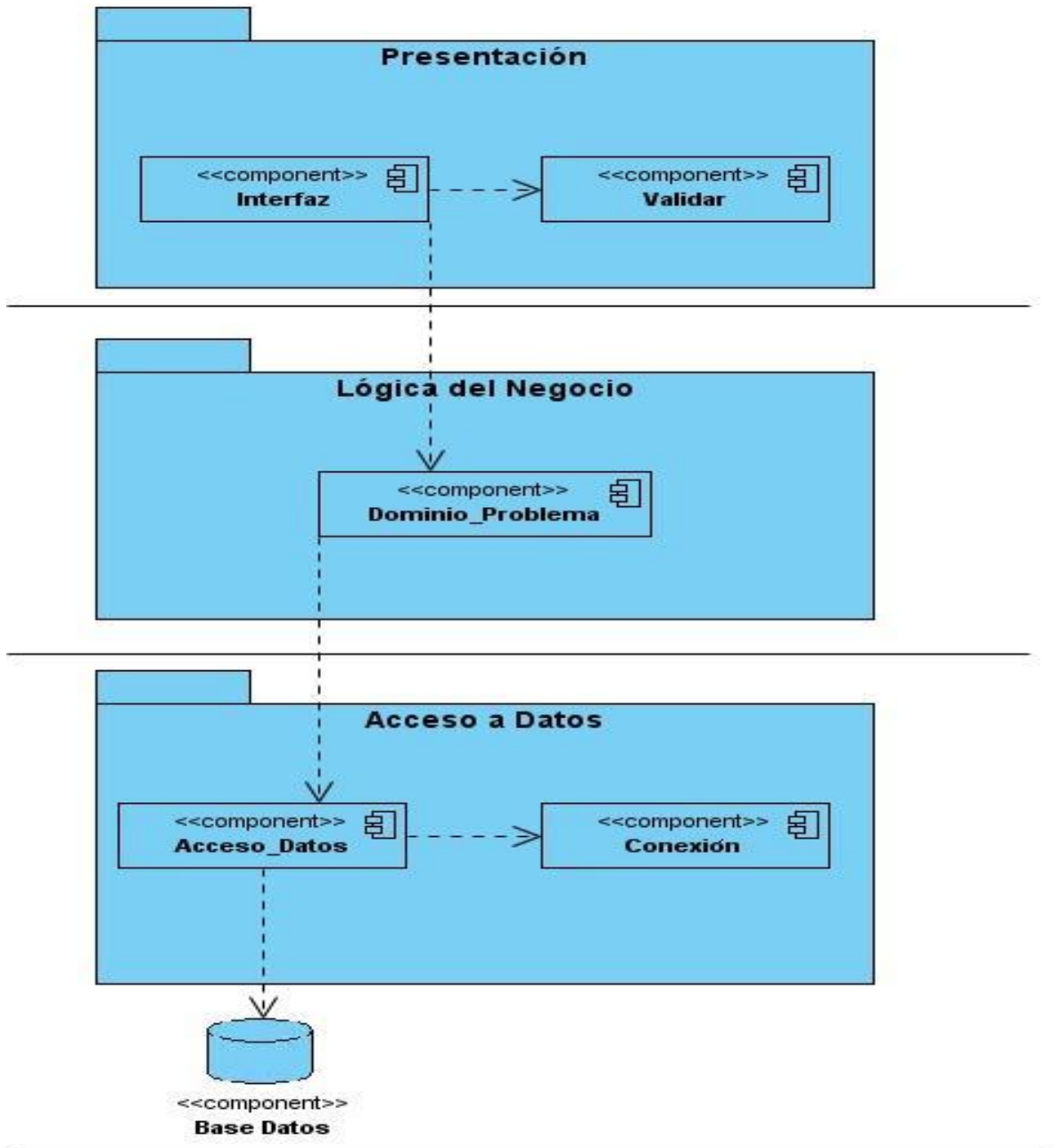


Figura 5: Diagrama de Componentes de los Submódulos de reporte y proyección de movimiento de la UJC Nacional

Capítulo 2: Descripción y análisis de la solución propuesta.

Paquete	Descripción
Presentación	El paquete contiene las clases Interfaz (vistas) del sistema.
Lógica del Negocio	El paquete contiene las clases del negocio.
Acceso a Datos	El paquete contiene las clases base que ofrece el trabajo con la capa de Acceso a Datos.

Tabla 2.50: Descripción del Diagrama de Componentes

2.9 Diagrama de despliegue

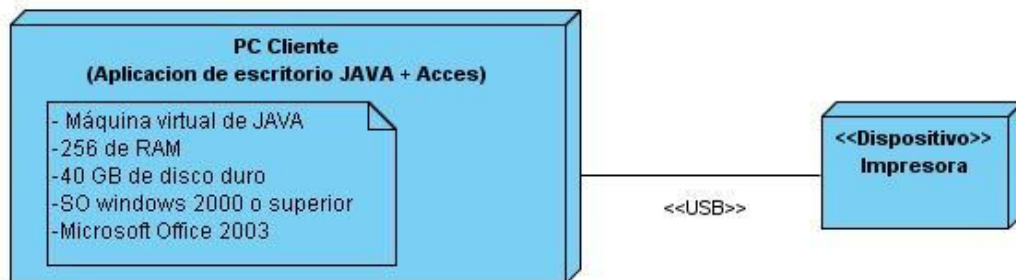


Figura 6: Diagrama de Despliegue. Submódulos de reportes y proyección de movimientos de la UJC Nacional

2.10 Conclusiones

Se concluye que:

- Después de realizado el análisis del diseño propuesto por los analistas se hicieron algunas modificaciones y aportes necesarios en el proceso de implementación para obtener los resultados deseados, como son el cambio de gestor de base datos DB4O por Access, JasperReports por DynamicJasper y la realización de un nuevo diseño de base datos logrando así establecer una línea base sencilla para la implementación de los submódulos.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

3.1 Introducción

Un producto de software no puede alcanzar la calidad requerida sin pasar primero por un proceso estricto de prueba y refinamiento. Cada prueba tiene su estrategia y propósito, estas no aseguran la ausencia de defectos; solo pueden demostrar que existen defectos en el software.

El objetivo de los casos de prueba es forzar al máximo el sistema en los puntos críticos para encontrar fallos y detectar defectos. Los tipos de prueba que se realicen deben ser los que mejor se adapten al sistema que se va a probar. Para esto se debe tener en cuenta el lenguaje de programación, el proceso de desarrollo, las características de los desarrolladores, el tipo de funcionalidad que se implementa, la plataforma en que se ejecutan los procesos, los errores más importantes, si la aplicación es de escritorio o web, si realiza conexiones a bases de datos entre otras observaciones.

3.2 Automatización de pruebas

El tener el proceso de prueba automatizado permite ilustrar el funcionamiento de la aplicación y al realizar un gran número de estas. Se puede mostrar la salida que produce la aplicación para una serie de entradas de prueba, brindando la posibilidad de entender el propósito de cada método. El proceso de automatización de pruebas es la parte del ciclo de calidad en la que el software de automatización es utilizado para controlar la ejecución de pruebas, comparación de resultados, preparación de precondiciones y realización de informes.

3.3 Descripción Preliminar del Modelo de Pruebas

El modelo de pruebas permite principalmente, probar los componentes ejecutables del modelo de interacción. También se puede probar aspectos específicos como, la interfaz. Contiene una gran cantidad de casos de prueba, procedimientos de prueba y componentes de prueba. Entre los métodos de prueba se puede distinguir las llamadas pruebas de “caja negra”.

Las pruebas de caja negra verifican el resultado de la interacción entre los usuarios y el sistema, comprobando que se cumplan las precondiciones y poscondiciones especificadas para cada caso de uso siguiendo la secuencia de acciones previstas para el mismo.

3.4 Pruebas de caja negra.

Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro.

Las pruebas de caja negra están especialmente indicadas en aquellos módulos que tienen interfaz gráfica, para la interacción usuario sistema en sentido general: teclado, pantalla, ficheros, canales de comunicaciones.

Las pruebas de caja negra se apoyan en la especificación de requisitos del módulo. De hecho, se habla de "cobertura de especificación" para dar una medida del número de requisitos que se han probado. Es fácil obtener coberturas del 100% en módulos internos, aunque puede ser más laborioso en módulos con interfaz al exterior. En cualquier caso, es muy recomendable conseguir una alta cobertura en esta línea.



Figura 8: Prueba de caja negra

3.4.1 Partición equivalente

Se realizaron pruebas de caja negra a cada caso de uso. La técnica seleccionada es la de partición equivalente. Para realizar esta se divide el dominio de entrada de un programa en clases de datos que tengan algo en común, de ahí derivan clases de prueba y por lo tanto también derivan clases de errores, de esta forma no hay necesidad de ejecutar muchas pruebas para encontrar errores genéricos. Se evalúan las clases de equivalencias posibles para una condición de entrada.

- Clases de equivalencias: Son los conjuntos de estados (válidos o no) para las condiciones de entrada.
- Condiciones de entrada: Valor numérico, rango de valores, condición booleana (si o no).

Capítulo 3: Validación de la solución propuesta.

- Si una condición de entrada especifica un rango, se propone definir una clase de equivalencia valida y 2 inválidas.
- Si una condición de entrada especifica un número, un valor, se define una clase de equivalencia igual al caso anterior, (1 valida y 2 inválidas).
- Si una condición de entrada especifica un miembro de un conjunto, se define una condición valida y una inválida.
- Si es booleana, se define una clase de equivalencia igual al caso anterior (1 valida y 1 inválida).

3.6 Aplicación de pruebas de caja negra

Descripción de los casos de prueba. (Ver resultado de las pruebas Anexo 18)

Caso de uso: Autenticar usuario.

ESCENARIO	ENTRADA	RESULTADO	CONDICIONES
Autenticar un usuario correctamente.	-Se ejecuta la aplicación. -Llenar todos los campos - Seleccionar "Aceptar".	-Se autentica un nuevo usuario al sistema. - Se accede al sistema.	-El usuario tiene que estar previamente registrado
Autenticar a un usuario dejando campos en blanco.	- Se ejecuta la aplicación. - Deja algún campo en blanco. - Seleccionar "Aceptar"	- Muestra un mensaje de error "Debe llenar todos los campos" -No se autentica.	-El usuario deja campos en blanco
Autenticar un usuario pasando datos incorrectos	- Se ejecuta la aplicación. -Llenar los campos pasando datos incorrectos. - Seleccionar "Aceptar".	- Muestra un mensaje de error "Debe llenar todos los campos correctamente"	-El usuario introduce datos incorrectos.

Capítulo 3: Validación de la solución propuesta.

	-Seleccionar "Aceptar"	-No se autentica.	
Cancelar la operación	- Se ejecuta la aplicación - Llena los campos. - Seleccionar "Cancelar".	No se autentica.	-El usuario desea cancelar la operación.

Tabla 3.1: Descripción del Caso de Prueba: Caso de Uso Autenticar Usuario

Caso de uso: Cambiar Contraseña.

ESCENARIO	ENTRADA	RESULTADO	CONDICIONES
Cambiar Contraseña correctamente	- Llenar todos los campos -Seleccionar "Aceptar".	-Muestra un mensaje indicando que su contraseña fue cambiada con éxito	-El usuario debe estar autenticado
Cambiar la contraseña dejando campos en blanco.	- Deja algún campo en blanco. -Seleccionar "Aceptar"	- Muestra un mensaje de error indicando que debe llenar todos los campos.	-El usuario debe estar autenticado -El usuario deja campos en blanco
Cambiar contraseña pasando datos incorrectos	- Llenar los campos pasando datos incorrectos. - Seleccionar "Aceptar".	-Se muestra un mensaje indicando que debe llenar todos los campos correctamente.	-El usuario debe estar autenticado -El usuario introduce datos incorrectos
Cancelar la operación	-Selecciona "Cancelar".	-Se muestra la interfaz principal.	-El usuario debe estar autenticado -El usuario desea cancelar la operación.

Tabla 3.2: Descripción del Caso de Prueba: Caso de Uso Cambiar Contraseña

Capítulo 3: Validación de la solución propuesta.

Caso de uso: Buscar usuario.

ESCENARIO	ENTRADA	RESULTADO	CONDICIONES
Buscar usuario correctamente	<ul style="list-style-type: none"> - Selección un tipo de rol -Selecciona "Buscar". -Se muestran los datos del usuario. 	Se busca un usuario en el sistema.	-El usuario tiene que estar autenticado.
Buscar usuario pasando el criterio incorrecto.	<ul style="list-style-type: none"> - Selección un tipo de rol erróneamente. -Selecciona "Buscar". 	<ul style="list-style-type: none"> -No se busca el usuario en el sistema. - Muestra un mensaje indicando que el criterio de búsqueda introducido no es correcto. 	-El usuario tiene que estar autenticado.
Buscar usuario sin pasar el criterio de búsqueda.	<ul style="list-style-type: none"> -No selección un tipo de rol -Selecciona "Buscar". 	<ul style="list-style-type: none"> No se busca un usuario en el sistema. - Muestra un mensaje de error indicando "Insertar el número del carnet de identidad" 	<ul style="list-style-type: none"> -El usuario tiene que estar autenticado. -El usuario no selecciona el rol.
Cancelar la operación.	<ul style="list-style-type: none"> -Se selecciona la opción "Cancelar". 	-Se muestra la interfaz principal.	<ul style="list-style-type: none"> --El usuario debe estar previamente autenticado -El usuario decide cancelar la operación.

Tabla 3.3: Descripción del Caso de Prueba: Caso de Uso Buscar Usuario

Capítulo 3: Validación de la solución propuesta.

Caso de uso: Gestionar Usuario.

ESCENARIO	ENTRADA	RESULTADO	CONDICIONES
Adicionar un usuario correctamente	<ul style="list-style-type: none"> -Se selecciona la opción "Adicionar". - Llenar todos los campos - Seleccionar "Adicionar". 	<ul style="list-style-type: none"> - Muestra un mensaje indicando que el usuario fue adicionado con éxito. - Se adiciona un usuario al sistema 	<ul style="list-style-type: none"> -El usuario debe estar previamente autenticado.
Adicionar un usuario dejando campos en blanco	<ul style="list-style-type: none"> -Se selecciona la opción "Adicionar". - Se dejan campos en blanco. -Seleccionar "Adicionar" 	<ul style="list-style-type: none"> - Muestra un mensaje de error indicando que debe llenar todos los campos. 	<ul style="list-style-type: none"> -El usuario debe estar previamente autenticado - El usuario deja campos en blanco.
Adicionar un usuario pasando datos incorrectos	<ul style="list-style-type: none"> -Se selecciona la opción "Adicionar". - Llenar los campos pasando datos incorrectos. -Seleccionar "Aceptar" 	<ul style="list-style-type: none"> -Se muestra un mensaje indicando que debe llenar todos los campos correctamente. 	<ul style="list-style-type: none"> -El usuario debe estar previamente autenticado - El usuario introduce datos incorrectos
Modificar un usuario correctamente	<ul style="list-style-type: none"> -Se busca el usuario -Se selecciona el usuario -Se selecciona la opción "Modificar" - Se cambian los datos deseados. - Seleccionar "Aceptar." 	<ul style="list-style-type: none"> - Muestra un mensaje indicando que el usuario fue modificado con éxito. -Se modifica el usuario en el sistema. 	<ul style="list-style-type: none"> -El usuario debe estar previamente autenticado.
Modificar un usuario pasando datos incorrectos o dejando campos en blanco.	<ul style="list-style-type: none"> -Se escoge un usuario. -Se selecciona la opción "Modificar" - Se cambian los datos deseados pasando datos incorrectos o dejando campos vacíos. -Seleccionar "Aceptar." 	<ul style="list-style-type: none"> Se muestra un mensaje indicando que debe llenar todos los campos correctamente. 	<ul style="list-style-type: none"> -El usuario debe estar previamente autenticado. - El usuario introduce datos incorrectos o deja campos en blanco.

Capítulo 3: Validación de la solución propuesta.

	-Seleccionar "Aceptar"		
Eliminar un usuario correctamente	<ul style="list-style-type: none"> -Se busca el usuario. -Se selecciona el usuario. -Se selecciona la opción "Eliminar". -Muestra un mensaje de comprobación: " Seguro que desea eliminar el usuario" -Seleccionar la opción "Aceptar" 	<ul style="list-style-type: none"> -Muestra un mensaje indicando que se eliminó el usuario. - Se elimina un usuario del sistema. 	-El usuario debe estar previamente autenticado
Cancelar la operación.	-Se selecciona la opción "Cancelar".	-Se muestra la interfaz principal	<ul style="list-style-type: none"> -El usuario debe estar previamente autenticado - El usuario decide cancelar la operación.

Tabla 3.4: Descripción del Caso de Prueba: Caso de Uso Gestionar Usuario

Caso de uso: Asignar Cargos.

ESCENARIO	ENTRADA	RESULTADO	CONDICIONES
Asignar Cargo correctamente	<ul style="list-style-type: none"> -Se selecciona la opción "Planificación" - Se selecciona la opción "Asignar Cargo" - Se llenan todos los campos. - Seleccionar "Asignar" 	<ul style="list-style-type: none"> -Muestra un mensaje indicando que el cargo fue asignado correctamente. - Se adiciona un nuevo cargo al sistema. 	- Es usuario tiene que estar autenticado.
Asignar Cargo dejando campos en blanco.	<ul style="list-style-type: none"> -Se selecciona la opción "Planificación" - Se selecciona la opción "Asignar Cargo" - Se dejan campos vacíos. - Seleccionar "Asignar" 	<ul style="list-style-type: none"> -Muestra un mensaje indicando que debe llenar todos los campos. 	<ul style="list-style-type: none"> - Es usuario tiene que estar autenticado. -El usuario deja campos en blanco.
Asignar Cargo introduciendo datos	Se selecciona la opción	-Muestra un mensaje indicando	- Es usuario tiene que estar autenticado.

Capítulo 3: Validación de la solución propuesta.

incorrectos.	<p>“Planificación”</p> <ul style="list-style-type: none"> - Se selecciona la opción “Asignar Cargo” - Se introducen campos incorrectos. - Seleccionar “Asignar”. 	que debe llenar todos los campos correctamente.	-El usuario introduce datos incorrectos.
Cancelar la operación	<p>Se selecciona la opción “Planificación”</p> <ul style="list-style-type: none"> - Se selecciona la opción “Asignar Cargo” - Seleccionar “Cancelar”. 	-Se muestra la interfaz principal.	<ul style="list-style-type: none"> - Es usuario tiene que estar autenticado. -El usuario decide cancelar la operación.

Tabla 3.5: Descripción del Caso de Prueba: Caso de Uso Asignar Cargo

Caso de uso: Gestionar Estructura.

ESCENARIO	ENTRADA	RESULTADO	CONDICIONES
Crear una estructura correctamente	<ul style="list-style-type: none"> -Se selecciona la opción “Planificación” - Se selecciona la opción “Gestionar Estructura” - Se selecciona la opción “Crear” - Se llenan todos los campos. - Seleccionar “Crear” 	<ul style="list-style-type: none"> -Muestra un mensaje indicando que la estructura fue creada satisfactoriamente. - Se crea una nueva estructura 	-El usuario tiene que estar autenticado
Crear una nueva estructura dejando campos en blanco.	<ul style="list-style-type: none"> Se selecciona la opción “Planificación” - Se selecciona la opción “Gestionar Estructura” - Se selecciona la opción “Crear” - Se dejan campos vacíos. - Seleccionar “Crear” 	<ul style="list-style-type: none"> -Muestra un mensaje indicando que debe llenar todos los campos. 	<ul style="list-style-type: none"> -El usuario tiene que estar autenticado -El usuario deja campos en blanco.

Capítulo 3: Validación de la solución propuesta.

<p>Crear una nueva estructura pasando datos incorrectos.</p>	<p>Se selecciona la opción "Planificación"</p> <ul style="list-style-type: none"> - Se selecciona la opción "Gestionar Estructura" - Se selecciona la opción "Crear" - Se dejan campos vacíos. - Seleccionar "Crear" 	<p>-Muestra un mensaje indicando que debe llenar todos los campos correctamente.</p>	<p>-El usuario tiene que estar autenticado</p> <p>-El usuario introduce datos incorrectos</p>
<p>Modificar una estructura correctamente.</p>	<p>Se selecciona la opción "Planificación"</p> <ul style="list-style-type: none"> - Se selecciona la opción "Gestionar Estructura" - Se selecciona la opción "Modificar" <p>-Se llenan todos los campos.</p> <p>- Seleccionar "Modificar"</p>	<p>-Muestra un mensaje indicando que se modificó la estructura correctamente.</p>	<p>-El usuario tiene que estar autenticado</p>
<p>Modificar estructura pasando datos incorrectos o dejar campos en blanco.</p>	<p>Se selecciona la opción "Planificación"</p> <ul style="list-style-type: none"> - Se selecciona la opción "Gestionar Estructura" - Se selecciona la opción "Modificar" <p>-Se introducen datos incorrectos o se dejan datos en blanco.</p> <p>- Seleccionar "Modificar"</p>	<p>-Muestra un mensaje indicando que debe llenar todos los campos correctamente.</p>	<p>-El usuario tiene que estar autenticado</p> <p>-El usuario deja campos en blanco o introduce datos incorrectos</p>
<p>Listar estructura correctamente</p>	<ul style="list-style-type: none"> -Se selecciona la opción "Planificación" - Se selecciona la opción "Gestionar Estructura" - Se selecciona la opción 	<p>-Muestra las estructuras en una tabla de datos</p>	<p>El usuario tiene que estar autenticado</p>

Capítulo 3: Validación de la solución propuesta.

	<p>“Listar”</p> <ul style="list-style-type: none"> -Se selecciona el tipo de comité. - Seleccionar “Listar” 		
Cancelar la operación.	<ul style="list-style-type: none"> -Se selecciona la opción “Planificación” - Se selecciona la opción “Gestionar Estructura” - Se selecciona la opción “Listar” -Seleccionar “Cancelar” 	-Se muestra la interfaz principal.	<ul style="list-style-type: none"> -El usuario tiene que estar autenticado -El usuario decide cancelar la operación

Tabla 3.6: Descripción del Caso de Prueba: Caso de Uso Gestionar Estructura

Caso de uso: Gestionar cargo.

ESCENARIO	ENTRADA	RESULTADO	CONDICIONES
Crear estructura de cargo correctamente	<ul style="list-style-type: none"> -Se selecciona la opción “Planificación” - Se selecciona la opción “Gestionar Cargo” - Se selecciona la opción “Crear” - Se llenan todos los campos correctamente. - Seleccionar “Crear” 	<p>Muestra un mensaje indicando que la estructura fue creada satisfactoriamente.</p> <p>-Se crea la estructura correctamente.</p>	-El usuario tiene que estar autenticado.
Crear una nueva estructura de cargo dejando campos en blanco.	<ul style="list-style-type: none"> -Se selecciona la opción “Planificación” - Se selecciona la opción “Gestionar Cargo” - Se selecciona la opción 	<p>Muestra un mensaje indicando que debe seleccionar todos los campos.</p>	<ul style="list-style-type: none"> -El usuario tiene que estar autenticado. - El usuario debe dejar campos en blanco.

Capítulo 3: Validación de la solución propuesta.

	<p>“Crear“</p> <ul style="list-style-type: none"> - Se dejan campos vacíos. - Seleccionar “Crear“ 	<p>-No se crea la estructura.</p>	
<p>Crear una nueva estructura de cargo pasando datos incorrectos.</p>	<ul style="list-style-type: none"> -Se selecciona la opción “Planificación“ - Se selecciona la opción “Gestionar Cargo“ - Se selecciona la opción “Crear“ - Se dejan campos vacíos. - Seleccionar “Crear“. 	<p>Muestra un mensaje indicando que debe llenar todos los campos correctamente.</p> <p>-No se crea la estructura.</p>	<p>-El usuario tiene que estar autenticado.</p> <p>-El usuario debe introducir datos incorrectos.</p>
<p>Modificar una estructura de cargo correctamente</p>	<ul style="list-style-type: none"> -Se selecciona la opción “Planificación“ - Se selecciona la opción “Gestionar Cargo“ - Se selecciona la opción “Modificar“ -Se llenan todos los campos. - Seleccionar “Modificar“ 	<p>-Muestra un mensaje indicando que se modificó la estructura correctamente.</p> <p>-Se modifica la estructura correctamente.</p>	<p>-El usuario tiene que estar autenticado.</p>
<p>Modificar estructura de cargo dejando datos en blanco.</p>	<ul style="list-style-type: none"> -Se selecciona la opción “Planificación“ - Se selecciona la opción “Gestionar Cargo“ - Se selecciona la opción “Modificar“ -No se selecciona todos los 	<p>-Muestra un mensaje indicando que debe llenar los campos.</p>	<p>-El usuario tiene que estar autenticado.</p> <p>-El usuario debe introducir datos incorrectos.</p>

Capítulo 3: Validación de la solución propuesta.

	<p>datos.</p> <ul style="list-style-type: none"> - Seleccionar "Modificar" 		
Listar estructura de cargo correctamente	<ul style="list-style-type: none"> -Se selecciona la opción "Planificación" - Se selecciona la opción "Gestionar Cargo" - Se selecciona la opción "Listar" -Se selecciona el tipo de comité. - Seleccionar "Listar". 	Muestra las estructuras en una tabla.	-El usuario tiene que estar autenticado.
Listar estructura de cargo sin seleccionar el tipo de comité	<ul style="list-style-type: none"> -Se selecciona la opción "Planificación" - Se selecciona la opción "Gestionar Cargo" - Se selecciona la opción "Listar" -No selecciona el tipo de comité. - Seleccionar "Listar" 	-No se listan las estructuras.	<p>-El usuario tiene que estar autenticado.</p> <p>-El usuario no selecciona el tipo de comité.</p>

Capítulo 3: Validación de la solución propuesta.

Cancelar operación.	la	<ul style="list-style-type: none"> -Se selecciona la opción "Planificación" - Se selecciona la opción "Gestionar Cargo" - Se selecciona la opción Cancelar 	- Se muestra la interfaz principal.	-El usuario tiene que estar autenticado.
---------------------	----	---	-------------------------------------	--

Tabla 3.7: Descripción del Caso de Prueba: Caso de Uso Gestionar Cargo

Caso de uso: Realizar Movimiento

ESCENARIO	ENTRADA	RESULTADO	CONDICIONES
Realizar movimiento correctamente	<ul style="list-style-type: none"> -Se selecciona la opción "Planificación" - Se selecciona la opción "Realizar Movimiento" - Se selecciona el tipo de comité -Se selecciona el período. - Seleccionar "Aceptar" -Se selecciona uno de los movimientos previstos. -Se selecciona "Realizar Movimiento" 	-Muestra un mensaje indicando que fue realizado correctamente el movimiento	-El usuario tiene que estar autenticado.
Realizar movimiento dejando campos en blanco.	<ul style="list-style-type: none"> -Se selecciona la opción "Planificación" - Se selecciona la opción "Realizar Movimiento" - Se deja algún campo en blanco. - Seleccionar "Aceptar" -Se selecciona uno de los 	-Muestra un mensaje indicando que debe llenar todos los campos.	-El usuario tiene que estar autenticado -Se dejan campos en blanco.

Capítulo 3: Validación de la solución propuesta.

	<p>movimientos previstos.</p> <p>-Se selecciona “Realizar Movimiento”</p>		
Realizar movimiento introduciendo datos incorrectos.	<p>-Se selecciona la opción “Planificación”</p> <p>- Se selecciona la opción “Realizar Movimiento”</p> <p>- Se introducen datos incorrectos</p> <p>- Seleccionar Aceptar“</p> <p>-Se selecciona uno de los movimientos previstos.</p> <p>-Se selecciona “Realizar Movimiento”</p>	-Muestra un mensaje indicando que debe llenar todos los campos correctamente.	<p>-El usuario tiene que estar autenticado</p> <p>-Se introducen datos incorrectos</p>
Cancelar la operación	<p>-Se selecciona la opción “Planificación”</p> <p>- Se selecciona la opción “Realizar Movimiento”</p> <p>-Seleccionar cancelar</p>	-Se muestra la interfaz principal.	<p>-El usuario tiene que estar autenticado</p> <p>-El usuario decide cancelar la operación.</p>

Tabla 3.8: Descripción del Caso de Prueba: Caso de Uso Realizar Movimiento

Caso de Uso: Gestionar reserva.

ESCENARIO	ENTRADA	RESULTADO	CONDICIONES
Asignar reserva correctamente	<p>-Se selecciona la opción “Planificación”</p> <p>- Se selecciona la opción “Gestionar Reserva”</p> <p>- Se selecciona la opción “Asignar”</p> <p>- Se llenan todos los campos.</p> <p>- Seleccionar “Aceptar”</p>	<p>-Muestra un mensaje indicando que la reserva fue asignada correctamente</p> <p>-Se asigna la reserva</p>	-El usuario debe estar autenticado.

Capítulo 3: Validación de la solución propuesta.

Asignar reserva dejando campos en blanco	<ul style="list-style-type: none"> -Se selecciona la opción "Planificación" - Se selecciona la opción "Gestionar Reserva" - Se selecciona la opción "Asignar" - Se dejan campos en blanco. - Seleccionar "Aceptar" 	-Muestra un mensaje indicando que debe llenar todos los campos.	<ul style="list-style-type: none"> -El usuario debe estar autenticado. -Se dejan campos en blanco
Asignar reserva introduciendo datos incorrectos	<ul style="list-style-type: none"> -Se selecciona la opción "Planificación" - Se selecciona la opción "Gestionar Reserva" - Se selecciona la opción "Asignar" - Se llenan los campos con datos incorrectos. - Seleccionar "Aceptar" 	-Muestra un mensaje indicando que debe llenar todos los campos correctamente.	<ul style="list-style-type: none"> -El usuario debe estar autenticado. -Se introducen datos incorrectos
Eliminar reserva correctamente.	<ul style="list-style-type: none"> -Se selecciona la opción "Planificación" - Se selecciona la opción "Gestionar Reserva" - Se selecciona la opción "Eliminar" - Se llenan todos los campos. - Seleccionar "Eliminar" 	-Muestra un mensaje indicando que la reserva fue eliminada correctamente	-El usuario debe estar autenticado.
Eliminar reserva sin seleccionar la reserva	<ul style="list-style-type: none"> -Se selecciona la opción "Planificación" - Se selecciona la opción "Gestionar Reserva" - Se selecciona la opción "Eliminar" 	-Muestra un mensaje indicando que debe seleccionar la reserva que desea eliminar.	<ul style="list-style-type: none"> -El usuario debe estar autenticado. -El usuario no selecciona la reserva

Capítulo 3: Validación de la solución propuesta.

	<ul style="list-style-type: none"> - No se selecciona la reserva. - Seleccionar "Eliminar" 		
Cancelar la operación	<ul style="list-style-type: none"> -Se selecciona la opción "Planificación" - Se selecciona la opción "Gestionar Reserva" -Seleccionar "Cancelar" 	-Se muestra la interfaz principal	<ul style="list-style-type: none"> -El usuario debe estar autenticado. -Se decide cancelar la operación.

Tabla 3.9: Descripción del Caso de Prueba: Caso de Uso Gestionar Reserva

Caso de Uso: Gestionar Planificación de Movimiento.

ESCENARIO	ENTRADA	RESULTADO	CONDICIONES
Planificar movimiento correctamente	<ul style="list-style-type: none"> -Se selecciona la opción "Planificación" - Se selecciona la opción "Gestionar Movimiento" - Se selecciona la opción "Planificar o Modificar" - Se seleccionan todos los datos. - Seleccionar "Planificar Movimiento" -Se llenan todos los campos -Se selecciona "Adicionar" 	-Muestra un mensaje indicando que el movimiento fue realizado correctamente.	-El usuario tiene que estar autenticado.
Planificar movimiento dejando campos en blanco.	<ul style="list-style-type: none"> Se selecciona la opción "Planificación" - Se selecciona la opción "Gestionar Movimiento" - Se selecciona la opción "Planificar o Modificar" - Se dejan campos en blanco. 	-Muestra un mensaje indicando que debe llenar todos los campos.	-El usuario tiene que estar autenticado.

Capítulo 3: Validación de la solución propuesta.

	<ul style="list-style-type: none"> - Seleccionar "Planificar Movimiento" -Se selecciona "Adicionar" 		
Modificar planificación correctamente.	<ul style="list-style-type: none"> Se selecciona la opción "Planificación" - Se selecciona la opción "Gestionar Movimiento" - Se selecciona la opción "Planificar o Modificar" - Se seleccionan todos los datos. - Seleccionar "Actualizar Movimientos" -Se llenan todos los campos -Se selecciona "Actualizar" 	<ul style="list-style-type: none"> -Muestra un mensaje indicando que se modificó correctamente. 	<ul style="list-style-type: none"> - El usuario tiene que estar autenticado.
Modificar planificación dejando campos en blanco.	<ul style="list-style-type: none"> -Se selecciona la opción "Planificación" - Se selecciona la opción "Gestionar Movimiento" - Se selecciona la opción "Planificar o Modificar" - Seleccionar "Actualizar Movimientos" -Se dejan campos en blanco -Se selecciona "Actualizar" 	<ul style="list-style-type: none"> -Muestra un mensaje indicando que debe llenar todos los campos 	<ul style="list-style-type: none"> - El usuario tiene que estar autenticado.
Cancelar la operación	<ul style="list-style-type: none"> Se selecciona la opción "Planificación" - Se selecciona la opción "Planificar o Modificar" - Seleccionar "Cancelar" 	<ul style="list-style-type: none"> -Se muestra la interfaz principal 	<ul style="list-style-type: none"> -El usuario tiene que estar autenticado.

Tabla 3.10: Descripción del Caso de Prueba: Caso de Uso Gestionar Planificación de Movimiento

3.7 Conclusiones

Se concluye que:

- Los casos de prueba son el mecanismo usado para asegurar que el sistema cumpla con los requerimientos funcionales, asegurando así la calidad del software.
- Se realizaron las pruebas de caja negra al sistema. Las cuales fueron satisfactorias, ya que se detectaron inconformidades, contribuyendo así a la obtención de un producto con calidad.

Conclusiones Generales

Después de culminar con la implementación los Submódulos de reporte y proyección de movimiento de los cuadros de la UJC Nacional, se puede decir que se cumplió con el objetivo general propuesto y además se cumplieron los objetivos específicos propuestos.

- El sistema favorece la gestión de la información y la proyección de movimiento de los cuadros de la UJC Nacional, garantizando la centralización de la información, la agilización de los procesos, la reducción de los gastos en materiales, la disminución de los riesgos por pérdida y deterioro de los documentos y el aumento de la seguridad de la misma.
- El objetivo trazado en el diseño teórico – metodológico para el desarrollo del sistema fue cumplido, comprobándose la hipótesis como respuesta del problema a resolver que originó esta investigación, contribuyendo directamente a la gestión de la información y la proyección de movimiento de los cuadros de la UJC Nacional.
- Las tareas investigativas facilitaron una adecuada organización de la investigación.
- La utilización de los métodos teóricos y empíricos facilitaron conocer el estado de los procesos de obtención de la información y proyección de movimiento de los cuadros de la UJC.
- El sistema fue avalado por el cliente.

Recomendaciones

Los objetivos generales de este trabajo han sido logrados, pero a lo largo de su desarrollo, han ido surgiendo ideas que podrían implementarse en un futuro, de forma que se logre una aplicación más útil y efectiva, para lo cual se recomienda:

- Se recomienda para posteriores versiones, tomar en cuenta la sustitución del gestor de base datos Access por otro gestor que sea libre y que cumpla con los requisitos no funcionales propuestos.
- Si la UJC mejora las prestaciones de sus computadoras, se recomendaría la realización de una aplicación web para que así se pueda acceder a la información desde cualquier lugar del país.

Glosario de Términos

RD: Reporte de Documento.

RT: Reporte de Tabla.

Cuadros: el sustantivo cuadro, obedece a la condición de dirigente, administrativo, ejecutivo o gerente.

Sector: área institucional donde trabaja el militante.

Rama: área ocupacional donde trabaja el militante en una institución en particular.

Mérito: premios que se le otorgan a una persona en correspondencia con su comportamiento.

Categoría: definición usada para clasificar y definir un militante.

Militantes: persona que pertenece a una organización determinada en este caso a la UJC.

Tipo Miembro: clasificación de una persona que forma parte de un grupo, organización o institución.

Clasificador Ocupacional: clasificación de datos e informaciones sobre las ocupaciones que facilita un marco para el análisis, la agregación y la descripción de los contenidos del trabajo, así como un sistema de niveles y áreas para ordenar las ocupaciones en el mercado de trabajo.

Labor: trabajo que se realiza.

Vía: forma de salida de un ciudadano del país (legal o ilegal).

GNU GPL: Licencia Pública General de GNU o *GNU General Public License*, es una licencia creada por la Fundación del Software Libre a mediados de los 80, orientada a proteger la libre distribución, modificación y uso de software.

Política de Cuadros: es el grupo de acciones que permiten preparar a una o varias personas para que ante la sustitución o liberación de un cargo, el proceso no se detenga. Es la forma en que se educa y desarrolla a los miembros de las organizaciones, como dirigentes revolucionarios. Es la estructura que permite garantizar una acertada dirección del proceso revolucionario bajo cualquier circunstancia.

Bibliografía

1. Programación a medida para empresas. Tipos de aplicaciones que se pueden desarrollar. [Citado el: 5 de enero de 2010.] <http://www.haztuprograma.com/tiposAplicaciones.html#web..>
2. Welcome to NetBeans. [En línea] 25 de marzo de 2008. [Citado el: 20 de diciembre de 2009.] <http://www.netbeans.org/>.
3. Db4o::Native Java & .NET Open Source Object Database. [En línea] 2000-2009. [Citado el: 10 de febrero de 2010.] <http://www.db4o.com/>.
4. Programación en Java. [En línea] [Citado el: 10 de enero de 2010.] http://www.programacion.com/java/articulo/jap_persis_db4o/.
5. Sistema Gestor de base de datos SGBD. [En línea] 1 de noviembre de 2004. [Citado el: 10 de enero de 2010.] http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php.
6. Entorno de Desarrollo Integrado. [En línea] [Citado el: 10 de enero de 2010.] http://es.wikipedia.org/wiki/Ambiente_integrado_de_desarrollo.
7. JFreeChart. [En línea] 2005-2009. [Citado el: 15 de enero de 2010.] <http://www.jfree.org/jfreechart/>.
8. Metodologías de desarrollo de software-Metodología de desarrollo de software-Wikilearning. [En línea] [Citado el: 14 de diciembre de 2009.] http://www.wikilearning.com/curso_gratis/metodologias_de_desarrollo_de_software-metodologia_de_desarrollo_de_software/3617-1.
9. Metodologías de Desarrollo de Software. [En línea] [Citado el: 22 de octubre de 2009.] <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>.
10. CASE. Herramientas CASE. [En línea] [Citado el: 10 de enero de 2010.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.
11. Generador de Reportes. [En línea] [Citado el: 6 de diciembre de 2009.] <http://www.insoftweb.com/contapyme/modulos/herramientas/generadordereportes/default.htm>.
12. Sistema generador de reportes . [En línea] 2009. [Citado el: 6 de diciembre de 2009.] <http://siu.uaaan.mx/siiaa/index.php/estadisticas/41-acceso-al-sistema/78-sistema-generador-de-reportes.html>.

13. Lenguajes de programación, Herramientas de programación. [En línea] [Citado el: 6 de enero de 2010.] <http://www.lenguajes-de-programacion.com/herramientas-de-programacion.shtml>.
14. ¿Qué es NetBeans? [En línea] 2007. [Citado el: 02 de enero de 2010.] http://www.netbeans.org/index_es.html
15. Java finalmente es software libre. [En línea]. [Citado el: 2 de enero de 2010.] http://www.softwarelibre.net/java_finalmente_es_software_libre
16. Bases de datos Orientadas a Objetos DB4O. [En línea]. [Citado el: 25 de Enero de 2010.] www.mhproject.org/media/blogs/mhpenlaces/Interno/Presentaciones/.../BD.%20Orientada%20a%20Objetos%20-%20db4objects.ppt.
17. Tutorial Herramienta de diseño iReport de JasperReports. [En línea] [Citado el: 01 de Marzo de 2010.] www.imapax.com/descargas/jasper/iReport%20Tutorial%20v2_1.doc
18. Paradigm Visual. visual-paradigm.com/product/vpuml/. [En línea] Visual Paradigm. , 2008. [Citado el: 18 de Febrero de 2009.] <http://www.visualparadigm.com/product/vpuml/>.

Referencias Bibliográficas

1. Navarro E, Suria M, Górriz J.L, Orellana FJ. Nefrosoft HD 3.0. [En línea] [Citado el: 9 de 12 de 2009.] <http://www.nefrosoft.com/>
2. SAFYR. [En línea] [Citado el: 14 de 11 de 2009.] <http://www.websafyr.net>
3. E.V.A.,Ingeniería de Software,Conf Tema#1. [En línea] [Citado el: 5 de noviembre de 2009.] <http://teleformacion.uci.cu>
4. Visual Paradigm for UML (ME). [En línea] [Citado el: 30 de noviembre de 2009.] [http://www.freownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_1_4720_p/](http://www.freownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_1_4720_p/)
5. CASE. Herramientas CASE. [En línea] [Citado el: 22 de enero de 2010.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.
6. Sistema generador de reportes . [En línea] 2009. [Citado el: 6 de diciembre de 2009.] <http://siu.uaaan.mx/siiaa/index.php/estadisticas/41-acceso-al-sistema/78-sistema-generador-de-reportes.html>
7. Tutorial Herramienta de diseño iReport de JasperReports. [En línea] [Citado el: 14 de enero de 2010] www.imapax.com/descargas/jasper/iReport%20Tutorial%20v2_1.doc
8. Reportadores dinámicos.[En línea][Citado el: 26 de marzo 2010] <http://dynamicjasper.sourceforge.net/>
http://www.javahispano.org/contenidos/es/agrega_dinamismo_a_los_reportes_en_jasperreports_con_dynamicjasper_11/
9. Programación en Java. [En línea] [Citado el: 10 de febrero de 2010.]http://www.programacion.com/java/articulo/jap_persis_db4o/.
10. ¿Qué es NetBeans? [En línea][Citado el: 6 de enero de 2010.] http://www.netbeans.org/index_es.html
11. Db4o::Native Java & .NET Open Source Object Database. [En línea] 2000-2009. [Citado el: 10 de diciembre de 2009.] <http://www.db4o.com/>.