

**Universidad de las Ciencias Informáticas**

**“Facultad 4”**



**Título: Desarrollo del Componente Trabajador  
del Subsistema Capital Humano del sistema  
Cedrux**

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

**Autor (es):** Oilede Pérez Ramírez  
Yonisley García López

**Tutor (es):** Rubén Rodríguez Torres

**Cotutor:** Ariagna González Landeiro

**Junio del 2009**

*“El hombre debe transformarse al mismo tiempo que la producción progresa; no realizaríamos una tarea adecuada si fuéramos tan sólo productores de artículos, de materias primas y no fuéramos al mismo tiempo productores de hombres.”*

*Ernesto Che Guevara*

AGRADECIMIENTOS

**Oilede:**

*A Fidel y a la Revolución por brindarme la posibilidad de lograr mis metas.*

*A mi familia que siempre me ha apoyado en todas las etapas de mi vida y que nunca defraudaré.*

*A mi compañero de tesis Yonisley, más que un amigo y compañero, un hermano.*

*A mi tutor Ruben que siempre estuvo presente como jefe, y amigo incondicional.*

*A mi cotutora Ariagna por dedicarme el tiempo y apoyo necesario.*

*A Checho, sin tu ayuda esto no hubiera sido posible.*

*A todos mis amigos que han compartido conmigo durante todos estos años de estudio, esfuerzo y sacrificio.*

**Yonisley:**

*A Fidel y a la Revolución por brindarme la posibilidad de estar aquí hoy.*

*A mi familia, gracias por confiar en mí en todo momento sin dudar ni una sola vez, nunca le hubiera fallado.*

*A mi compañero de tesis Oilede, más que amigo y compañero, un hermano.*

*A mi tutor Rubén, gracias hermano por tu apoyo incondicional.*

*A mi cotutora Ariagna, muchas gracias por habernos apoyado en todo momento.*

*A mi novia, gracias mi amor, sin tu apoyo y amor constante esto no hubiera sido posible.*

*A todos mis amigos y compañeros que han compartido conmigo todos estos años de estudio y sacrificio.*

## AGRADECIMIENTOS

---

*A todos los que de una forma u otra me han ayudado a cumplir con una meta tan importante en mi vida.*

DEDICATORIA

***A Fidel, a la Revolución y a nuestros familiares.***

### RESUMEN

Los procesos de gestión de los trabajadores en las entidades empresariales y presupuestadas no se gestionan de la forma más eficiente dado que la existencia de los sistemas que están implantados no cumplen con las exigencias de las empresas cubanas, esto provoca que exista pérdida en la documentación de los movimientos de nómina de los trabajadores, descontrol en la plantilla de trabajadores en las entidades y la cantidad de documentación que se manejan de los mismos dificultan la operatividad del trabajo. Con el objetivo de mejorar los procesos referentes a los trabajadores en las entidades del territorio nacional se propone desarrollar un componente Trabajador que forma parte del Subsistema Capital Humano del sistema integral de gestión Cedrux.

Para el desarrollo del componente se utilizaron como herramientas de implementación, PHP en su versión 5.2, por el lado del servidor, Java Script y ExtJS por el lado del cliente, empleando además Zend Studio para Eclipse 6.0 como (IDE) de desarrollo. Como servidor web se utilizó Apache y gestor de base de datos PostgreSQL. Para la estructuración y construcción de la solución se utilizó el Modelo de desarrollo orientado a Componentes, se usó UML para la modelación del Análisis y el Diseño y la Notación para el Modelado de Procesos de Negocio (BPMN) para modelar el Negocio, aplicándose ambos mediante la herramienta Visual Paradigm for UML 6.4.

Por consiguiente se obtuvo como resultado una solución que permite realizar de manera integral la gestión de los procesos relacionados con los trabajadores para las entidades cubanas.

### PALABRAS CLAVE

[Capital Humano, proceso, gestión de los trabajadores, componente]

ÍNDICE

ÍNDICE.....	7
INTRODUCCIÓN.....	12
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....	15
Introducción .....	15
1.1 Conceptos Generales.....	15
1.1.1 Capital Humano.....	15
1.1.2 Trabajador.....	15
1.1.3 Normas para el Movimiento de la Fuerza de Trabajo .....	16
1.1.3.1 Movimiento de la Fuerza de Trabajo.....	16
1.1.3.2 Registro de trabajadores .....	18
1.1.3.3 Datos contables.....	19
1.1.4 Métodos para la identificación de requerimientos .....	19
1.1.5 Técnicas de validación de requerimientos .....	23
1.1.6 Patrones de diseño .....	23
1.1.7 Componente.....	24
1.2 Sistemas estudiados .....	25
1.3 Modelo de desarrollo de software orientado a componentes.....	30
1.4 Herramientas y tecnologías.....	30
1.4.1 Visual Paradigm .....	30
1.4.2 Lenguaje de Modelado .....	30
1.4.2.1 UML.....	30
1.4.2.2 Notación para el Modelado de Procesos de Negocio (BPMN) .....	31
1.4.3 Lenguajes de programación .....	31
1.4.3.1 AJAX .....	32
1.4.3.2 ExtJS 2.0.....	35
1.4.3.3 PHP.....	35
1.4.4 Ambiente de desarrollo integrado (IDE).....	35
1.4.4.1 Zend Studio para Eclipse.....	36
1.4.5 Framework .....	37
1.4.5.1 Zend Framework .....	37
1.4.5.2 Zend_Ext Framework .....	37
1.4.5.3 Doctrine Framework .....	38
1.4.6 Servidor Web .....	38
1.4.6.1 Apache.....	38
1.4.7 Sistema Gestor de Base de Datos (SGBD) .....	38
1.4.7.1 PostgreSQL.....	38
1.4.8 Herramientas de base de datos.....	39
1.4.8.1 EMS SQL Manager .....	39
1.4.9 Navegadores.....	39
1.4.9.1 Mozilla Firefox .....	39
1.4.10 Control de versiones.....	39
1.4.10.1 Subversion .....	39

## ÍNDICE

---

Conclusiones .....	41
<b>CAPÍTULO 2: MODELACIÓN DEL NEGOCIO Y ANÁLISIS.....</b>	<b>42</b>
Introducción .....	42
2.1 Modelo del negocio .....	42
2.2 Mapa de procesos.....	42
2.3 Procesos de negocio.....	43
2.3.1 Descripción del proceso Movimiento de la Fuerza de Trabajo.....	43
2.4 Modelo conceptual .....	45
2.5 Métodos utilizados para identificar los requerimientos.....	46
2.6 Identificación de los requerimientos funcionales.....	46
2.6.1 Requerimientos funcionales .....	46
2.7 Especificación de requerimientos .....	47
2.7.1 Requisito funcional Emitir movimiento de nómina.....	47
2.7.2 Requisito funcional Movimiento de altas.....	47
2.7.3 Requisito funcional Movimiento de reubicación .....	51
2.7.4 Requisito funcional Movimiento de bajas.....	54
2.7.5 Requisito funcional Actualizar datos contables.....	56
2.7.6 Requisito funcional del requisito Buscar trabajador .....	57
2.7.7 Requisito funcional Buscar fonéticamente un trabajador .....	59
2.7.8 Requisito funcional Filtrar datos de los trabajadores .....	60
2.7.9 Requisito funcional Mostrar registro de los trabajadores .....	61
2.8 Técnicas utilizadas para validar los requerimientos.....	62
Conclusiones .....	63
<b>CAPÍTULO 3: DISEÑO DEL SISTEMA.....</b>	<b>64</b>
Introducción .....	64
3.1 Modelo de diseño del sistema .....	64
3.1.1 Aplicación de los patrones.....	64
3.1.1.1 Patrón Arquitectónico Modelo Vista Controlador (MVC) .....	64
3.1.1.2 Patrones GRASP.....	66
3.1.1.3 Patrones de comportamiento.....	67
3.1.2 Diagramas de clases del diseño .....	68
3.2 Modelo de datos.....	68
3.2.1 Descripción de las tablas de la base de datos.....	69
3.3 Evaluación del modelo de diseño propuesto .....	77
3.3.1 Tamaño operacional de clase (TOC).....	78
3.3.1.1 Resultados del instrumento de evaluación de la métrica Tamaño Operacional de Clase (TOC).....	78
3.3.2 Relaciones entre clases (RC).....	81
3.3.2.2 Resultados del instrumento de evaluación de la métrica Relaciones entre Clases (RC).....	82
Conclusiones .....	86
<b>CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBAS .....</b>	<b>87</b>
Introducción .....	87
4.1 Modelo de implementación .....	87

## ÍNDICE

---

4.1.1 Diagrama de componentes.....	87
4.2 Pruebas de software .....	90
4.2.1 Pruebas de caja negra .....	91
4.2.2 Pruebas de caja blanca .....	91
4.2.2.1 Cálculo de la complejidad ciclomática a partir de un segmento de código .....	96
Conclusiones .....	102
CONCLUSIONES .....	103
RECOMENDACIONES.....	104
REFERENCIA BIBLIOGRÁFICA .....	105
BIBLIOGRAFÍA.....	107
ANEXOS.....	109
GLOSARIO.....	128

### ÍNDICE DE TABLAS

Tabla 1 Especificación del requisito Movimiento de altas.....	47
Tabla 2 Especificación del requisito Movimiento de reubicación .....	51
Tabla 3 Especificación del requisito Movimiento de bajas.....	54
Tabla 4 Especificación del requisito Actualizar datos contables .....	56
Tabla 5 Especificación del requisito Buscar trabajador .....	57
Tabla 6 Especificación del requisito Buscar fonéticamente un trabajador .....	59
Tabla 7 Especificación del requisito Filtrar datos de los trabajadores.....	60
Tabla 8 Especificación del requisito Mostrar registro de los trabajadores .....	61
Tabla 9 Descripción de la tabla dat_trabajador .....	69
Tabla 10 Descripción de la tabla dat_modelomovnomina .....	70
Tabla 11 Descripción de la tabla nom_motivomovnomina.....	71
Tabla 12 Descripción de la tabla nom_tipomovnomina .....	72
Tabla 13 Descripción de la tabla nom_tiporeubicacion .....	73
Tabla 14 Descripción de la tabla dat_datoscontables .....	73
Tabla 15 Descripción de la tabla nom_contrato .....	74
Tabla 16 Descripción de la tabla nom_sistemapago .....	74
Tabla 17 Descripción de la tabla dat_registropagosadicionalestrabajador .....	75
Tabla 18 Descripción de la tabla his_registropagosadicionalestrabajador.....	76

## ÍNDICE

---

Tabla 19 Tamaño operacional de clase (TOC).....	78
Tabla 20 Relaciones entre clases (RC).....	81
Tabla 21 Caminos básicos del flujo.....	97
Tabla 22 Rango de valores de para la evaluación técnica de los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización) relacionados con la métrica TOC.....	120
Tabla 23 6 Resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización).....	120
Tabla 24 Rango de valores de para la evaluación técnica de los atributos de calidad (Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas) relacionados con la métrica RC..	123
Tabla 25 Resultados de la evaluación de la métrica RC y su influencia en los atributos de calidad (Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas). .....	124

## ÍNDICE DE FIGURAS

Figura 1 La figura muestra el mapa de procesos donde se define el proceso Movimiento de Fuerza de Trabajo .....	43
Figura 2 Representa el modelo conceptual.....	45
Figura 3 Funcionamiento del patrón MVC.....	65
Figura 4 Modelo de datos del componente Trabajador .....	69
Figura 5 Representación de los resultados obtenidos en el instrumento agrupados en los intervalos definidos .....	79
Figura 6 Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos .....	79
Figura 7 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.....	80
Figura 8 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de Implementación .....	80
Figura 9 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.....	81
Figura 10 Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos .....	82
Figura 11 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento .....	83
Figura 12 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de Mantenimiento .....	83

## ÍNDICE

---

Figura 13 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de Pruebas .....	84
Figura 14 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización.....	84
Figura 15 Diagrama de componentes para el componente Trabajador .....	90
Figura 16 Representación de pruebas de Caja Blanca .....	91
Figura 17 Notación de grafos de flujo para las instrucciones: Secuenciales, If, While .....	92
Figura 18 Notación de grafos de flujo para la instrucción Case.....	93
Figura 19 Representación del algoritmo adicionarMovalta(\$idpagosadicionales).....	94
Figura 20 Grafo de flujo asociado al algoritmo adicionarMovalta(\$arrModelomov, \$arrTrabajador, \$arrPagosadicionales) .....	95
Figura 21 Representa el prototipo de interfaz de usuario Emitir movimiento de nómina .....	109
Figura 22 Representa el prototipo de interfaz de usuario Movimiento de altas.....	110
Figura 23 Representa el prototipo de interfaz de usuario Movimiento de reubicación .....	111
Figura 24 Representa el prototipo de interfaz de usuario Movimiento de bajas.....	112
Figura 25 Representa el prototipo de interfaz de usuario Actualizar datos contables.....	113
Figura 26 Representa el prototipo de interfaz de usuario Buscar trabajador .....	114
Figura 27 Diagrama de clases de diseño asociado al requisito Buscar trabajador .....	115
Figura 28 Diagrama de clases de diseño asociado al requisito Movimiento de alta .....	116
Figura 29 Diagrama de clases de diseño asociado al requisito de Movimiento de baja .....	118
Figura 30 Diagrama de clases de diseño asociado al requisito de Datos contables.....	119
Figura 31 Gráfica de los resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización).....	123
Figura 32 Gráfica de los resultados de la evaluación de la métrica RC agrupados por la tendencia de los valores .....	127

### INTRODUCCIÓN

Hoy en día la verdadera riqueza de las diferentes organizaciones no radica en sus recursos financieros ni mucho menos en las modernas tecnologías que tengan en explotación, sino en su Capital Humano y la manera en que el mismo se logre gestionar. Resulta complejo organizar los procesos de evaluación del desempeño, capacitación, remuneración, motivación y gestión de los trabajadores sino se dispone de un inventario bien estructurado y actualizado de todas las personas con que cuenta la organización.

La Revolución Cubana ha estado constantemente trazándose estrategias para lograr una mayor eficiencia y productividad en cada proceso que se desarrolla, recorriendo un intenso camino de transformaciones y más en estos últimos años, donde se puede ver el siglo XXI como un siglo de gran auge en la automatización y la aplicación de técnicas en el desarrollo de distintas actividades.

El control estricto de los trabajadores junto a los datos personales de cada uno, constituye la base para la realización de todos los procesos asociados a la gestión de Capital Humano. En Cuba se han utilizado innumerables sistemas informáticos para estos fines, los cuales de una manera u otra logran automatizar parcialmente los procesos de gestión de los trabajadores sin llegar a cumplimentar los requisitos establecidos en las Normas Cubanas (NC) para un Sistema de Gestión Integrada de Capital Humano.

La inexistencia de sistemas que realicen una gestión integral de los trabajadores que permita realizar un análisis detallado de la amplia gama de aspectos que caracterizan el comportamiento de una persona en un entorno laboral enfocado a las entidades cubanas, provoca:

- Pérdida en la documentación de los movimientos de nómina que se realizan a los trabajadores.
- Descontrol en la plantilla de los trabajadores de las entidades.
- La cantidad de documentos que se maneja y se archivan dificultan la operatividad de las informaciones estadísticas que se solicitan.
- Ineficiencia en el trabajo de los departamentos de Capital Humano.

Como parte del amplio proceso de automatización de la economía cubana que se viene desarrollando a lo largo de todo el país y ante los enormes retos que nos impone el mundo actual, la necesidad de identificar y explotar las gigantescas reservas de productividad del trabajo y poder satisfacer las exigencias requeridas en la legislación laboral vigente se impone la necesidad de gestionar a los trabajadores desde un enfoque integral y de manera flexible que permita la adaptabilidad al entorno cambiante que nos rodea.

Analizando lo anterior queda definido el siguiente **problema a resolver**: Se carece de un sistema que gestione los procesos de los trabajadores en las entidades cubanas de forma integral. A partir del problema planteado se define como **objeto de estudio**: los procesos de gestión de Capital Humano para las entidades cubanas y como **campo de acción**: los procesos de gestión de los trabajadores en las entidades cubanas.

Para darle solución al problema planteado se determina como el **objetivo general**: desarrollar un componente que permita la gestión de los trabajadores en las entidades cubanas.

Para cumplir con el objetivo general se definieron los siguientes **objetivos específicos**:

- Realizar un estudio de las vías existentes para la gestión de los trabajadores.
- Modelar la solución de software.
- Implementar la solución bajo los estándares definidos en el proyecto ERP-Cuba.
- Comprobar la calidad del componente.

De esta manera para darle cumplimiento a los objetivos específicos propuestos se determinan las siguientes **tareas de investigación**:

- Estudiar los procesos para la gestión de los trabajadores.
- Estudiar sistemas existentes que gestionen los procesos referentes a la gestión de los trabajadores.
- Realizar modelo de negocio.
- Realizar descripción de los requisitos funcionales.
- Diseñar el modelo datos teniendo en cuenta el análisis realizado.
- Estudiar documentación del marco de trabajo.
- Estudiar documentación de las herramientas a utilizar para el desarrollo del componente.
- Realizar pruebas de caja blanca al componente.

### **Hipótesis**

Si se realiza un componente que permita la gestión integral de los trabajadores se contribuirá a la eficiencia en la gestión de los procesos relacionados con la gestión de los trabajadores en las entidades cubanas.

El presente trabajo está estructurado a través de 4 capítulos de la manera siguiente:

**CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA:** En este capítulo se abordan varios temas entre los que se encuentran la fundamentación de algunos conceptos generales con el objetivo de un mejor entendimiento de los procesos relacionados con la gestión de los trabajadores, además se hace una investigación sobre sistemas existentes que están vinculados a estos procesos, además de un estudio de las metodologías, tecnologías y herramientas empleadas para el desarrollo de la solución propuesta.

**CAPÍTULO 2: MODELACIÓN DEL NEGOCIO Y ANÁLISIS:** Se definen los procesos de negocio y su descripción así como el mapa de procesos para mostrar la interacción entre ellos. Se realiza una representación del modelo conceptual para determinar las relaciones entre las entidades del negocio. Se describe además la solución propuesta mediante los requisitos funcionales así como la descripción de cada uno de ellos. Se validan los requisitos funcionales teniendo en cuenta las técnicas definidas para ello.

**CAPÍTULO 3: DISEÑO:** Se abordan temas entre los que se encuentran los patrones de diseño utilizados, se modelan los diagramas de clases del diseño permitiendo dar paso a la implementación del sistema, se representa además el diagrama de datos para mostrar las tablas vinculadas al sistema y sus relaciones, y se realiza una evaluación del modelo de diseño realizado, así como también se valoran los resultados obtenidos.

**CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS:** Se modela el diagrama de componentes, permitiendo con ello tener una visión mas clara de la implementación del sistema, y de las integraciones entre los componentes que están relacionados con el sistema. Se abordan temas como las pruebas realizadas al software, en específico las pruebas de caja blanca, además se hace una valoración de las mismas según los resultados obtenidos.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### **Introducción**

Este capítulo tiene como objetivo fundamental abordar varios temas, entre los que se pueden citar; algunos conceptos teóricos que servirán de base para el desarrollo del trabajo. Además se realizará una investigación en base a sistemas existentes que están vinculados a los proceso de gestión de los trabajadores, así como se analizarán las metodologías, tecnologías y las herramientas utilizadas para el desarrollo del sistema propuesto.

### **1.1 Conceptos Generales**

#### **1.1.1 Capital Humano**

El concepto de Capital Humano tiene una trascendencia a través de la historia dado su significado, hace muchos siglos atrás sería interpretado su nombre como referencia a los fondos que se utilizarían para la puesta en funcionamiento de una empresa.

Aun así en la actualidad existen disímiles concepciones e interpretaciones ya sean de carácter económico o no de acuerdo al concepto de Capital Humano, de esta manera es muy común referirse al Capital Humano como las cualidades y características de las personas de una organización o sea, sus aspectos intangibles, por ejemplo, experiencia, capacidades, formación, educación, escolarización, el conocimiento, salud, condiciones de vida y trabajo e información, y cuando se trata de los aspectos tangibles en la organización como cantidad, salario, contratación y demás, entonces se consideran recursos humanos. (1)

El Capital Humano es “conjunto de conocimientos, experiencias, habilidades, sentimientos, actitudes, motivaciones, valores y capacidad para hacer, portados por los trabajadores para crear más riquezas con eficiencia. Es, además, conciencia, ética, solidaridad, espíritu de sacrificio y heroísmo”. (2)

#### **1.1.2 Trabajador**

Un trabajador es una persona que con la edad legal suficiente y “quien voluntariamente preste sus servicios retribuidos por cuenta ajena y dentro del ámbito de organización y dirección de otra persona, física o jurídica, denominada empleador o empresario”. (41)

Según la Ley Federal del Trabajo en su artículo 8 plantea que un trabajador es “la persona física que presta a otra, física o moral, un trabajo personal subordinado. Para los efectos de esta disposición, se entiende por trabajo toda actividad humana, intelectual o material, independientemente del grado de preparación técnica requerido por cada profesión u oficio.”(42)

### **1.1.3 Normas para el Movimiento de la Fuerza de Trabajo**

#### **1.1.3.1 Movimiento de la Fuerza de Trabajo**

El Movimiento de la Fuerza de Trabajo norma el procedimiento a seguir para la realización de los movimientos de altas, bajas, reubicaciones y modificaciones salariales del trabajador como parte del proceso de gestión de los trabajadores dentro de la gestión integrada de Capital Humano. Permite además confeccionar el modelo correspondiente (Modelo SC-4-02) constituyendo así el documento que respalda las anotaciones para mantener actualizados el maestro de nóminas que sirve de fuente para la preparación y pago de las nóminas.

Según la resolución 13 planteada en el 2007 por el Ministerio de Finanzas y Precios de Cuba para la realización de este modelo son de uso obligatorio los siguientes datos:

- Nombre y código de la entidad.
- Fecha de emisión del modelo.
- Nombres, primero y segundo apellidos del trabajador y número de su Expediente Laboral.
- Tipo de movimiento: alta, baja, cambio de cargo, reubicación temporal.
- Fecha del movimiento y área de ubicación.
- Fecha del cambio de la estructura salarial, relación de cada concepto que integra la estructura salarial modificada.
- Motivo de la modificación salarial.
- Firma del trabajador que elabora el modelo.
- Firma del trabajador que aprueba la modificación.
- Firma del trabajador encargado de revisar las Nóminas como acuse de recibo del modelo.

- Firma del trabajador que actualiza el modelo SC-4-01 - Control de Datos de Nóminas.
- Firma del trabajador encargado de elaborar las Nóminas como acuse de recibo.
- Número consecutivo del modelo.

De esta manera queda elaborado el Movimiento de Nóminas al que se le incluyen además una serie de datos para facilitar la actualización del maestro de nóminas, tales como el régimen de trabajo y descanso, forma de pago, categoría ocupacional, grupo de complejidad así como la cantidad de horas que debe trabajar. El Movimiento de la Fuerza de Trabajo permite además la actualización del registro de los trabajadores.

Entre los procedimientos que norman el Movimiento de la Fuerza de Trabajo se encuentran:

### ➤ **Movimiento de altas**

Se consideran como un movimiento de alta todas las incorporaciones de personas al registro de trabajadores de una entidad mediante la formalización de la relación laboral en cualquiera de sus modalidades e independientemente de la causa que lo originan. Las fuentes de reclutamiento pueden agruparse en:

- Personas sin vínculo laboral.
- Reubicación de disponibles.
- Egresados de nivel medio y superior.
- Procedentes de otras entidades.
- Procedentes de sectores especiales.
- Otras fuentes de altas.

### ➤ **Movimiento de reubicación**

Las reubicaciones comprenden todos los movimientos internos de los trabajadores que implican cambios salariales, de cargos, de áreas de trabajo, entre otras. Por ejemplo si se le añade o elimina un pago adicional, puesto, área o cargo implicaría generar un movimiento de este tipo. Pueden agruparse en:

- Cambios definitivos de cargo, área o puesto de trabajo.

- Cambio provisionales de cargo, área o puesto de trabajo.
- Modificaciones salariales.
- Suspensión de la relación laboral.
- Movimiento de cuadro.

➤ **Movimiento de bajas**

Se consideran como bajas todos los trabajadores que son disminuidos del registro de trabajadores de la entidad, independientemente del tipo de contrato y las causas que lo originen. La relación laboral termina por las causas generales siguientes:

- Acuerdo de las partes.
- Iniciativa del trabajador.
- Iniciativa de la administración.
- Jubilación del trabajador.
- Fallecimiento del trabajador.
- Extinción de la entidad laboral, cuando no existe otra que se subroga en su lugar.

Existen además otro tipo de bajas conocidas como **bajas por fluctuación** las que se caracterizan porque comprenden las disminuciones de trabajadores del registro producidas por la propia voluntad del trabajador, originándose con ello un movimiento no planificado de la fuerza de trabajo cuyas causas se atribuyen a razones de índole personal o social. Las bajas por fluctuación se clasifican en tres grupos:

- Bajas por iniciativa del trabajador.
- Bajas por separación definitiva.

### **1.1.3.2 Registro de trabajadores**

Es el documento donde aparecen relacionados todos los trabajadores que conforman el personal de la Entidad una vez formalizada su relación laboral.

### **1.1.3.3 Datos contables**

El objetivo de los datos contables es asociarle al trabajador una cuenta, un centro de costo y el número de la tarjeta del banco en caso de que el trabajador tenga. Estos datos se llenan cada vez que se le hace un movimiento de alta a un trabajador y cada vez que se le haga una modificación a algún dato del trabajador que implique un cambio de los datos contables como el centro de costo y la tarjeta del banco.

### **1.1.4 Métodos para la identificación de requerimientos**

Cuando se va a realizar un producto de software primeramente se debe identificar lo que se quiere que el mismo realice, para lograr esto se aplican una serie de técnicas que dan lugar a las características que va a poseer el sistema. Teniendo en cuenta esto quedan definidas a continuación algunas de ellas:

#### **➤ Introspección**

Método en el cual el analista imagina que le gustaría al usuario y lo hace intentando ponerse en su lugar.

Actualmente es un método poco afectivo debido a que si se usara se pueden omitir muchos detalles.

Por consiguiente la persona, en este caso, el Ingeniero de Requerimientos desconocería las habilidades del usuario con respecto al uso de la computadora así como lo que el mismo desee que el sistema tenga como finalidad.

#### **➤ Entrevistas**

Es el método clásico. La entrevista es la técnica más utilizada en la obtención de requisitos de software, se emplea para recolectar información verbal a través de preguntas elaboradas previamente por el analista para después aplicarla al cliente, también puede de alguna manera ponerse en contacto con el cliente en un ambiente neutro y controlado. La meta de una entrevista es enterarse de opiniones, sentimientos acerca del sistema, la organización y metas del sistema y personales del cliente logrando obtener toda la información posible desde la visión que el entrevistado tiene de los requisitos.

Por lo general no se entrevista a toda la gente que se relacionará con el sistema, sino a una selección de personas que represente a todos los sectores críticos de la organización, con especial atención puesto en los sectores más afectados o que harán un uso más frecuente del nuevo sistema.

Los analistas necesitan ser perceptivos a las dificultades que algunos entrevistados crean durante la entrevista y saber cómo lidiar con problemas potenciales. Es importante también considerar no sólo la

información que se consigue a través de la misma sino también, su relevancia. El entrevistador debe centrar la entrevista cuando esta se desvía.

Los requerimientos que surgen de las entrevistas frecuentemente se contradicen entre ellos o se expresan desde el desconocimiento de los detalles del funcionamiento del sistema, sus potencialidades, interdependencias o limitaciones; por lo que se debe trabajar con los mismos para corregir sus fallas. Esta técnica depende en gran medida de las habilidades del entrevistador, la fluidez de su lenguaje y perspicacia para captar las ideas en un tiempo bastante limitado.

Tienen su aspecto negativo debido a que las personas no siempre están dispuestas a ser entrevistadas debido a que les molesta hacer un mal papel, no expresan todo o que saben, temen perder poder si revelan lo que conocen o no se sienten en confianza con el analista especialmente sí es de otro lugar ajeno a la empresa.

En [Kendall, K, Kendall, J (1995)] se proponen cinco pasos para la preparación de una entrevista:

- Leer el material referente a la aplicación y antecedentes del sistema.
- Establecer objetivos de la entrevista.
- Decidir a quiénes entrevistar. Deben escogerse personas clave, que son aquellas que afectarán el sistema de alguna manera.
- Preparar al entrevistado. Avisarle que vaya pensando en el sistema y solicitarle cita y tiempo de duración de la entrevista.

Decidir sobre el tipo de preguntas y la estructura de la entrevista. Se señalan dos tipos de preguntas las abiertas que permiten al entrevistado expresar más de lo que se le pregunta y cerradas que limita la respuesta a una lista de donde escoger.

Para lograr una entrevista exitosa se debe acordar una cita por anticipado con las personas que se entrevistarán, avisar a los entrevistados sobre la naturaleza de la entrevista, planear una entrevista común por no más de una hora, prepararla conociendo de antemano a los individuos que se van a entrevistar y familiarizarse con el tema y preparar un conjunto apropiado de preguntas.

### ➤ **Tormenta de ideas**

Este es un método que se usa para generar ideas. Consiste en una reunión de varios interesados en la que todos ellos expresan sus ideas sobre el problema y su solución. Pueden participar todas aquellas personas que se encuentren afectados por la aplicación, es decir los que la van a utilizar y el personal experimentado del equipo de desarrollo, como los analistas de sistema. La forma de llevarla a cabo es que cada participante diga su idea sin ser interrumpido por otro, las ideas se pueden ir anotando en un lugar donde puedan ser vistas y no deben ser criticadas por otros. Después se van eliminando en base a distintos criterios como por ejemplo, "caro", "impracticable", "imposible". Al finalizar la sesión de tormenta de ideas se puede hacer una recolección de ideas sin duplicidad. Este método es sólo de apoyo y resulta bueno para iniciar la interacción con los clientes y recolectar una primera impresión del sentimiento de los participantes en el sistema de software. Entre las desventajas de la lluvia de ideas está el hecho de que personas con muy buenas ideas, no participan y se las reservan porque no les gusta hablar en público.

### ➤ **Cuestionarios**

Consiste en un conjunto de preguntas presentadas a una persona para su respuesta. Su aplicación sirve para obtener información acerca de: actitudes, creencias, comportamiento y características de las personas clave en la organización donde se realizará el sistema de software. Idealmente el cuestionario debe ser simple, específico, no patrocinado, con precisión técnica y dirigido a quienes deben conocerlo. Los cuestionarios son útiles si la gente se encuentra muy dispersa, es mucha la gente involucrada en el sistema, hay necesidad de recolectar más datos antes de recomendar alternativas o es necesario sensibilizar a la gente antes de proceder a las entrevistas. Es necesario asegurarse de la validez y confiabilidad del cuestionario. Se entiende que es válido si mide lo que se intenta medir y es confiable si los resultados son consistentes. Resulta importante el ordenamiento con sentido y la agrupación de las preguntas para entender las posibles respuestas.

Entre los problemas que se presentan con los cuestionarios es que el lenguaje utilizado debe ser muy preciso y se debe hacer un análisis previo del dominio del problema. Se necesita bastante práctica en la elaboración de un buen cuestionario. La forma de medida es dictada por los datos y la escala puede ser: nominal, ordinal, de intervalo o razón.

### ➤ **Análisis de Protocolos**

La técnica consiste en pedirles a los usuarios potenciales que describan en voz alta las actividades que realizan dentro del sistema, procediendo así el analista a definir metas y submetas. La herramienta principal es directamente el lenguaje natural que puede traer como consecuencia que la cantidad de enunciados ambiguos dependientes del contexto sea bastante grande, haciendo con esto que a el analista le surjan muchas dudas sobre lo que se quiso decir el usuario al hacer su relato del sistema. Por otro lado, para el usuario resulta poco natural estar hablando en voz alta sin interlocutor sobre lo que hace, por lo que es sumamente probable que deje muchos detalles sin expresar.

### ➤ **Revisión de documentos**

Esta técnica depende de la información almacenada por las entidades referentes a los procesos y términos que se manejan dentro de la misma. Las entidades guardan información acerca de sus procesos, los modelos o informes necesarios para el desarrollo de la misma o para rendir cuenta a los organismos superiores.

Pueden ser además videoconferencias y otros soportes dependiendo del grado de automatización de las mismas. Esta cantidad de información es estudiada por los analistas en busca de entender bien todos los procesos para determinar los requisitos asociados a estos, y que luego deben ser verificados por otras técnicas de captura de requisitos. La revisión de documentos no es efectiva por sí sola, pero debe ser combinada con otra (s) para lograr un resultado efectivo.

### ➤ **Elaboración de prototipos**

Este método es útil porque permite observar las reacciones del usuario y sus habilidades, se pueden incorporar sugerencias e innovaciones antes de entregar una primera versión completa. Los prototipos son simulaciones del posible producto que después son utilizados por el usuario final permitiendo conseguir una retroalimentación en cuanto a si el sistema diseñado en base a los requisitos recolectados le permite al usuario realizar su trabajo de manera eficiente. El sistema resultante será más cercano al usuario.

### 1.1.5 Técnicas de validación de requerimientos

#### ➤ **Prototipos de interfaz de usuario**

Algunas propuestas se basan en obtener de la definición de requisitos que, sin tener la totalidad de la funcionalidad del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario. (3)

#### ➤ **Listas de chequeo**

Son frecuentemente usadas en inspecciones o revisiones de artefactos generados en el proceso de producción de software; son listas de aspectos que deben ser completados o verificados. (3)

#### ➤ **Auditorías**

La revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra una (Lista de Chequeo) predefinida o definida a comienzos del proceso, es decir sólo una muestra es revisada. (3)

#### ➤ **Diseño de casos de prueba**

El objetivo del proceso de diseño de casos de prueba es crear un conjunto de casos de prueba que sean efectivos descubriendo defectos en los programas y muestren que el sistema satisface sus requerimientos. (4)

### 1.1.6 Patrones de diseño

Los patrones de diseño surgen ante la necesidad de la reutilización de diseños y de soluciones a problemas frecuentes encontrado por los ingenieros. Con la reutilización se consigue la reducción de tiempos y la disminución del esfuerzo de mantenimiento, esto trae consigo mayor eficiencia y consistencia en el diseño de la solución. Existen varias formas de hacer reutilización de diseño en las tareas del diseño de sistema como por ejemplo el diseño basado en componentes, frameworks, objetos distribuidos o los patrones de diseño. Para crear reutilización en el diseño no necesariamente se necesita de un consumidor de reutilización, porque el propio tiempo de desarrollo convertirá el equipo de diseño en consumidores de diseño.

Las características que presentan los patrones son las siguientes:

- Son soluciones concretas: Proponen soluciones a problemas concretos, no son teorías genéricas.
- Son soluciones técnicas: Indican resoluciones técnicas basadas en Programación Orientada a Objetos (POO). En ocasiones tienen más utilidad con algunos lenguajes de programación y en otras son aplicables a cualquier lenguaje.
- Se utilizan en situaciones frecuentes: Ya que se basan en la experiencia acumulada al resolver problemas reiterativos.
- Favorecen la reutilización de código: Ayudan a construir software basado en la reutilización, a construir clases reutilizables. Los propios patrones se reutilizan cada vez que se vuelven a aplicar.
- El uso de un patrón no se refleja en el código: Al aplicar un patrón, el código resultante no tiene por qué delatar el patrón o patrones que lo inspiró. No obstante últimamente hay múltiples esfuerzos enfocados a la construcción de herramientas de desarrollo basados en los patrones y frecuentemente se incluye en los nombres de las clases el nombre del patrón en que se basan facilitando así la comunicación entre desarrolladores. (37) (38)

### 1.1.7 Componente

Para que un elemento pueda ser considerado un componente existen algunas características claves que hay que tener en cuenta, entre ellas se encuentran:

- **Identificable:** Debe tener una identificación que permita acceder fácilmente a sus servicios y que permita su clasificación.
- **Auto contenido:** Un componente no debe requerir de la utilización de otros para finalizar la función para la cual fue diseñado.
- **Puede ser reemplazado por otro componente:** Se puede reemplazar por nuevas versiones u otro componente que lo reemplace y mejore.
- **Con acceso solamente a través de su interfaz:** Debe asegurar que estas no cambian a lo largo de su implementación.
- **Sus servicios no varían:** Las funcionalidades ofrecidas en su interfaz no deben variar, pero su implementación sí.
- **Bien Documentado:** Un componente debe estar correctamente documentado para facilitar su búsqueda si se quiere actualizar, integrar con otros, adaptarlo.

- **Es genérico:** Sus servicios debe servir para varias aplicaciones.
- **Reutilizado dinámicamente:** Puede ser cargado en tiempo de ejecución en una aplicación.
- **Independiente de la plataforma:** Hardware, Software, S.O. (5)

En si un componente de software es todo aquel recurso desarrollado para un fin concreto y que puede formar solo o junto con otros, un entorno funcional requerido por cualquier proceso predefinido.

## 1.2 Sistemas estudiados

En Cuba actualmente se emplean varios sistemas integrales de gestión, algunos de carácter internacional, desarrollados por compañías extranjeras y otros que son productos de carácter nacional. Estos sistemas cuentan con la posibilidad de gestionar los Recursos Humanos y a su vez incluyen como uno de sus procesos la gestión de los trabajadores teniendo en cuenta las especificidades en cada caso. Como ejemplos vigentes están:

- ASSETS NS
- SAGE MAS 500
- EXACT GLOBE
- KEWAN
- VERSAT SARASOLA
- RODAS XXI

### ➤ **ASSETS NS**

El mismo se introdujo en Cuba en el año 1997, es utilizado por diferentes sectores en el país entre los que están, el Ministerio de la Educación Superior, el Consejo de Estado, la Aduana General de la República, el Ministerio de Auditoría y Control, el de Justicia y el de Finanzas y Precios. Posee un módulo en el cual se desarrollan los procesos relacionados con la gestión de los Recursos Humanos donde se pueden controlar íntegramente los recursos laborales: empleados, estructura organizativa de la entidad y plantilla, teniendo en cuenta además que permite realizar funciones como la de introducción de altas, bajas y otros movimientos a los empleados en una organización determinada.

Entre sus requerimientos técnicos se encuentran:

- Trabaja sobre el sistema operativo Windows 98 o superior.
- Soporte para bases de datos SQL Server 2000.
- Implementado en Visual Basic.
- Utiliza Crystal Reports 7.0 para la generación de reportes. (6)

Este sistema aunque logra la gestión de los trabajadores no permite mantener un registro actualizado de todo el personal con el que cuenta la entidad en un momento determinado, además por sus bases tecnológicas se demuestra que no es un sistema multiplataforma, y el lenguaje que se utiliza para su desarrollo, en este caso Visual Basic, no cumple con la soberanía que requiere para las entidades cubanas.

### ➤ **KEWAN**

Es un sistema de gestión integral desarrollado por Dimensión Informática, una compañía Española. Tiene antecedentes en Cuba a través de su uso en el ministerio de la construcción como por ejemplo en las empresas de construcción en Varadero, así como pronósticos de aplicarse en el Instituto de Aeronáutica Civil de Cuba (IACC). Este software actúa a partir de una base de datos única que trabaja con módulos independientes o conectados según los intereses, en las áreas de economía y finanzas, logística y comercial, recursos humanos, gestión de proyectos y producción. (7)

El sistema dirige más sus esfuerzos a las gestiones contables y sus activos, presentando algunas deficiencias en cuanto al modelo cubano para el diseño e implementación de un Sistema de Gestión Integrada de los Recursos Humanos, no presentando la posibilidad de tener el control de los diferentes movimientos realizados sobre un trabajador así como llevar un registro actualizado de los trabajadores que posee una entidad.

### ➤ **SAGE MAS 500**

Este sistema contable financiero cuenta con un conjunto de módulos disponibles, integrados entre si, los cuales cubren las áreas de negocios de Finanzas, Administración de Proyectos, Manufactura, Adaptación, Distribución, Inteligencia de Negocios e informes, y la administración de los Recursos Humanos integrada con Nóminas. En SAGE MAS 500 los Recursos humanos incluyen tres módulos que

son Recursos Humanos Abra, Nómina Abra y Control de Asistencia Abra, a través del primero se provee la gestión de los recursos humanos, y a su vez incluye la gestión de los empleados. Además brinda la posibilidad de manejar los datos de los empleados así como un alto nivel de seguridad de los mismos. Permite además la integración con otros sistemas y de la misma manera gestionar la información de los trabajadores. Mantiene una información detallada de los empleados a través de un historial.

Herramientas que se emplean en este sistema:

- El modulo de Recursos Humanos esta integrado con Microsoft Office, donde hace uso de Microsoft Word y Microsoft Excel a la hora de analizar los datos y tomar decisiones.
- Utiliza además Crystal Reports 7.0 para la generación de informes.
- Esta implementado en Visual Basic.

Requerimientos del sistema:

- Como sistema operativo para el servidor cuenta con Windows 2000 Server.
- Como servidor de base de datos utiliza SQL Server 2000. (8)

SAGE MAS 500 gestiona de forma más integral los trabajadores dentro de su módulo de Recursos Humanos sin embargo no es una propuesta viable para la economía del país debido a que se incurre en costosos gastos en conceptos de licenciamientos y mantenimientos de software y las tecnologías que se utilizan para el desarrollo y sustento del sistema son propietarios.

### ➤ **VERSAT SARASOLA**

Es un producto cubano cuyo Sistema Integral de gestión es el primer sistema de contabilidad certificado, desarrollado para la gestión económica, este estuvo representado por el Ministerio de Finanzas y Precios, consultorías internacionales y el organismo encargado de la seguridad informática. Surge en 1998 con el objetivo de gestionar y actualmente es utilizado en nuestro país en alrededor de 200 entidades de varias provincias y en lo adelante será introducido en más de dos mil 500 unidades presupuestadas. Este sistema integrado cuenta con un conjunto de 12 módulos entre los que se encuentran por ejemplo: Costos y procesos, Control de inventarios, Finanzas y caja, Contratación, Facturación y otros.

Entre las características que posee este sistema están:

- Implementado en Delphi.

- Trabaja sobre el sistema operativo Windows.
- Soporte para bases de datos SQL Server 2000. (9) (10)

Dicho sistema aunque gestiona los recursos humanos, no lo hace de forma organizada debido a que no cuenta con un módulo específico que gestione los mismos, su objetivo principal va dirigido hacia la elaboración de la nómina , por lo solo recoge de los trabajadores los datos necesarios para la misma. No disponiendo de una plataforma de intercambio de información controlado, flexible y ágil, y no es un sistema multiplataforma.

### ➤ **EXACT GLOBE**

Es un producto de software que forma parte de las soluciones de la compañía Exact Software establecida desde sus inicios en Holanda, presenta una solución completa y constituye un sistema integral de gestión diseñado para empresas medianas y recomendable para empresas internacionales, en este se incluyen diferentes funcionalidades relacionadas con las áreas de contabilidad, distribución, logística, servicios, Nóminas, Proyectos, Fabricación así como la gestión de los Recursos Humanos (R.H). Incluido en los procesos de Gestión de los R.H, Exact Globe posibilita que se desarrolle la gestión de los trabajadores con un carácter organizativo, y permite que se guarden todos los procesos y movimientos relacionados con el personal en una empresa mediante la asignación de ficha personal para el almacenamiento de sus datos. (11) (12)

Herramientas que se emplean en este sistema:

- Utiliza Crystal Report 7.0 para la generación de informes.
- Posee integración con Microsoft Excel.

Aunque es un sistema avanzado con respecto a las exigencias funcionales que requiere el país los gastos en cuestiones de licencia y mantenimientos del software son muy elevados y afectan considerablemente la economía del país, además de no cumplir fielmente con lo establecido en las normas cubanas por lo que requiere de algunas mejoras en caso que se quiera aplicar a nivel nacional.

### ➤ **RODAS XXI**

Es un Sistema Integral Económico Administrativo cubano desarrollado por CIMATEL. El mismo es un sistema multiempresa y cuenta con un conjunto de módulos entre los que se encuentran: Contabilidad,

Activos Fijos, Nóminas, Inventario, Facturación, Recursos Humanos. Cada uno de estos módulos se encuentran integrados entre sí logrando un intercambio de información de manera eficiente y coherente. Este sistema permite la gestión, planificación y control de los Recursos Humanos mediante la actualización de los datos de los trabajadores incluyendo su foto, así como el registro de incidencias y los datos necesarios para el módulo de Nóminas. Se pueden realizar movimientos de nóminas. Permite además la actualización de los datos de los trabajadores. Para darle alta a un trabajador deben estar conformadas las plantillas de cargos y ocupaciones de la Entidad. Se lleva además todo el registro de bajas.

Características generales de este sistema:

Como configuración mínima requerida para ejecutar el sistema esta:

- Computadora IBM PC o compatible.
- Pentium III o superior.
- 256 MB de memoria RAM o más.
- Unidad de disco flexible, un disco duro, lector de CD y sistema operativo Windows 2000/XP.
- 20GB de espacio en disco duro. (13)

El RODAS XXI es un sistema que gestiona los trabajadores de forma organizada sin embargo no cumple con requisitos establecidos en las Normas Cubanas (NC) para un Sistema de Gestión Integrada de Capital Humano, además de no ser multiplataforma pues solo es compatible con el sistema operativo Windows 2000/XP.

Teniendo en cuenta la investigación con relación a varios de los sistemas analizados se llega a la conclusión de la necesidad de crear un nuevo sistema que permita la gestión de los trabajadores y que cumpla con las necesidades de las entidades a nivel nacional, tomándose la decisión de desarrollar un componente Trabajadores que forme parte del subsistema Capital Humano del Sistema Cedrux, y que respalde los lineamientos nacionales de desarrollo de soluciones bajo el principio de independencia tecnológica.

### **1.3 Modelo de desarrollo de software orientado a componentes**

El modelo de desarrollo de software orientado a componentes es la metodología definida por el proyecto ERP-Cuba, y por consiguiente se aplica en la realización de este trabajo. Este modelo de desarrollo está orientado a las necesidades y artefactos que se realizan durante todo el proceso de desarrollo del proyecto. Además este constituye una combinación de diferentes metodologías obteniendo lo más provechoso y conveniente de estas para llevar a cabo el proyecto. Entre las características que posee están que permite la modelación del negocio a través de procesos; posibilita realizar la ingeniería de requisitos mucho más explícita; dado que es orientada a componentes permite por ejemplo que exista independencia en las funciones del sistema en caso de que se modifique el sistema funcional; emplea como lenguaje de modelado para el negocio BPMN y para el diseño UML. Además este modelo de desarrollo permite que en los flujos de análisis y diseño se generen artefactos significativos como son:

modelo de proceso de negocio, descripción de procesos de negocio, modelo conceptual, prototipo de interfaz de usuario, especificación de requisitos, casos de prueba, diagrama de clases y descripción del diseño de clases.

### **1.4 Herramientas y tecnologías**

#### **1.4.1 Visual Paradigm**

Visual Paradigm para UML (Lenguaje de Modelado) es una herramienta que emplea UML como lenguaje de modelado, soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Se usó para el modelado del sistema propuesto debido a que entre sus utilidades permite construir aplicaciones con mejor calidad, además posibilita entre sus funciones realizar diagramas de clases, código inverso, así como generar código desde diagramas y generar documentación.

#### **1.4.2 Lenguaje de Modelado**

##### **1.4.2.1 UML**

El Lenguaje Unificado de Modelado, en sus siglas en inglés (UML), es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

UML un lenguaje de propósito general para el modelado orientado a objetos. UML es también un lenguaje de modelamiento visual que permite una abstracción del sistema y sus componentes.

Objetivos del UML:

- UML es un lenguaje de modelado que puede ser usado por todos los modeladores.
- UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.
- Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.(15)

### **1.4.2.2 Notación para el Modelado de Procesos de Negocio (BPMN)**

La Notación para el Modelado de Procesos de Negocio es un estándar nuevo para el flujo de procesos del modelado de negocios y los servicios web. BPMN define un Diagrama de Procesos de Negocio basado en la técnica de diagramado de flujos que ajusta modelos gráficos de operación de procesos de negocio.

Este diagrama está estructurado en categorías básicas de elementos, entre las que se encuentran:

- objetos de flujo.
- objetos de conexión.
- Artefactos.
- Carriles. (16)

BPMN se aplicará para la modelación de negocio del sistema en la elaboración del Mapa de procesos dado el enfoque que este representa hacia los proceso de negocio con el objetivo de definir como se desarrolla la gestión de los procesos que forman parte de la solución así como representar el flujo de información de los mismos.

### **1.4.3 Lenguajes de programación**

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo. Son considerados además herramientas que nos

permiten crear programas y software. Estos facilitan las tareas de programación ya que poseen formas adecuadas para su entendimiento y resultan independientes de la computadora a utilizar.

### 1.4.3.1 AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML) es una tecnología que facilita la creación de aplicaciones interactivas en la Web que se ejecutan en el navegador de los usuarios y mantienen comunicación asíncrona con el servidor; posibilitando que se puedan efectuar cambios sobre una página sin necesidad de recargarla, aumentando de esta manera la interactividad, velocidad y usabilidad de la misma.

AJAX está conformado por:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

Además,

- Provee un mecanismo para mezclar y hacer coincidir XML con XHTML.
- Las aplicaciones son más rápidas e interactivas, al estilo aplicaciones de escritorio.
- Reduce de manera significativa tener que cargar información continuamente del servidor, actualizando solamente porciones de la página.
- Cuando se utiliza AJAX adecuadamente en el desarrollo de una aplicación, se reduce de manera significativa los tiempos de carga inicial. (17)

A continuación se explican las características más importantes de algunas de las tecnologías que componen AJAX:

#### ➤ XML

Es el estándar de Extensible Markup Language (Lenguaje de Etiquetado Extensible), que se encuentra conformado por un conjunto de reglas para definir etiquetas semánticas orientadas a organizar un documento en diferentes partes. Permite al usuario definir sus propios lenguajes de anotación adaptados a sus necesidades y contiene tres características muy importantes que son: extensibilidad, estructura y validación. Ofrece un formato para la descripción de datos estructurados, facilitando declaraciones de contenido más precisas y resultados de búsquedas más significativos en varias plataformas. XML permite compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes. (18)

### ➤ **JSON**

JSON, acrónimo de “JavaScript Object Notation”, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML. Es muy sencillo de usar, especialmente como alternativa a XML. Una de las ventajas de JSON sobre XML como formato de intercambio de datos es que es mucho más sencillo escribir un analizador semántico de JSON.

(19)

### ➤ **XHTML**

XHTML es el acrónimo en inglés de Extensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto). Es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de remplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML.

Está encaminado al uso de un etiquetado correcto, por lo que exige una serie de requisitos básicos a cumplir en cuanto al código.

Algunos de estos requisitos son:

- Elementos correctamente anidados.
- Etiquetas en minúsculas.
- Elementos cerrados correctamente.
- Atributos de valores entrecomillados. (20)

### ➤ **CSS**

Es un lenguaje de hojas de estilos en Cascada en sus siglas en ingles (Cascading Style Sheets) creado para controlar la presentación de documentos estructurados, aspectos como: el color, el tamaño, el tipo de letra, la separación entre párrafos y la tabulación con la que se muestran los elementos de una lista. El propósito del desarrollo de CSS es separar la estructura y el contenido de la presentación estética en un documento, esto permite un control mayor del documento y sus atributos, convirtiendo al HTML en un documento muy versátil y liviano.

Entre los beneficios concretos de CSS se encuentran:

- Control de la presentación de muchos documentos desde una única hoja de estilo.
- Control más preciso de la presentación. (21)

### ➤ **JavaScript**

Es un lenguaje basado en objetos y guiado por eventos, diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet. Los programas escritos con este lenguaje se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios, convirtiéndolo en un lenguaje interpretado. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario.

Ventajas de JavaScript:

- Los programas escritos en este lenguaje no requieren de mucha memoria ni tiempo adicional de transmisión, por ser pequeños y compactos.
- JavaScript no requiere un tiempo de compilación; ya que los scripts se pueden desarrollar en un período de tiempo relativamente corto.
- Es independiente de la plataforma hardware o sistema operativo, y funciona correctamente siempre y cuando exista un navegador con soporte JavaScript. (22)

### 1.4.3.2 ExtJS 2.0

La nueva versión ExtJS 2.0 contiene nuevos rasgos y características que los distinguen entre los que se encuentran el aumento considerable de la biblioteca. Existen además muchos rasgos adicionales y cambios arquitectónicos que hacen el funcionamiento con 2.0 más intuitivo y flexible que liberaciones anteriores.

Se determinan algunos rasgos únicos de esta versión 2.0 entre los que se encuentran:

- Columna de nivel sumamente configurable que agrupa capacidades así como resumen.
- Vistas de árbol que apoya columnas adicionales para cada nodo de hoja.
- Etiquetas de desplazamiento. (23)

### 1.4.3.3 PHP

PHP (Hypertext Processor) es un lenguaje script (no se compila para conseguir códigos máquina si no que existe un intérprete que lee el código y se encarga de ejecutar las instrucciones que contiene éste código), para el desarrollo de páginas web dinámicas del lado del servidor, Es un lenguaje Open Source (código abierto), es el más popular y extendido en la web. (24)

Entre las características que posee este lenguaje y que lo convierten en una potente herramienta están:

- Es un software de código abierto.
- Soporta muchas bases de datos entre las que se encuentran (MySQL y PostgreSQL).
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. (25)

### 1.4.4 Ambiente de desarrollo integrado (IDE)

El ambiente de desarrollo (Development Environment) es algo imprescindible en la producción de software. Es donde se definen el conjunto de herramientas y tecnologías (frameworks), versiones a usar y su integración, que intervienen en un proceso de desarrollo de software. Un entorno de desarrollo integrado o Integrated Development Environment (IDE), en inglés, es un programa compuesto por un

conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien, poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Los IDEs pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes. El lenguaje Visual Basic por ejemplo puede ser usado dentro de las aplicaciones de Microsoft Office, lo que hace posible escribir sentencias Visual Basic en forma de macros para Microsoft Word. Los IDEs proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Java, C#, Delphi, Visual Basic y Object Pascal. Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación. Este es el caso de Eclipse, que mediante pluggins se le puede añadir soporte de lenguajes adicionales. (26)

### 1.4.4.1 Zend Studio para Eclipse

El Zend Studio para Eclipse constituye la nueva generación del Zend Studio. Proporciona a las organizaciones que hagan uso del mismo ya que posee un entorno mucho más flexible y profesional para controlar todo el ciclo de vida de un desarrollo. Entre sus funcionalidades, destacaría las capacidades de refactorización del código fuente, funcionalidad que permite adecuar el comportamiento externo de una función/clase sin cambiar el funcionamiento interno, que junto a las capacidades de generación de código le facilitaría el trabajo a los desarrolladores.

Entre las nuevas características que incluye este IDE están:

- El acceso al ecosistema de plugins de Eclipse.
- Apoyar el desarrollo de múltiples idiomas.
- Mejora el Editor de PHP con el formato avanzado, para listas de tareas y problemas de vista.
- Mejora del soporte de JavaScript.
- Mejora de apoyo, incluyendo HTML, códigos plegables, arrastrar y soltar los componentes.
- Mejora de depuración y Perfiles con Path Mapping.
- Mejora de Zend Framework con el apoyo del Proyecto Framework, plantillas y código MVC. (27)

### 1.4.5 Framework

En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Un framework representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio. También podemos definirlo en el contexto de la programación como un set de funciones o código genérico que realiza tareas comunes y frecuentes en todo tipo de aplicaciones (creación de objetos, conexión a base de datos, limpieza del código, etc.). En general, los frameworks son construidos en base a lenguajes orientados a objetos. Además, en la mayoría de los casos un framework implementará uno o más patrones de diseño de software que aseguren la escalabilidad del producto. (28) (29)

#### 1.4.5.1 Zend Framework

El Zend Framework, es un framework con una arquitectura flexible con el que podremos construir sin demasiadas complicaciones potentes aplicaciones web. Además se trata de un framework para desarrollo de servicios Web con PHP, te brinda soluciones para construir sitios web modernos y seguros. Además es Open Source y trabaja con PHP 5.

Entre sus principales características se encuentran:

- Trabaja con MVC (Model View Controller).
- El Marco de Zend también incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar su base de datos, sin tener que escribir ninguna consulta SQL.
- Completa documentación y pruebas de alta calidad.
- Robustas clases para autenticación y filtrado de entrada. (30)

#### 1.4.5.2 Zend\_Ext Framework

Es un framework Open Source, que esta diseñado para PHP 5 y buenas capacidades de ampliación. Es elaborado a partir de Zend Framework cumpliendo con todas sus características. Este trae de novedoso

un controlador vertical para el control de las acciones realizada por las vistas hacia el controlador, un motor de reglas para las validaciones en el servidor, se le incluyó el IoC para la comunicación en entre los módulos o componentes. Se le incorporo la integración con el ORM Doctrine Framework para trabajo en la capa de abstracción a base de datos y el ExtJs Framework para el desarrollo de las vistas.

### **1.4.5.3 Doctrine Framework**

Es un potente y completo sistema ORM (Object Relational Mapper) para PHP 5.2+ que incorpora una (capa de abstracción a base de datos), en sus siglas en inglés (DBL). Posee la habilidad de escribir opcionalmente las preguntas de la base de datos orientado a objeto. Esto les brinda una alternativa poderosa a diseñadores de SQL que mantiene un máximo de flexibilidad sin requerir la duplicación del código innecesario. También permite exportar una base de datos existente a sus clases correspondientes y también convierte clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos. (31)

### **1.4.6 Servidor Web**

#### **1.4.6.1 Apache**

Entre sus principales características se encuentran:

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Es una tecnología gratuita de código fuente abierto.
- Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.
- Tiene una alta configurabilidad en la creación y gestión de logs.(40)

### **1.4.7 Sistema Gestor de Base de Datos (SGBD)**

#### **1.4.7.1 PostgreSQL**

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS), de código abierto. Ofrece incorporar los siguientes cuatro conceptos adicionales básicos de manera que los usuarios pueden extender fácilmente el sistema: clases, herencia, tipos, funciones. Otras características aportan

potencia y flexibilidad adicional: Restricciones (Constraints), Disparadores (triggers), Reglas (rules), Integridad transaccional. Debido a eso se considera la base de datos de código abierto más avanzada hoy día disponible, soporta casi toda la sintaxis SQL y cuenta también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, PHP, Java, perl, tcl y python). (33)

### **1.4.8 Herramientas de base de datos**

#### **1.4.8.1 EMS SQL Manager**

EMS SQL Manager es una herramienta de alto rendimiento para la administración de bases de datos PostgreSQL y el desarrollo. Funciona con cualquier versión de PostgreSQL hasta la más reciente y soporta las últimas características incluyendo PostgreSQL enumerados, búsqueda de texto, XML y tipos de arreglos de tipos compuestos. SQL Manager para PostgreSQL ofrece muchas de las poderosas herramientas de base de datos, como bases de datos visuales Designer, Visual Query Builder para crear consultas de PostgreSQL complicado. (34)

### **1.4.9 Navegadores**

#### **1.4.9.1 Mozilla Firefox**

Mozilla Firefox es un navegador de software libre. Entre sus características se encuentran que presenta una forma rápida y eficiente de navegar por la web, que le permite abrir varias páginas en una misma ventana mediante el empleo de pestañas separadas. Mozilla Firefox además protege al usuario de la publicidad de ventanas emergentes no solicitadas. Permite además realizar una búsqueda más eficiente teniendo incorporada la barra de Google que proporciona el acceso necesario al mejor buscador de la web. (35)

### **1.4.10 Control de versiones**

#### **1.4.10.1 Subversion**

Subversion es un sistema de control de versiones libre y de código fuente abierto, es decir, maneja ficheros y directorios a través del tiempo. Hay un árbol de ficheros en un repositorio central. El repositorio es como un servidor de ficheros ordinario, excepto porque recuerda todos los cambios hechos a sus ficheros y directorios. Esto le permite recuperar versiones antiguas de sus datos, o examinar el historial de cambios de los mismos. Subversion puede acceder al repositorio a través de redes, lo que le permite ser

usado por personas que se encuentran en distintos ordenadores. Entre las características que tienen estos sistemas están:

- Entendimiento con los lenguajes de programación.
- Suministro de herramientas para la construcción de software.
- Subversion es un sistema general que puede ser usado para administrar cualquier conjunto de ficheros. (36)

### ➤ **TortoiseSVN**

Es un cliente de código abierto para el sistema de control de versiones Subversion. Maneja ficheros y directorios a lo largo del tiempo. Es fácil de usar, permite ver el estado de los archivos desde en el explorador de Windows y permite también crear gráficos de todas las revisiones asignadas. (36)

## **Conclusiones**

En la elaboración de este capítulo se abordaron varios conceptos generales que servirán de guía para la realización de este trabajo, se realizó una descripción de cómo se gestionan los procesos referente a la gestión de los trabajadores así como se realizó un estudio y análisis sobre algunos sistemas existentes que gestionan estos procesos. Además se determinaron las tecnologías, lenguajes, herramientas y metodología a utilizar para el desarrollo de la solución propuesta, demostrando que estas son las idóneas.

### CAPÍTULO 2: MODELACIÓN DEL NEGOCIO Y ANÁLISIS

#### **Introducción**

En el siguiente capítulo se realizará una descripción de los procesos del negocio así como se mostrará el comportamiento e interacción de los procesos de negocio con sus respectivas entradas y salidas mediante el mapa de procesos. Además se hará una representación del modelo conceptual para definir en un ámbito generalizado las relaciones entre las entidades del negocio. Para determinar los requerimientos que tendrá la solución a desarrollar se aplicaran métodos de identificación de requisitos. Por consiguiente quedarán expuestos cada unos de los requerimientos funcionales y la especificación de cada uno de ellos, así como se diseñarán los prototipos de interfaz de usuario para obtener una visión final de las interfaces. Además para validar los requisitos definidos se aplicará técnicas de validación de requisitos.

#### **2.1 Modelo del negocio**

En el ciclo de desarrollo del un producto de software la modelación del negocio es una disciplina de vital importancia ya que a través de esta se identifican los procesos de negocio correspondientes definiendo de esta manera las fronteras por las que se guiará el posterior proceso de modelado. Además servirá como base para identificar y definir los requisitos del sistema.

#### **2.2 Mapa de procesos**

El mapa de procesos es un artefacto que nos muestra la interacción que se establece entre los procesos de negocio fundamentales que han sido identificados con sus respectivas entradas y salidas.

El proceso que se identifico fue:

- Movimiento de la Fuerza de Trabajo.

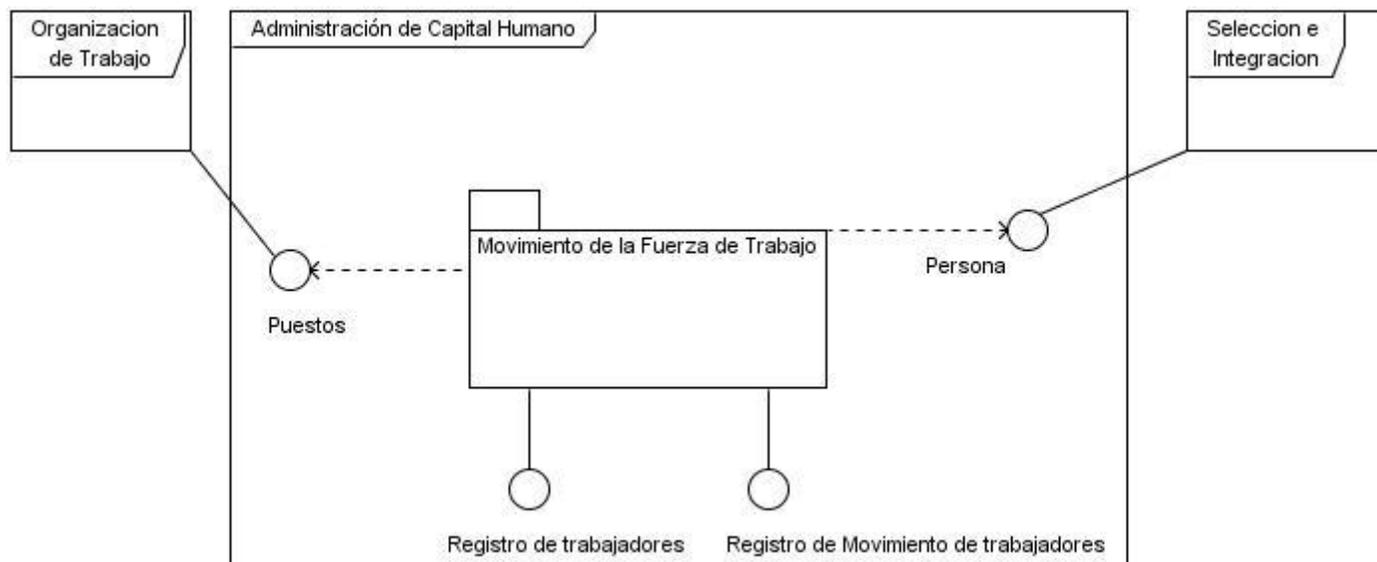


Figura 1 La figura muestra el mapa de procesos donde se define el proceso **Movimiento de Fuerza de Trabajo**

## 2.3 Procesos de negocio

Una parte de vital importancia dentro del modelado del negocio está la definición de los procesos del negocio y su descripción ya que permite tener una información más detallada de los mismos.

### 2.3.1 Descripción del proceso Movimiento de la Fuerza de Trabajo

Uno de los procesos claves dentro del subsistema de Administración de Capital Humano lo constituye la realización de los Movimientos de la Fuerza de Trabajo, lo cual permite mantener actualizado el registro de trabajadores y la actualización de la plantilla de personal. Los Movimientos de la Fuerza de Trabajo pueden clasificarse en tres subprocesos fundamentales: Altas, Reubicaciones y Bajas.

Para efectuar los movimientos de Altas es necesario que previamente se haya definido la estructura de la Entidad, los cargos, puestos de trabajo asociados a ellos, así como haberse registrado las personas que resultaron seleccionadas para ocupar un determinado cargo. El proceso de alta se resume en la selección de la persona a la que se desea realizar la formalización de la relación laboral y la asociación a ella de un puesto de trabajo que se encuentre vacante, también se registran los pagos adicionales asociados a dicha persona y un grupo de datos laborales necesarios para la elaboración del contrato de trabajo y el procesamiento de las nóminas.

Una vez formalizada la relación laboral del trabajador, el mismo puede ser objeto de movimientos de Reubicaciones o Bajas en dependencia de estudios organizacionales realizados en la entidad, aplicación de medidas disciplinarias o cualquier otra causa que implique variaciones en los elementos de su salario o cambios de puestos de trabajo o cargos. Los movimientos de reubicaciones pueden ser permanentes o temporales y en los dos casos deben especificarse las causas que motivaron dichos movimientos.

Una condición obligatoria para efectuar un movimiento de baja es que la persona haya sido registrada como trabajador, estos tipos de movimientos se producen por cualquiera de las causas especificadas en la legislación laboral vigente para este caso.

En todos los Movimientos de Fuerza de Trabajo debe especificarse obligatoriamente la fecha efectiva a partir de la cual es valido el cambio realizado, así como los motivos que generan a los mismos.

## 2.4 Modelo conceptual

En el siguiente modelo conceptual del componente trabajador del subsistema Capital Humano se representan las entidades que lo conforman y sus atributos, estableciendo una representación de los conceptos del dominio del problema dada las asociaciones y relaciones que se determinan entre ellas.

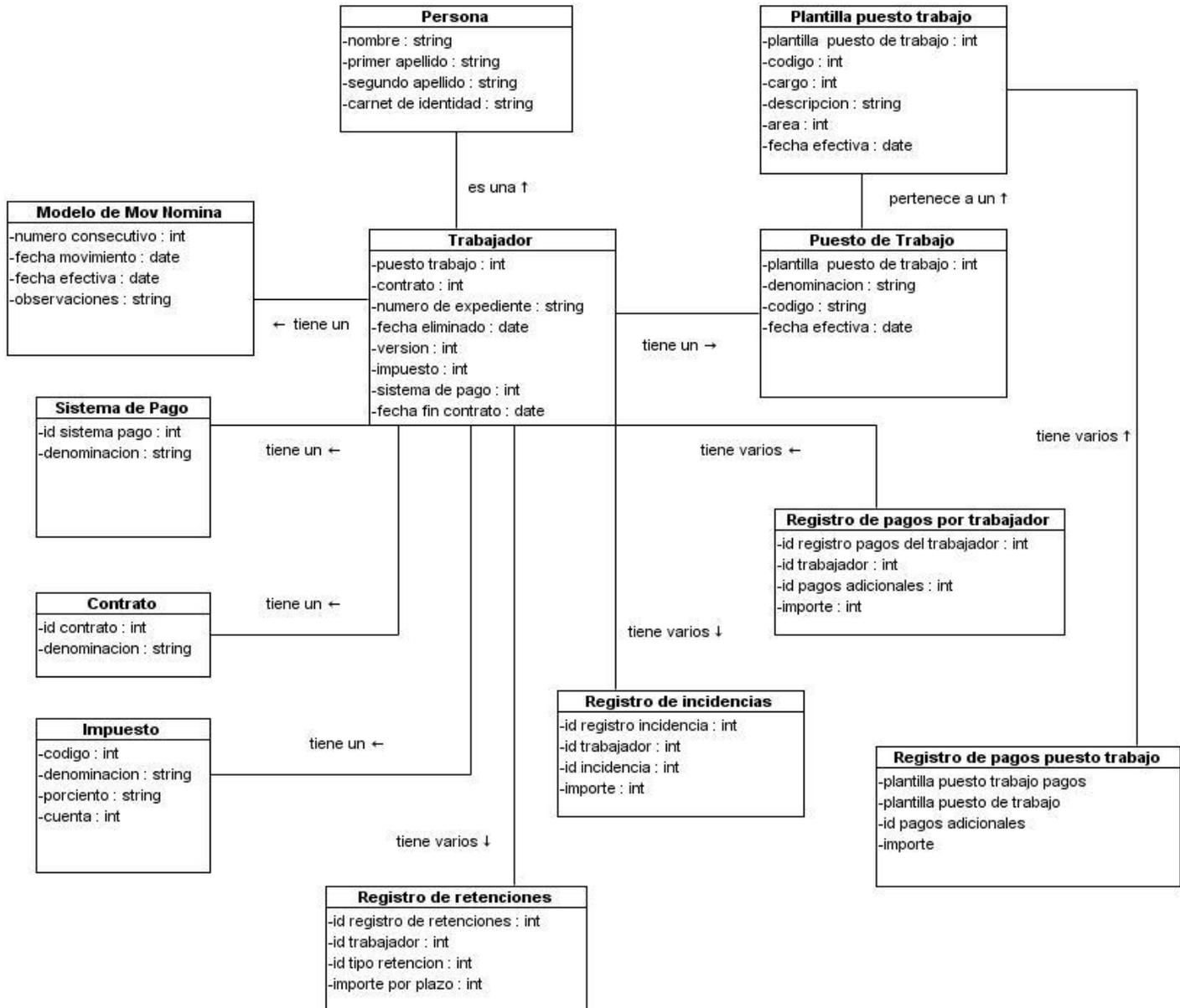


Figura 2 Representa el modelo conceptual

## 2.5 Métodos utilizados para identificar los requerimientos

Los métodos que se utilizaron para identificar los requisitos de sistema son:

- **Tormentas de ideas:** En varias ocasiones se reunieron los analistas del subsistema Capital Humano y el equipo de desarrolladores en conjunto con los especialistas funcionales y cada involucrado expresó su opinión de las funcionalidades que debía tener el futuro sistema.
- **Revisión de la documentación:** Los analistas revisaron y analizaron la documentación de la descripción de los procesos de negocio para derivar de estos los requisitos que debía tener el sistema.

## 2.6 Identificación de los requerimientos funcionales

Teniendo en cuenta los métodos que se aplicaron para identificar los requerimientos y los procesos del negocio anteriormente descritos se llegó a la conclusión de que los requisitos con que debe contar el sistema son los siguientes:

### 2.6.1 Requerimientos funcionales

R1 – Emitir movimiento nómina

R1.1 Emitir movimiento de altas

R1.2 Emitir movimiento de reubicación

R1.3 Emitir movimiento de bajas

R2 Actualizar datos contables

R3 Buscar trabajador

R4 Buscar fonéticamente un trabajador

R5 Filtrar datos del trabajador

R6 Mostrar registro de los trabajadores

## 2.7 Especificación de requerimientos

El propósito de la especificación de los requisitos es realizar una descripción detallada de cada uno de los requisitos funcionales definidos con el objetivo de entender cada una de las funcionalidades asociadas a los procesos del negocio así como determinar el alcance del sistema.

### 2.7.1 Requisito funcional Emitir movimiento de nómina

Permite informar todo el movimiento de personal que se realice y produzca modificaciones en su estructura salarial, cargo o área de trabajo, constituyendo el documento que respalda las anotaciones para mantener actualizados el registro de trabajadores que sirve de fuente para la preparación y pago de las nóminas.

- Prototipo de Interfaz de Usuario Emitir movimiento de nómina. **Anexo1**

### 2.7.2 Requisito funcional Movimiento de altas

Tabla 1 Especificación del requisito Movimiento de altas

Conceptos tratados	Conceptos	Atributos
	Trabajador	Nombre(s), primero y segundo apellidos del trabajador, expediente laboral y número de su expediente interno, sistema de pago, pagos adicionales asociados al trabajador, tipo de contrato, fecha fin del contrato.
	Movimiento de nómina.	Número consecutivo, fecha efectiva, motivo, observaciones.
	Puesto de trabajo.	Área, grupo de trabajo, puestos de trabajo, denominación, pagos asociados al puesto de trabajo.

<b>Precondiciones</b>	<b>Precondiciones</b>	<b>Pre-requisito</b>
	<p>Deben existir personas en la base de datos.</p> <p>Debe estar definida la plantilla de la empresa en estructura y composición.</p> <p>Deben agregarse puestos de trabajo.</p> <p>Debe definirse nomenclador de motivo del movimiento.</p> <p>Debe definirse nomenclador de sistema de pago de la empresa.</p>	<p>Buscar trabajador</p>
<b>Descripción</b>	<p>Seleccionar Movimiento de alta.</p> <p>Inmediatamente de haber seleccionado movimiento de alta el sistema permitirá buscar a la persona por los siguientes criterios:</p> <p>Nombre, primer apellido, segundo apellido, carnet de identidad.</p> <p>El sistema muestra las personas que coinciden con el criterio empleado. Si la búsqueda no genera resultados, se notifica al usuario.</p> <p>Escoger la persona.</p> <p>El sistema muestra el nombre y los apellidos y carnet de identidad del mismo</p>	

	<p>y permite registrar los datos del movimiento de alta, los cuales son:</p> <p><b>Fecha efectiva:</b> Esta es la fecha a partir de la cual se pone en vigor el alta al trabajador.</p> <p><b>Expediente interno:</b> Un número que define la empresa.</p> <p><b>Expediente laboral:</b> Sería el carnet de identidad de la persona, que pasaría a ser el expediente interno del trabajador.</p> <p><b>Tipo de contrato:</b> Los contratos son determinados, indeterminados, a domicilio y aprendizaje.</p> <p><b>Fecha fin:</b> Esta es la fecha fin del contrato la cual no se activa si el contrato es indeterminado.</p> <p><b>Número consecutivo:</b> es el número consecutivo que tiene el movimiento es entrado por el usuario.</p> <p><b>Motivo movimiento:</b> Aquí aparecerán los motivos por los cuales se puede dar alta a una persona.</p> <p><b>Área de trabajo:</b> Escoger el área de trabajo de las definidas en la empresa.</p> <p><b>Grupo de trabajo:</b> Grupo de trabajo al cual va a estar asignado ese trabajador.</p> <p><b>Puesto de trabajo:</b> Puesto que ocupará una vez escogida el área y el grupo de trabajo. Cuando selecciona el puesto de trabajo, automáticamente salen el cargo que ocupará y con él la categoría ocupacional, grupo escala, salario escala y la contribución la seguridad social. También cuando se selecciona el puesto de trabajo, automáticamente aparecen los pagos adicionales que hay asociados a ese puesto.</p> <p><b>Sistema de pago:</b> Escoger entre los sistemas de pagos definidos, los</p>
--	--

	<p>cuales son Tarifa horaria, Jornal diario, Sueldo mensual.</p> <p><b>Pagos adicionales asociados el trabajador:</b> Se adicionan los pagos adicionales que vienen con el trabajador, si se desea se puede modificar el importe del pago que se quiera.</p> <p><b>Pagos adicionales asociados al puesto de trabajo:</b></p> <p>Se muestra un listado de los pagos adicionales asociados al puesto de trabajo que ocupará el trabajador.</p> <p><b>Observaciones:</b> Se agrega algo que falte y que se considere importante.</p> <p>Se validan los datos.</p> <p>Si los datos son correctos se realiza y se registra el movimiento de alta al trabajador y se notifica al usuario. El trabajador al que se le realizó el alta pasa a estado activo.</p> <p>Si los datos no son correctos se notifica al usuario y se permite corregirlos.</p>
<b>Validaciones</b>	Se validan según el ERP-CH-Modelo conceptual v1.7.doc.
<b>Post-condiciones</b>	<p>Se asoció un puesto de trabajo y un cargo a la persona.</p> <p>Se creó el modelo de movimiento de alta a la nómina.</p> <p>Se actualizó el registro de trabajadores.</p> <p>Se modificaron pagos adicionales y se asociaron al trabajador.</p>
<b>Post-requisito</b>	Actualizar datos contables.

➤ Prototipo de Interfaz de Usuario del requisito Movimiento de altas. **Anexo 2**

### 2.7.3 Requisito funcional Movimiento de reubicación

Tabla 2 Especificación del requisito Movimiento de reubicación

Conceptos tratados	Conceptos	Atributos
	Trabajador	Sistema de pago, pagos adicionales asociados al trabajador, tipo de contrato, fecha fin del contrato.
	Movimiento de nómina.	Número consecutivo, fecha efectiva, fecha del movimiento, puesto de trabajo, motivo, fecha fin, observaciones.
	Puesto de trabajo.	Área, grupo de trabajo, puesto de trabajo.
Precondiciones	Precondiciones	Pre-requisito
	<p>Debe existir al menos un trabajador en el maestro de trabajadores para poder realizarle el movimiento de reubicación.</p> <p>Debe estar definido el nomenclador de motivo de la reubicación.</p> <p>Debe estar definido el nomenclador de</p>	<p>Movimiento de altas.</p> <p>Adicionar puesto de trabajo.</p>

	Carácter de la reubicación.	
<b>Descripción</b>	<p>Seleccionar Movimiento de Reubicación.</p> <p>El sistema permitirá realizar una búsqueda avanzada de los trabajadores por los siguientes criterios:</p> <p>Expediente laboral, expediente interno, tarjeta, nombre, primer apellido, segundo apellido, área, cargo, centro de costo, tipo contrato, estado (activo/inactivo).</p> <p>El sistema mostrará de manera automática los datos anteriores a la reubicación del trabajador que se seleccionó.</p> <p>Se especifica el tipo de reubicación que se va a llevar a cabo, en caso de ser permanente se activa lo siguiente:</p> <p><b>Fecha efectiva:</b> A partir de cuando se va a realizar el cambio.</p> <p><b>Tipo de contrato:</b> Los contratos son determinados, indeterminados, a domicilio y aprendizaje.</p> <p><b>Fecha fin del contrato:</b> Esta es la fecha fin del contrato la cual no se activa si el contrato es indeterminado.</p> <p><b>Motivo:</b> Motivo por el cual se le realiza la reubicación.</p> <p><b>Número consecutivo:</b> es el número consecutivo que tiene el movimiento es entrado por el usuario.</p>	

	<p><b>Área de trabajo:</b> Aparecerá el área anterior que ocupaba el trabajador, con la posibilidad de escoger una nueva área de trabajo de las definidas en la empresa.</p> <p><b>Grupo de trabajo:</b> Aparecerá el grupo de trabajo que tenía antes el trabajador y se da la posibilidad de cambiar el grupo de trabajo si es necesario.</p> <p><b>Puesto de trabajo:</b> Aparecerá el puesto que ocupó anteriormente, si se desea se puede cambiar el puesto de trabajo sin necesidad de cambiar el área de trabajo anterior. Si se realizan cambios en el puesto de trabajo automáticamente salen el cargo que ocupará y con él la categoría, grupo escala, salario escala y la contribución la seguridad social. También cuando se selecciona el puesto de trabajo, automáticamente aparecen los pagos adicionales que hay asociados a ese puesto.</p> <p><b>Sistema de pago:</b> Escoger entre los sistemas de pagos definidos, los cuales son Tarifa horaria, Jornal diario, Sueldo mensual, en caso que no hayan cambios en el sistema de pago se queda el que estaba anterior al movimiento de reubicación.</p> <p><b>Pagos adicionales asociados el trabajador:</b> Se modifican o se adicionan los pagos adicionales asociados al trabajador.</p> <p><b>Observaciones:</b> Se agrega algo que falte y que se considere importante.</p> <p>Pagos asociados al puesto de trabajo: Aquí saldrán los pagos asociados al puesto de trabajo que se seleccionó.</p> <p>Si la reubicación es temporal se mantienen los datos anteriores activados y se llenan de la misma manera que la reubicación permanente, el único cambio que se efectuará es:</p> <p><b>Fecha fin:</b> Esta es la fecha en la que se termina la reubicación</p>
--	--

	<p>temporal.</p> <p>Se validan los datos.</p> <p>Si los datos son correctos se realiza y se registra el movimiento reubicación al trabajador y se notifica al usuario.</p> <p>Si los datos no son correctos se notifica al usuario y se permite corregirlos.</p>
<b>Validaciones</b>	<p>Se valida que se haya seleccionado un trabajador cuando se va a modificar.</p> <p>Se validan según el ERP-CH-Modelo conceptual v1.7.doc.</p>
<b>Post-condiciones</b>	<p>En caso de haberse modificado algún dato se crea un modelo de movimiento de nóminas de tipo reubicación y en caso contrario no se registra el movimiento y se notifica al usuario.</p>
<b>Post-requisito</b>	<p>Actualizar datos contables.</p>

➤ Prototipo de Interfaz de Usuario del requisito Movimiento de reubicación. **Anexo 3**

### 2.7.4 Requisito funcional Movimiento de bajas

Tabla 3 Especificación del requisito Movimiento de bajas

<b>Conceptos tratados</b>	<b>Conceptos</b>	<b>Atributos</b>
	Trabajador.	Sistema de pago, pagos adicionales asociados al trabajador, tipo de contrato, puesto de trabajo
	Movimiento de nómina.	Fecha efectiva, fecha del movimiento, observaciones., tipo de movimiento, motivo

Precondiciones	Precondiciones	Pre-requisito
	<p>Deben existir trabajadores en el maestro de trabajadores.</p>	<p>Movimiento de Altas.</p>
<p><b>Descripción</b></p>	<p>Seleccionar Movimiento de baja.</p> <p>El sistema permitirá realizar una búsqueda avanzada de los trabajadores por los siguientes criterios: Expediente laboral, expediente interno, tarjeta, nombre, primer apellido, segundo apellido, área, cargo, centro de costo, tipo contrato, estado (activo/inactivo).</p> <p>El sistema mostrará de manera automática los datos actuales del trabajador.</p> <p>Se completan los siguientes datos:</p> <p style="padding-left: 40px;"><b>Fecha efectiva:</b> Fecha a partir de cuando se realizara ese movimiento.</p> <p style="padding-left: 40px;"><b>Motivo:</b> Motivo por el cual se realizo la baja del trabajador.</p> <p style="padding-left: 40px;"><b>Número consecutivo:</b> es el número consecutivo que tiene el movimiento es entrado por el usuario.</p> <p style="padding-left: 40px;"><b>Observaciones:</b> En caso de existir alguna.</p> <p>Se validan los datos.</p> <p>Si los datos son correctos se realiza y se registra el movimiento de baja al trabajador y se notifica al usuario.</p> <p>Si los datos no son correctos se notifica al usuario y se permite corregirlos.</p> <p>El estado del trabajador seleccionado pasa a inactivo.</p>	
<p><b>Validaciones</b></p>	<p>Se validan según el ERP-CH-Modelo conceptual v1.7.doc.</p>	
<p><b>Post-condiciones</b></p>	<p>Se elimina el trabajador del registro de trabajadores.</p>	

	<p>Se eliminan los datos contables del trabajador.</p> <p>Se elimina el trabajador del submayor de vacaciones.</p> <p>Se actualiza el registro de personas en la bolsa de la entidad.</p>
<b>Post-requisito</b>	No procede.

➤ Prototipo de Interfaz de Usuario del requisito Movimiento de bajas. **Anexo 4**

### 2.7.5 Requisito funcional Actualizar datos contables

Tabla 4 Especificación del requisito Actualizar datos contables

<b>Conceptos tratados</b>	<b>Conceptos</b>	<b>Atributos</b>
	Trabajador.	Centro de costo, cuenta, número de tarjeta.
<b>Precondiciones</b>	<b>Precondiciones</b>	<b>Pre-requisito</b>
	Debe existir al menos un trabajador en el maestro de trabajadores.	Movimiento de Altas.
<b>Descripción</b>	<p>Seleccionar Actualizar datos contables. Aparece lo siguiente:</p> <p>El sistema permitirá realizar una búsqueda avanzada de los trabajadores por los siguientes criterios: Expediente laboral, expediente interno, tarjeta, nombre, primer apellido, segundo apellido, área, cargo, centro de costo, tipo contrato, estado (activo/inactivo).</p> <p>El sistema mostrará de manera automática algunos datos del trabajador que se seleccionó, los cuales son: nombre, primer apellido, segundo apellido, puesto de trabajo, cargo, grupo escala, categoría y salario escala.</p> <p>El sistema mostrará lo siguiente:</p>	

	<p><b>Datos contables:</b> Aquí se le asocia el trabajador con una cuenta y con un centro de costo; además se le agrega el número de tarjeta en caso de que el trabajador tenga.</p> <p><b>Tarjeta de banco:</b> En caso de tener tarjeta se le agrega.</p> <p>Liquida vacaciones acumuladas</p> <p><b>Fecha efectiva:</b> Fecha a partir de cuando se realizara la actualización.</p> <p>Se agregan los datos contables al trabajador que se seleccionó.</p> <p>Se validan los datos.</p> <p>Si los datos son correctos se realiza la actualización y se notifica al usuario.</p> <p>Si los datos no son correctos se notifica al usuario y se permite corregirlos.</p>
<b>Validaciones</b>	Se validan según el ERP-CH-Modelo conceptual v1.7.doc
<b>Post-condiciones</b>	Se actualizan los datos contables del trabajador
<b>Post-requisito</b>	No procede.

➤ Prototipo de Interfaz de Usuario del requisito Actualizar datos contables. **Anexo 5**

### 2.7.6 Requisito funcional del requisito Buscar trabajador

Tabla 5 Especificación del requisito Buscar trabajador

Conceptos tratados	Conceptos	Atributos
	Trabajador.	Nombre(s), primero y segundo apellidos del trabajador, No. Carnet, expediente laboral, número de su expediente interno, tarjeta, área, cargo, centro de costo, tipo de contrato.

<b>Precondiciones</b>	<b>Precondiciones</b>	<b>Pre-requisito</b>
	Debe existir al menos un trabajador en el maestro de trabajadores.	
<b>Descripción</b>	<p>Seleccionar Buscar trabajador. Aparece lo siguiente:</p> <p>El sistema permitirá realizar una búsqueda avanzada de los trabajadores por los siguientes criterios: Expediente laboral, expediente interno, tarjeta, nombre, primer apellido, segundo apellido, área, cargo, centro de costo, tipo contrato, estado (activo/inactivo).</p> <p>El sistema mostrará de manera automática algunos datos del trabajador que se seleccionó, los cuales son: consecutivo, expediente interno, Nro. de carnet, Nombre(s), primero apellido, segundo apellido, área, cargo.</p> <p>Se validan los datos.</p> <p>Si los datos son correctos finaliza la búsqueda y se muestra al usuario el resultado.</p> <p>Si los datos no son correctos se notifica al usuario y se permite realizar nuevamente la operación.</p>	
<b>Validaciones</b>	Se validan según el ERP-CH-Modelo conceptual v1.7.doc	
<b>Post-condiciones</b>	Se realiza la búsqueda del trabajador por los criterios establecidos.	
<b>Post-requisito</b>	No procede.	

➤ Prototipo de Interfaz de Usuario del requisito Buscar trabajador. **Anexo 6**

### 2.7.7 Requisito funcional Buscar fonéticamente un trabajador

Tabla 6 Especificación del requisito Buscar fonéticamente un trabajador

<b>Conceptos tratados</b>	<b>Conceptos</b>	<b>Atributos</b>
	Trabajador.	Nombre(s), primero y segundo apellidos del trabajador, No. Carnet, expediente laboral, número de su expediente interno, tarjeta, área, cargo, centro de costo, tipo de contrato.
<b>Precondiciones</b>	<b>Precondiciones</b>	<b>Pre-requisito</b>
	Debe existir al menos un trabajador en el maestro de trabajadores.	
<b>Descripción</b>	<p>Seleccionar Buscar trabajador. Aparece lo siguiente:</p> <p>El sistema permitirá realizar una búsqueda fonética de los trabajadores por los siguientes criterios: nombre, primer apellido, segundo apellido, No. Carnet, Expediente laboral, expediente interno, tarjeta, área, cargo, centro de costo, tipo contrato, estado (activo/inactivo).</p> <p>El sistema mostrará de manera automática algunos datos del trabajador que se seleccionó, los cuales son: expediente laboral, Nombre(s), primero y segundo apellidos del trabajador.</p>	
<b>Validaciones</b>	Se validan según el ERP-CH-Modelo conceptual v1.7.doc	

<b>Post-condiciones</b>	Se realiza la búsqueda del trabajador por los criterios establecidos.
<b>Post-requisito</b>	No procede.

### 2.7.8 Requisito funcional Filtrar datos de los trabajadores

Tabla 7 Especificación del requisito Filtrar datos de los trabajadores

<b>Conceptos tratados</b>	<b>Conceptos</b>	<b>Atributos</b>
	Trabajador.	Nombre(s), primero y segundo apellidos del trabajador, No. Carnet, expediente laboral, número de su expediente interno, tarjeta, área, cargo, centro de costo, tipo de contrato.
<b>Precondiciones</b>	<b>Precondiciones</b>	<b>Pre-requisito</b>
	Debe existir al menos un trabajador en el maestro de trabajadores.	
<b>Descripción</b>	<p>Seleccionar Filtrar. Aparece lo siguiente:</p> <p>El sistema permitirá filtrar los datos de los trabajadores dado los siguientes criterios: nombre, primer apellido, segundo apellido, No. Carnet, Expediente laboral, expediente interno, tarjeta, área, cargo, centro de costo, tipo contrato, estado (activo/inactivo).</p> <p>El sistema mostrará de manera automática algunos datos del trabajador que se seleccionó, los cuales son: consecutivo, expediente interno, Nro. de carnet, Nombre(s), primero apellido, segundo apellido, área, cargo.</p>	

	Se muestra el resultado al usuario.
<b>Validaciones</b>	Se validan según el ERP-CH-Modelo conceptual v1.7.doc
<b>Post-condiciones</b>	Se filtran los datos del trabajador por los criterios establecidos.
<b>Post-requisito</b>	No procede.

### 2.7.9 Requisito funcional Mostrar registro de los trabajadores

Tabla 8 Especificación del requisito Mostrar registro de los trabajadores

<b>Conceptos tratados</b>	<b>Conceptos</b>	<b>Atributos</b>
	Trabajador.	Nombre(s), primero y segundo apellidos del trabajador, No. Carnet, expediente laboral, número de su expediente interno, tarjeta, área, cargo, centro de costo, tipo de contrato.
<b>Precondiciones</b>	<b>Precondiciones</b>	<b>Pre-requisito</b>
	Deben trabajadores en el maestro de trabaja	
<b>Descripción</b>	Seleccionar Emitir movimiento nómina. Aparece lo siguiente:  El sistema mostrará de manera automática algunos datos del trabajador los cuales son: consecutivo, expediente interno, Nro. Carnet, Nombre(s), primero y segundo apellido, área, cargo.	
<b>Validaciones</b>	Se validan según el ERP-CH-Modelo conceptual v1.7.doc	

<b>Post-condiciones</b>	Se muestra el registro de los trabajadores.
<b>Post-requisito</b>	No procede.

## 2.8 Técnicas utilizadas para validar los requerimientos

Una vez que se definen los requisitos, estos son validados. Por consiguiente la validación de los requisitos tiene como objetivo demostrar que la definición de los requisitos propone realmente el sistema que el usuario necesita. Por lo tanto se debe asegurar que el análisis realizado y los resultados obtenidos en la etapa de definición de los requisitos sean correctos.

Las técnicas que se aplicaron para validar los requerimientos fueron:

- **Prototipo de interfaz de usuario:** Se basó fundamentalmente en la obtención de prototipos de la definición de requisitos que, sin tener la totalidad de la funcionalidad del sistema, permitieron al usuario hacerse una idea de la estructura de la interfaz del sistema. Una vez identificados y especificados cada uno de los requisitos se utilizó como herramienta de modelado Visual Paradigm para diseñar los prototipos de interfaces. Para realizar un prototipo de interfaz es necesario poder contar con una especificación lo más clara y precisa posible, pues las mismas son un reflejo del futuro sistema.
- **Diseño de casos de prueba:** Esta técnica fue una de las más utilizadas pues un buen diseño de casos de pruebas comprueba que la especificación de requisitos se realizó con éxito. Los casos de prueba se hacen teniendo en cuenta la especificación de los requerimientos. Por cada uno de ellos se elaboraron casos de prueba.

### **Conclusiones**

Mediante el desarrollo de este capítulo se realizó la modelación del negocio con el objetivo de lograr una visión y un mejor entendimiento de los procesos y subprocesos definidos. Además se realizó un modelo conceptual a través de cual quedaron expuestos cada uno de los conceptos del dominio del problema, dada las entidades y sus relaciones. Se establecieron cada uno de los requisitos funcionales incluyendo su especificación detallada, con los que el sistema propuesto debe cumplir.

### CAPÍTULO 3: DISEÑO DEL SISTEMA

#### **Introducción**

En este capítulo se abordarán los temas referentes al diseño, se modelarán los artefactos que permitirán dar continuación a la implementación del sistema, como los diagramas de clases del diseño a los cuales se les aplican patrones de diseño para hacer el trabajo más eficiente y el modelo de datos representando las tablas de la base de datos vinculadas a la propuesta de solución, así como sus relaciones. También se hará un estudio del diseño propuesto a través de métricas que evalúan el diseño.

#### **3.1 Modelo de diseño del sistema**

El modelo de diseño del componente Trabajador del subsistema Capital Humano del sistema integral Cedrux, representa una continuación del modelo de análisis. Sirve además de abstracción de la implementación del sistema y es, de ese modo, utilizado como una entrada fundamental de las actividades de implementación. Este modelo contiene clases, paquetes, subsistemas, interfaces, relaciones, colaboraciones y atributos. Para la realización del modelo de diseño de la solución propuesta el equipo de trabajo se basó en los estándares definidos en el marco del proyecto del ERP.

##### **3.1.1 Aplicación de los patrones**

Para realizar un diseño más eficiente se utilizaron un conjunto de patrones, que al ser experiencias de diseñadores expertos en orientación a objetos permiten dar solución a problemas a través de la codificación del conocimiento y principios existentes, facilitando notablemente el trabajo posterior.

Cada patrón describe un problema que ocurre una y otra vez en nuestro ambiente, y luego describe el núcleo de la solución a ese problema.

Los patrones que se utilizaron fueron: el patrón arquitectónico Modelo – Vista – Controlador (MVC), los patrones de diseño, GRASP para la asignación de responsabilidades y Patrones de Comportamiento.

##### **3.1.1.1 Patrón Arquitectónico Modelo Vista Controlador (MVC)**

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la lógica de negocio.

Descripción del patrón:

- Modelo: Es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.
- Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.



**Figura 3 Funcionamiento del patrón MVC**

Ejemplo de utilidad en nuestro sistema del patrón MVC para el caso de que el cliente desee hacer un movimiento de alta a un trabajador:

- El usuario interactúa con la interfaz seleccionando la opción **Alta**.
- El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario.
- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción **Alta**, solicitada por el usuario.
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, se carga un listado con los tipos de contratos que se le harán al trabajador, un listado con las aéreas de trabajo a que se pueden ubicar).
- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

### 3.1.1.2 Patrones GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

**GRASP** es un acrónimo que significa **General Responsibility Assignment Software Patterns** (patrones generales de software para asignar responsabilidades). El nombre se eligió para indicar la importancia de **captar (grasping)** estos principios, si se quiere diseñar eficazmente el software orientado a objetos. (38)

Los patrones GRASP que se utilizan son:

- Creador: trata de resolver el problema de diseño de quién debería ser responsable de crear una nueva instancia de alguna clase. La creación de objetos es una de las actividades más frecuentes en un sistema orientado a objetos. En consecuencia, conviene contar con un principio general para asignar las responsabilidades concernientes a ella. El diseño, bien asignado, puede soportar un bajo acoplamiento, una mayor claridad, el encapsulamiento y la reutilizabilidad.

Para dar solución a esta problemática, el patrón creador plantea asignarle a la clase B la responsabilidad de crear una instancia de clase A en uno de los siguientes casos:

1. B agrega los objetos A.
2. B contiene los objetos A.
3. B registra las instancias de los objetos A.
4. B utiliza específicamente los objetos A.
5. B tiene los datos de inicialización que serán transmitidos a A cuando este objeto sea creado (así que B es un Experto respecto a la creación de A).
6. B es un creador de los objetos A.

Si existe más de una opción, prefiera la clase B que agregue o contenga la clase A. Este patrón es sumamente básico, y su solución en muchos casos no cubre los problemas que se presentan actualmente en el diseño de persistencia de software, pero es la base de todas las demás familias de patrones y framework relacionados con la creación de objetos; por su importancia ha sido incluido en este trabajo (LARMAN 1998).

- Alta Cohesión: trata de resolver el problema de diseño de cómo mantener la complejidad dentro de límite manejable.

Para dar solución a esta problemática, el patrón alta cohesión plantea asignar una responsabilidad a cada clase o grupo de clases según su contexto que no asuma responsabilidades fuera de su dominio de modelación (ejemplo: si la clase `DatTrabajador` esta modelando la gestión de trabajadores no tiene por qué recibir responsabilidades de la clase `DatModelomovnomina`), de esta forma se logra una alta cohesión en el sistema. La alta cohesión al igual que el bajo acoplamiento facilita una diseño mantenible, fácil de reutilizar y adaptar, más legibilidad para los programadores y diseñadores en general; la extensibilidad y flexibilidad del diseño aumentan, ya que las clases usan solo lo que necesitan y en su mayoría son recursos propios, por otro lado, cada cual hace lo que le corresponde hacer según la parte del problema que esta modelando (LARMAN 1998).

- Bajo Acoplamiento: trata de resolver el problema de diseño de cómo dar soporte a una dependencia escasa y a un aumento de la reutilización.

Para dar solución a esta problemática, el patrón bajo acoplamiento plantea asignar una responsabilidad a cada clase o grupo de clases según su contexto que no involucre recursos concurrentes de otras clases que no están enmarcadas en este contexto de modelación del problema, para mantener de este modo bajo el acoplamiento en el diseño (LARMAN 1998). Un ejemplo de ello son las integraciones del componente `Trabajador` con otros componentes, el mismo no conoce las clases existentes en los otros componentes pero mediante el IOC para obtener las funcionalidades necesarias, se logra una gran dependencia y reutilización.

### 3.1.1.3 Patrones de comportamiento

Se recurren a patrones de comportamiento en el diseño del sistema, los mismos estudian las relaciones de llamadas diferentes objetos, normalmente ligados con la dimensión temporal.

Los patrones de comportamiento que se utilizan son:

- Cadena de responsabilidad: Permite establecer la línea que deben llevar los mensajes para que los objetos realicen la tarea indicada.

El patrón cadena de responsabilidad es muy utilizado en todas las aplicaciones modernas dado que es uno de los principios que aplicamos en el tratamiento de excepciones. Un ejemplo del mismo es cuando se produce un error al adicionar un trabajador en la base de datos este es captado por las

capas superiores y estas reenvían la excepción hasta la capa de aplicación donde traduce al lenguaje del usuario y se le informa.

– Estado: Permite que un objeto modifique su comportamiento cada vez que cambie su estado interno.

Este patrón es utilizado por ejemplo, en las validaciones realizadas utilizando el `TrabajadorService` dentro del componente `Trabajador`, donde se llama un grupo de funciones según sea la validación que se quiere realizar teniéndose en tiempo de ejecución la construcción de un nuevo conjunto de operaciones.

– Estrategia: Permite disponer de varios métodos para resolver un problema y elegir cuál utilizar en tiempo de ejecución.

El uso del patrón estrategia, se pone de manifiesto en las clases controladoras, un ejemplo de esto es en la clase controladora `emitir movimiento de reubicación` donde se utilizan métodos definidos en las clases del dominio según las operaciones a realizar en el movimiento de reubicación.

### 3.1.2 Diagramas de clases del diseño

Los diagramas de clases del diseño referente al componente `Trabajador` del subsistema `Capital Humano` del sistema `CedruX`, describen gráficamente las clases y de las interfaces de la aplicación propuesta.

Estos diagramas contienen la siguiente información:

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Información sobre los tipos de los atributos.
- Navegabilidad.
- Dependencias.

En el ([Anexo 7](#)) se muestran los diagramas de clases de diseño asociado al componente `Trabajador`.

### 3.2 Modelo de datos

El modelo de datos propuesto es una representación de las tablas existentes en la base de datos así como las relaciones entre ellas. El mismo cuenta con diez tablas que representan de manera general el negocio del componente `Trabajador` del subsistema `Capital Humano` del sistema `CedruX`.

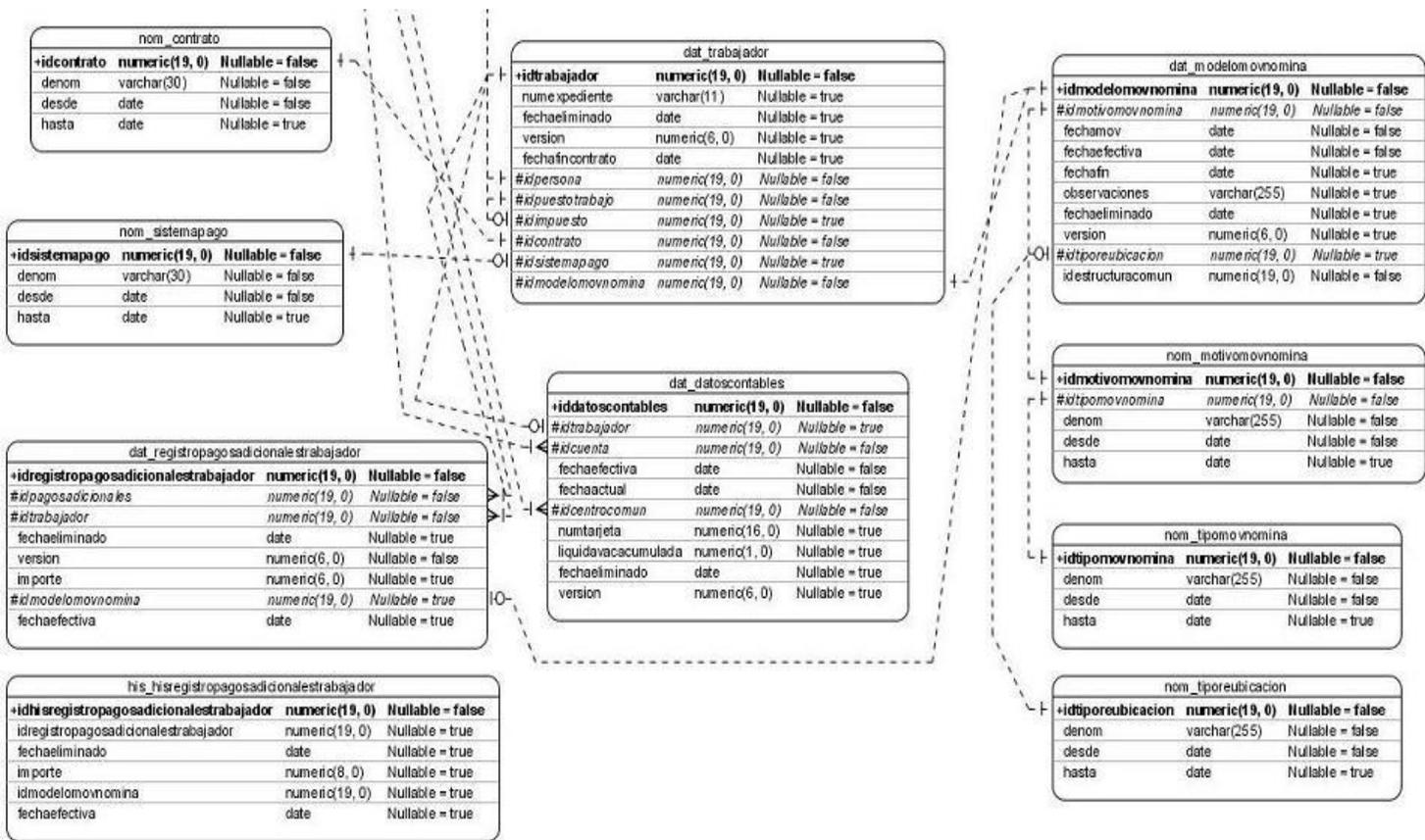


Figura 4 Modelo de datos del componente Trabajador

### 3.2.1 Descripción de las tablas de la base de datos

En este epígrafe se representa la descripción de las tablas de la base de datos relacionadas con el componente Trabajador.

Tabla 9 Descripción de la tabla dat\_trabajador

Nombre: dat\_trabajador

<b>Descripción:</b> Almacena los Trabajadores.		
Atributo	Tipo	Descripción
idtrabajador	numeric(19,0)	Identificador de Trabajador
idpersona	numeric(19,0)	Identificador de Persona
idpuestotrabajo	numeric(19,0)	Denominación de Puesto de Trabajo
idcontrato	numeric(2,0)	Identificador de Contrato
numexpediente	varchar (11)	Numero de Expediente
fechaeliminado	date	Se utiliza para no eliminar físicamente de la Base de Datos
version	numeric(6,0)	Se utiliza para la Concurrencia.
idimpuesto	numeric(19,0)	Identificador de Impuesto
idmodelomovnomina	numeric(19,0)	Identificador de Modelo de Movimiento de Nómina
idsistemapago	numeric(19,0)	Identificador de Sistema de Pago
fechafincontrato	date	Fecha de fin de contrato

**Tabla 10 Descripción de la tabla dat\_modelomovnomina**

<b>Nombre:</b> dat_modelomovnomina		
<b>Descripción:</b> Almacena los Movimientos de nómina.		
Atributo	Tipo	Descripción
idmodelomovnomina	numeric(19,0)	Identificador del Modelo de Movimiento de Nómina
idmotivomovnomina	numeric(19,0)	Identificador del Motivo de Movimiento de Nómina
fechamov	date	Fecha en que se realiza el Movimiento
fechaefectiva	date	Fecha a partir de la cual se hará vigente el Movimiento
fechafin	date	Fecha donde termina el Movimiento
observaciones	varchar(255)	Observaciones acerca del Movimiento de Nómina
fechaeliminado	date	Se utiliza para no eliminarlo físicamente de la Base de Datos
version	numeric(6,0)	Utilizado para la Concurrencia
idtiporeubicacion	numeric(19,0)	Identificador del Tipo de Reubicación
idestructuracomun	numeric(19,0)	Identificador de Entidad (lo provee Estructura y Composición)

Tabla 11 Descripción de la tabla nom\_motivomovnomina

<b>Nombre:</b> nom_motivomovnomina		
<b>Descripción:</b> Almacena los motivos de movimiento de nóminas.		
Atributo	Tipo	Descripción
idmotivomovnomina	numeric(19,0)	Identificador de Motivo de Movimiento de Nómina
idtipomovnomina	numeric(19,0)	Identificador de Movimiento de Nómina
denom	varchar(255)	Denominación del motivo de movimiento de nómina
desde	date	Fecha en la que se inserta el motivo de movimiento de nómina
hasta	date	Fecha en la que se elimina el motivo de movimiento de nómina

**Tabla 12 Descripción de la tabla nom\_tipomovnomina**

<b>Nombre:</b> nom_tipomovnomina		
<b>Descripción:</b> Almacena los tipos de movimiento de <u>nóminas</u> .		
Atributo	Tipo	Descripción
idtipomovnomina	numeric(19,0)	Identificador del tipo de movimiento de nómina
denom	varchar(255)	Descripción del tipo de movimiento de nómina
desde	date	Fecha en la que se inserta el tipo de movimiento de nómina
hasta	date	Fecha en la que se elimina el tipo de movimiento de nómina

**Tabla 13 Descripción de la tabla nom\_tiporeubicacion**

<b>Nombre:</b> nom_tiporeubicacion		
<b>Descripción:</b> Almacena los tipos de reubicación.		
Atributo	Tipo	Descripción
idtiporeubicacion	numeric(19,0)	Identificador del tipo de reubicación
denom	varchar(255)	Descripción del tipo de reubicación
desde	date	Fecha en la que se inserta el tipo de reubicación
hasta	date	Fecha en la que se elimina el tipo de reubicación

**Tabla 14 Descripción de la tabla dat\_datoscontables**

<b>Nombre:</b> dat_datoscontables		
<b>Descripción:</b> Almacena los datos contables de los trabajadores.		
Atributo	Tipo	Descripción
iddatoscontables	numeric(19,0)	Identificador de Datos Contables
idtrabajador	numeric(19,0)	Identificador del Trabajador
idcuenta	numeric(19,0)	Identificador de Cuenta (lo Provee Contabilidad)
fechaefectiva	date	Fecha a partir de que va a ser efectivo el cambio
fechaactual	date	Fecha actual del sistema

idcentrocomun	numeric(19,0)	Centro de Costo del Trabajador (lo provee Costos y Procesos)
numtarjeta	numeric(16,0)	Número de Tarjeta del Trabajador
liquidavacacumulada	numeric(1,0)	Si Liquida vacaciones acumuladas o no(0- si no liquida, 1- si liquida)
fechaeliminado	date	Se utiliza para no eliminarlo físicamente de la Base de Datos
version	numeric(6,0)	Utilizado para la Concurrencia

**Tabla 15 Descripción de la tabla nom\_contrato**

<b>Nombre:</b> nom_contrato		
<b>Descripción:</b> Almacena los contratos.		
Atributo	Tipo	Descripción
idcontrato	numeric(2,0)	Identificador de Contrato
denom	varchar(255)	Denominación de Contrato
desde	date	Fecha de inicio de Contrato
hasta	date	Fecha de Fin de Contrato

**Tabla 16 Descripción de la tabla nom\_sistemapago**

<b>Nombre:</b> nom_sistemapago
--------------------------------

<b>Descripción:</b> Almacena los sistemas de pagos.		
Atributo	Tipo	Descripción
idsistemapago	numeric(2,0)	Identificador del sistema de pago
denom	varchar(255)	Descripción del sistema de pago
desde	date	Fecha en la que se insertó el sistema de pago
hasta	date	Fecha en la que se elimina el sistema de pago

**Tabla 17 Descripción de la tabla dat\_registropagosadicionalestrabajador**

<b>Nombre:</b> dat_registropagosadicionalestrabajador		
<b>Descripción:</b> Almacena el registro de los pagos adicionales asociados a un trabajador.		
Atributo	Tipo	Descripción
idregistropagosadicionalestrabajador	numeric(19,0)	Identificador de Registro de Pagos Adicionales asociados a un Trabajador
idpagosadicionales	numeric(19,0)	Identificador de Pagos Adicionales
idtrabajador	numeric(19,0)	Identificador del Trabajador
fechaeliminado	date	Se utiliza para no eliminar físicamente de la Base de Datos
version	numeric(6,0)	Se utiliza para la Concurrencia
importe	numeric(6,2)	Importe de Pagos Adicionales

		asociado a un Trabajador
idmodelomovnomina	numeric(19,0)	identificador del Modelo de Movimiento de Nómina
fechaefectiva	date	Fecha en que se hace efectivo el cambio efectuado en el Registro de Pagos Adicionales asociados a un Trabajador

**Tabla 18 Descripción de la tabla his\_registropagosadicionalestrabajador**

<b>Nombre:</b> his_registropagosadicionalestrabajador		
<b>Descripción:</b> Almacena el histórico de registro de pagos adicionales asociados a un Trabajador.		
Atributo	Tipo	Descripción
idhisregistropagosadicionalestrabajador	numeric(19,0)	Identificador de Histórico de Registro de Pagos Adicionales asociados a un Trabajador
idregistropagosadicionalestrabajador	numeric(19,0)	Identificador de Registro de Pagos Adicionales asociados a un Trabajador
fechaeliminado	date	Fecha en la que fue eliminado lógicamente de la Base de Datos
importe	numeric(8,2)	Importe de Pagos Adicionales
idmodelomovnomina	numeric(19,0)	Identificador de Modelo de

		Movimiento de Nómina
fechaefectiva	date	Fecha en que se hace efectivo el Pago Adicional

### 3.3 Evaluación del modelo de diseño propuesto

Son varios los puntos de vista relacionados con la calidad del software. Desde metodologías hasta las distintas normas de calidad, que pueden estar orientados tanto a los procesos de desarrollo como a los productos de software. No es objetivo de este trabajo abundar sobre los temas de calidad, pero sí desarrollar una evaluación del diseño obtenido en la solución propuesta de diseño de software al componente Trabajadores del subsistema Capital Humano integrado al sistema Cedrux.

Para la evaluación de la calidad del modelo de diseño propuesto se hizo un estudio de la creación de métricas básicas inspiradas en el estudio de la calidad del diseño orientado a objeto referenciadas por Pressman en (PRESSMAN 1998); en el mismo se abarcan atributos de calidad que permiten medir la calidad del diseño propuesto.

Atributos de calidad que se abarcan:

- Responsabilidad: Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.
- Complejidad del diseño: Consiste en la complejidad que posee una estructura de diseño de clases.
- Complejidad de implementación: Consiste en el grado de dificultad que tiene implementar un diseño de clases determinado.
- Reutilización: Consiste en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.
- Acoplamiento: Consiste en el grado de dependencia o interconexión de una clase o estructura de clase, con otras, está muy ligada a la característica de Reutilización.
- Complejidad del mantenimiento: Consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costos y la planificación del proyecto.

- Cantidad de pruebas: Consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad (Unidad) del producto (Componente, módulo, clase, conjunto de clases, etc.) diseñado.

Las métricas concebidas como instrumento para evaluar la calidad del diseño del componente Trabajador del subsistema Capital Humano integrado al sistema Cedrux y su relación con los atributos de calidad definidos son las siguientes:

- Tamaño Operacional de Clase (TOC).
- Relaciones entre Clases (RC).

### 3.3.1 Tamaño operacional de clase (TOC)

El tamaño operacional de clase (TOC), está dado por el número de métodos asignados una clase.

**Tabla 19 Tamaño operacional de clase (TOC)**

Atributo que afecta	Modo en que lo afecta
Responsabilidad	Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.
Complejidad de implementación	Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.
Reutilización	Un aumento del TOC implica una disminución en el grado de reutilización de la clase.

#### 3.3.1.1 Resultados del instrumento de evaluación de la métrica Tamaño Operacional de Clase (TOC)

El instrumento y la tabla de resultados, se muestran en el [Anexo 8](#).

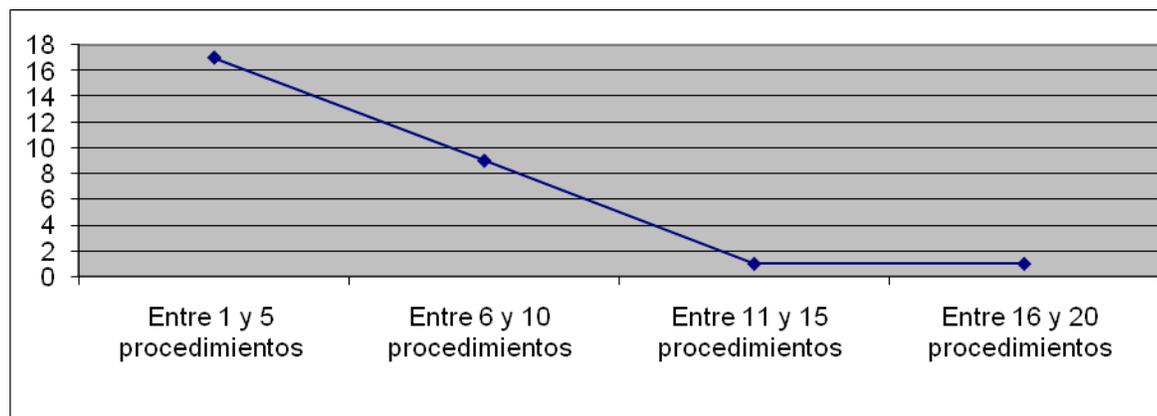


Figura 5 Representación de los resultados obtenidos en el instrumento agrupados en los intervalos definidos

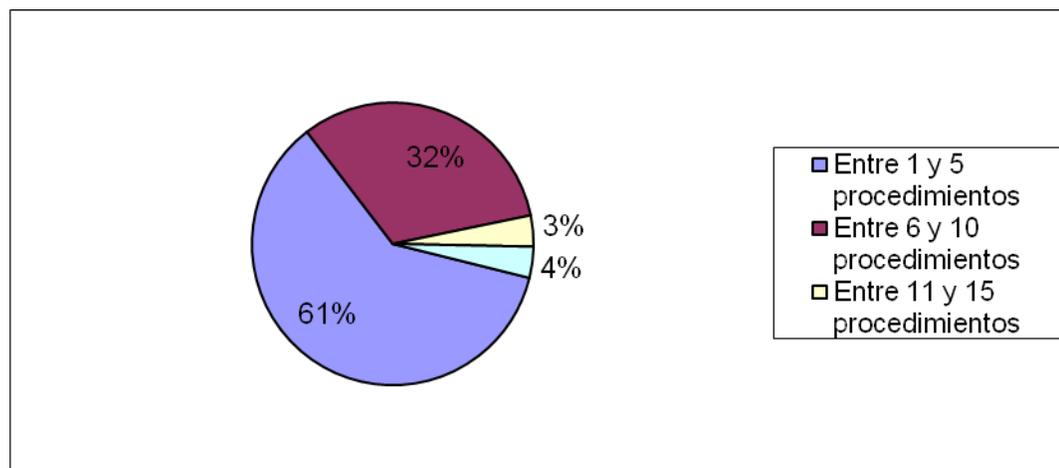
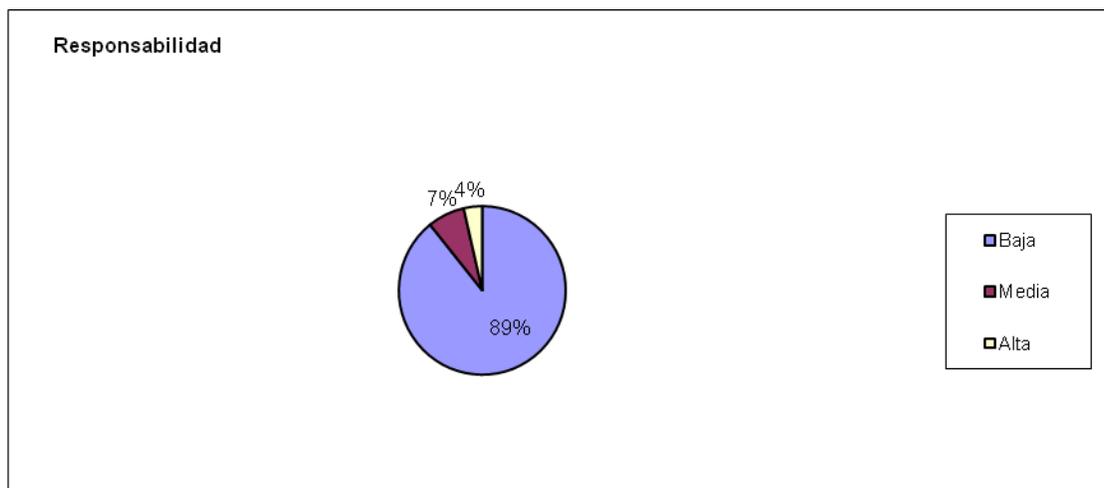
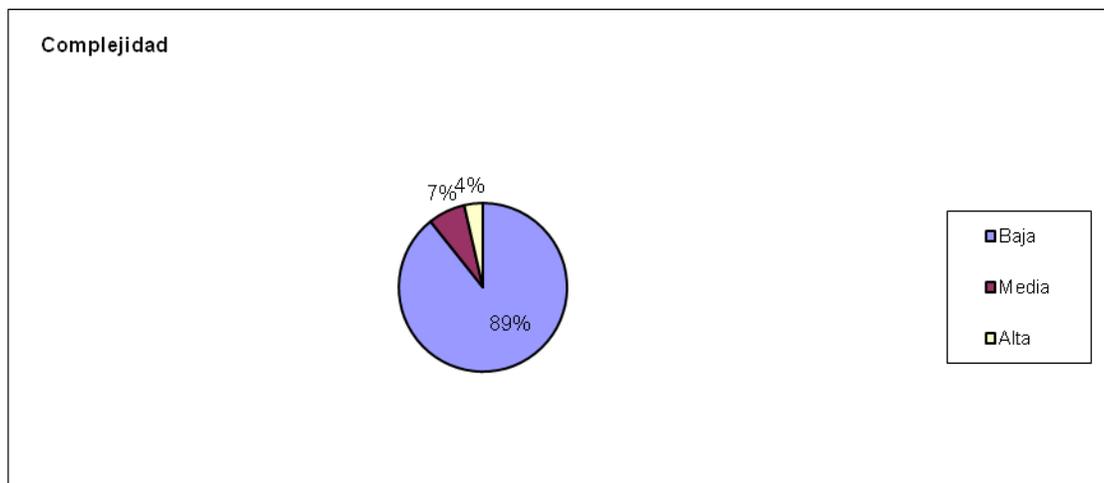


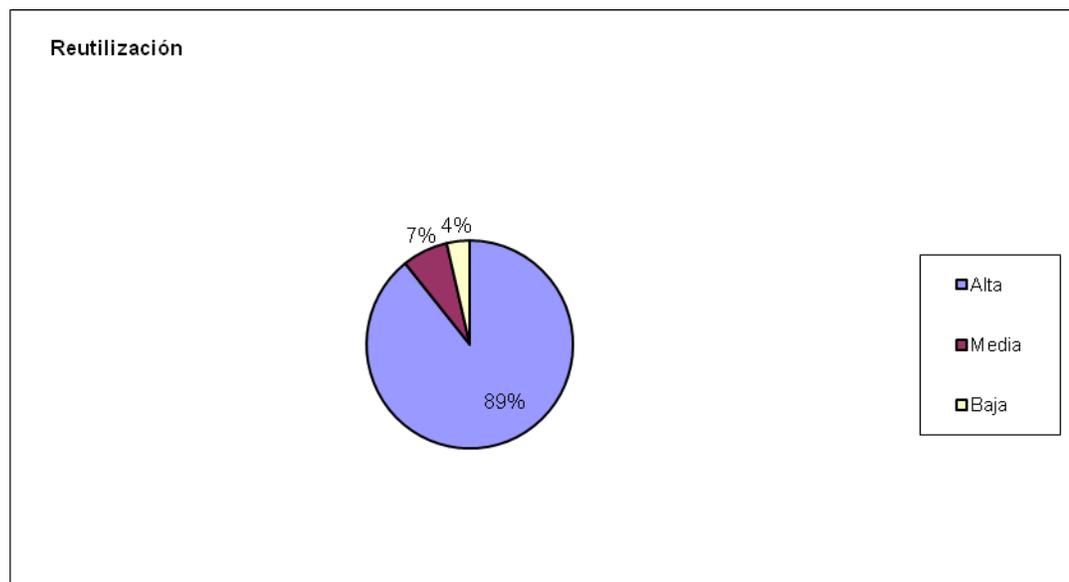
Figura 6 Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos



**Figura 7 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad**



**Figura 8 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de Implementación**



**Figura 9 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización**

Analizando los resultados obtenidos en la evaluación del instrumento de medición de la métrica TOC, teniendo en cuenta que el 93% de las clases pertenecientes al componente Trabajador cuenta con un rango de funcionalidades entre 1 y 6, y que el 89% de las clases de dicho componente poseen evaluaciones positivas en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización), podemos concluir que el diseño es aceptable.

### 3.3.2 Relaciones entre clases (RC)

Las relaciones entre las clases (RC), está dado por el número de relaciones de uso de una clase con otras.

**Tabla 20 Relaciones entre clases (RC)**

Atributo que afecta	Modo en que lo afecta
Acoplamiento	Un aumento del RC implica un aumento del Acoplamiento de la clase.
Complejidad del mantenimiento	Un aumento del RC implica un aumento de la

	complejidad del mantenimiento de la clase.
Reutilización	Un aumento del RC implica una disminución en el grado de reutilización de la clase.
Cantidad de pruebas	Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.

A continuación se presentan los resultados obtenidos a partir de la evaluación de los instrumentos aplicados a las métricas descritas anteriormente así como una valoración de los mismos.

### 3.3.2.2 Resultados del instrumento de evaluación de la métrica Relaciones entre Clases (RC)

El instrumento y la tabla de resultados, se muestran en el [Anexo 9](#).

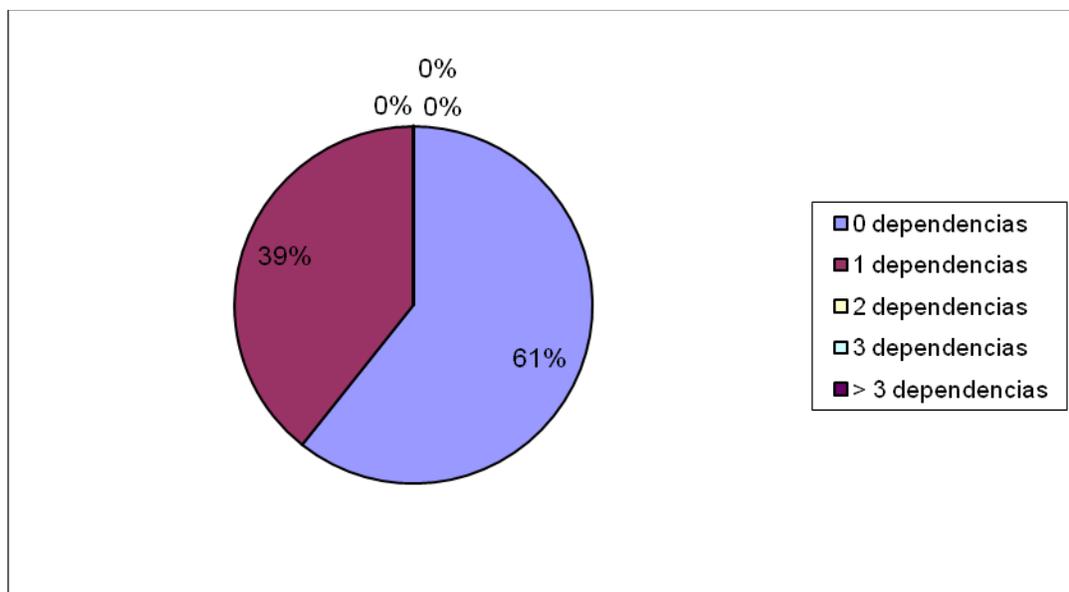


Figura 10 Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos

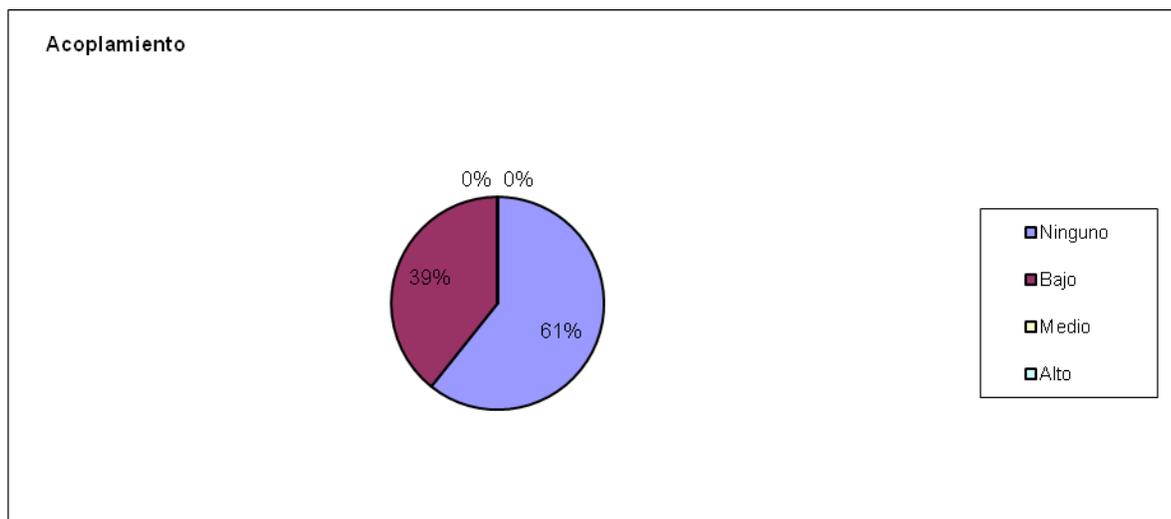


Figura 11 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento

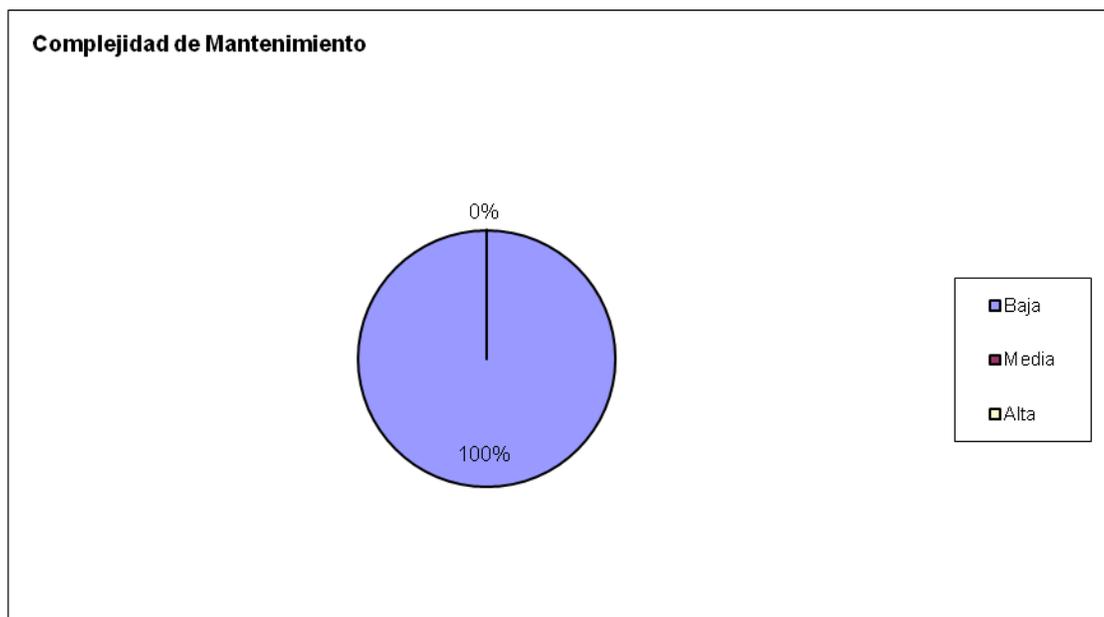
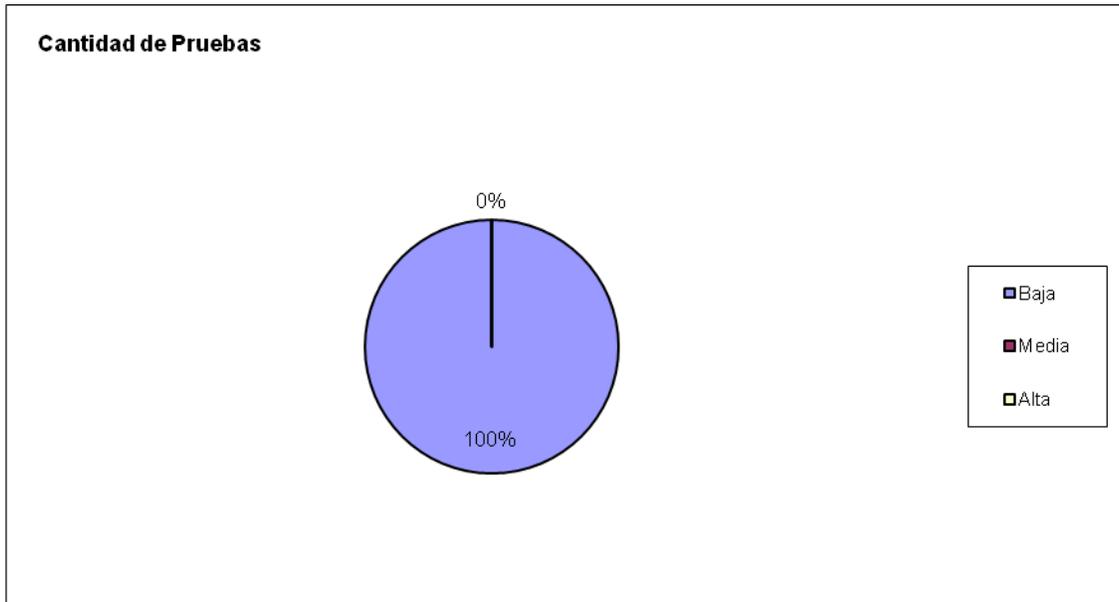
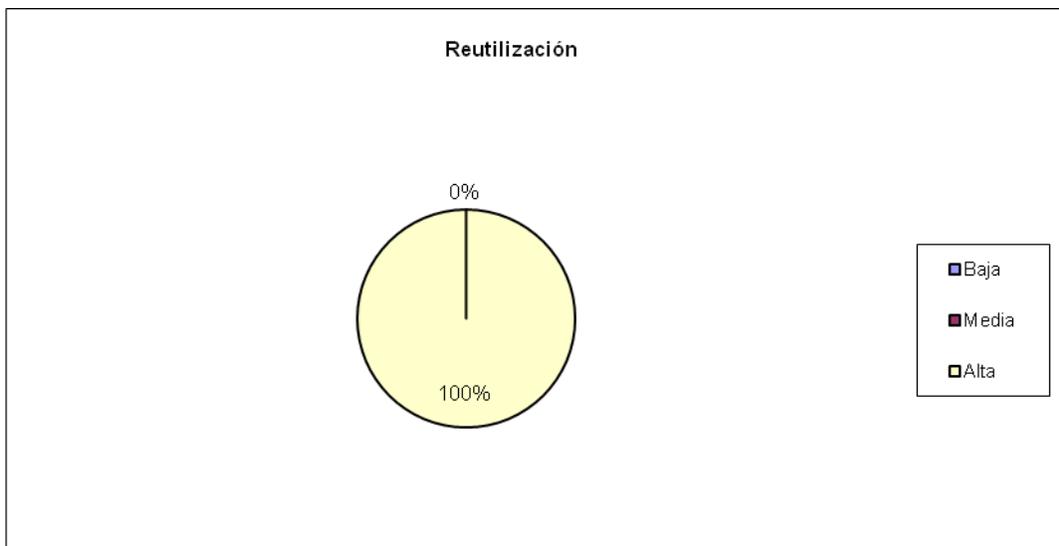


Figura 12 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de Mantenimiento



**Figura 13** Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de Pruebas



**Figura 14** Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización

Analizando los resultados obtenidos en la evaluación del instrumento de medición de la métrica RC, teniendo en cuenta que el 93% de las clases pertenecientes al componente Trabajador 61% no posee dependencias y que el 39% posee solo una dependencia con otras clases, además de que el 61% no tiene acoplamiento y que el resto poseen un bajo acoplamiento, así mismo los atributos de calidad Complejidad de Mantenimiento, Cantidad de Pruebas y Reutilización se comportan satisfactoriamente, podemos concluir que el diseño es aceptable.

### **Conclusiones**

A modo de conclusión se puede decir que en éste capítulo se abarcaron los aspectos más significativos del diseño del componente Trabajador del subsistema Capital Humano del sistema Cedrux, se exponen los patrones de diseños utilizados, también se elaboran artefactos como los diagramas de clases de diseño, que permiten desarrollar de manera más eficiente la implementación del componente. Además se evalúa el diseño propuesto mediante métricas de calidad donde se obtienen resultados aceptables.

### CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBAS

#### **Introducción**

En el presente capítulo serán abordados todos los temas referentes a la implementación y pruebas del sistema. En el mismo se muestra el modelo de implementación, donde se representan las clases de diseño en términos de componentes, se muestra además el diagrama de componentes y se especifica el modelo de pruebas donde se especifican los casos de prueba que componen el sistema. Para realizar la implementación del sistema el equipo de trabajo se baso en los estándares de codificación definidos en el marco del ERP.

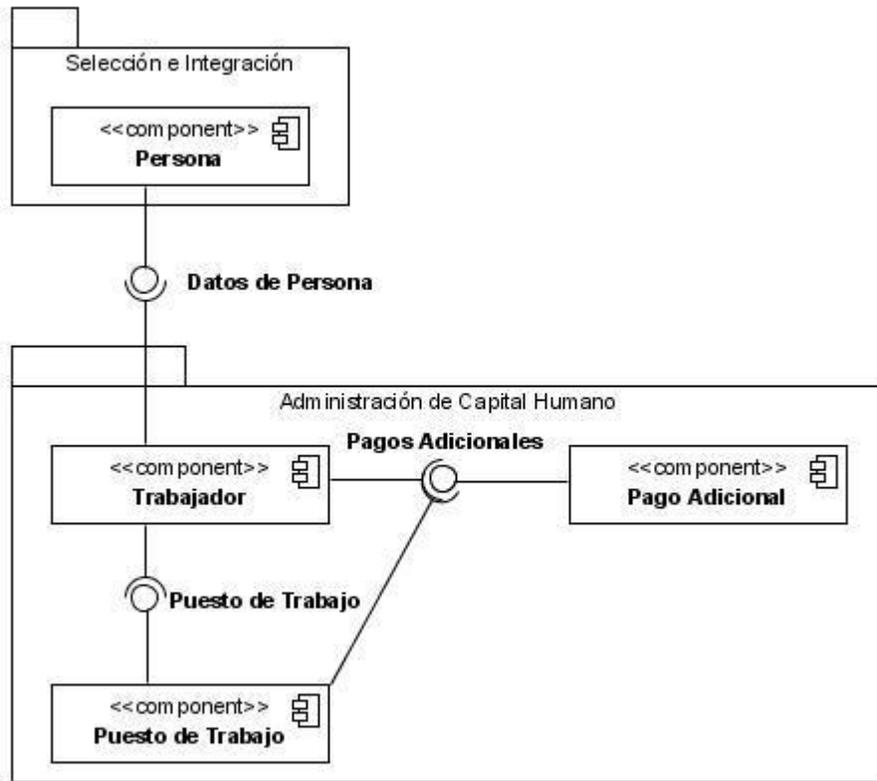
#### **4.1 Modelo de implementación**

El modelo de implementación del componente Trabajador del subsistema Capital Humano integrado al sistema Cedrux, describe como los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, ya sean ficheros de código fuente y ejecutables. Este modelo describe como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje, además en el mismo se muestra como dependen estos componentes unos de otros.

##### **4.1.1 Diagrama de componentes**

El diagrama de componentes correspondiente al componente Trabajador del subsistema Capital Humano integrado al sistema Cedrux contiene componentes, interfaces, y relaciones entre ellos, incluyendo paquetes utilizados para agrupar elementos en el modelo.

En la Figura a continuación se muestra el diagrama de componentes asociado al componente Trabajador.



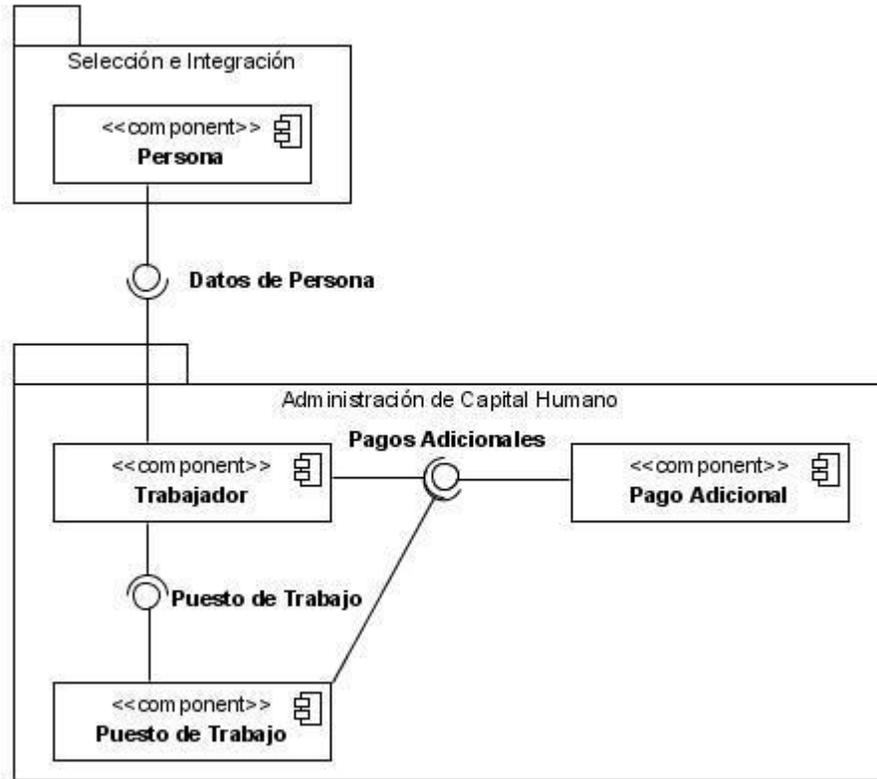


Figura 15 Diagrama de componentes para el componente Trabajador

## 4.2 Pruebas de software

La importancia de los costos que están asociados a los errores promovió a la definición y aplicación de pruebas detalladas y bien planificadas al componente Trabajador del subsistema Capital Humano integrado al sistema CedruX. Las mismas fueron un elemento fundamental para determinar la calidad de la solución propuesta, representando una revisión final de las especificaciones, del diseño y de la codificación.

Las pruebas, por su gran importancia se llevan a cabo durante todo el ciclo de vida del producto, su mayor punto de desarrollo se encuentra en la etapa de implementación.

Dentro de las pruebas se destacan principalmente dos tipos de pruebas:

- Caja Negra.

- Caja Blanca.

#### 4.2.1 Pruebas de caja negra

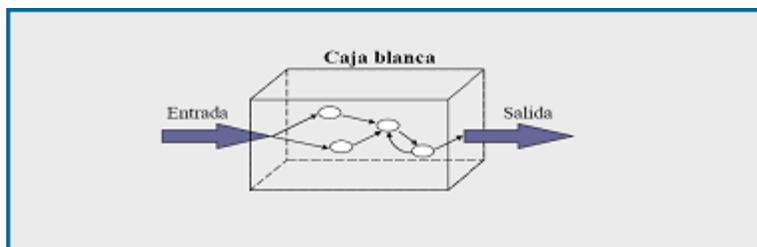
Las pruebas de **caja negra** son las que se llevan a cabo sobre la interfaz del software, o sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

Estas pruebas permiten encontrar:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación. (39)

#### 4.2.2 Pruebas de caja blanca

Las pruebas de la **caja blanca** se realizan sobre las funciones internas de un módulo en concreto, están dirigidas a las funciones internas. Entre las técnicas usadas se encuentran; la cobertura de caminos, pruebas sobre las expresiones lógico-aritméticas, pruebas de camino de datos, comprobación de bucles.



**Figura 16 Representación de pruebas de Caja Blanca**

Tipos de prueba de caja blanca:

- Prueba de Condición: Es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa.
- Prueba de Flujo de Datos: Se selecciona caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa.

- Prueba de Bucles: Es una técnica de prueba de caja blanca que se centra exclusivamente en la validez de las construcciones de bucles.
- Prueba del Camino Básico: Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico.

De las pruebas anteriormente mencionadas al sistema desarrollado se le aplicaron pruebas de caja negra, realizadas por el departamento de calidad del proyecto y las pruebas de caja blanca. Dentro de la prueba de caja blanca, la técnica que se utilizó fue la de prueba del camino básico. Esta prueba permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución, además también garantiza que durante la prueba en los casos de prueba obtenidos a través del camino básico se ejecute cada sentencia del programa por lo menos una vez.

Para aplicar la técnica del camino básico se debe introducir la notación para la representación del flujo de control, este puede representarse por un Grafo de Flujo en el cual:

- Cada nodo del grafo corresponde a una o más sentencias de código fuente.
- Todo segmento de código de cualquier programa se puede traducir a un Grafo de Flujo.
- Se calcula la complejidad ciclomática del grafo.

Para construir el grafo se debe tener en cuenta la notación para las instrucciones.

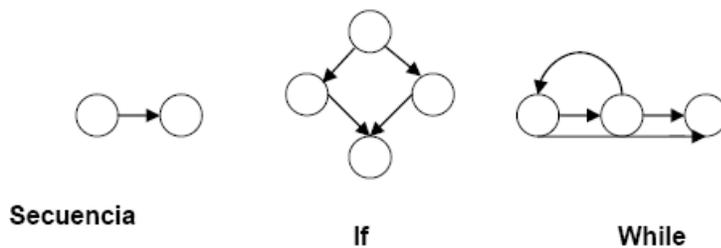
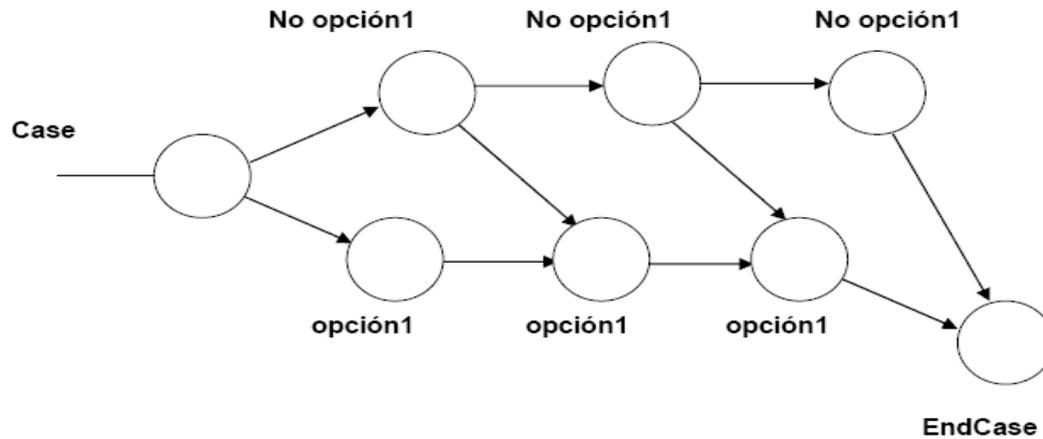


Figura 17 Notación de grafos de flujo para las instrucciones: Secuenciales, If, While



**Figura 18 Notación de grafos de flujo para la instrucción Case**

Un grafo de flujo está formado por 3 componentes fundamentales que ayudan a su elaboración y comprensión, estos brindan información para confirmar que el trabajo se está haciendo adecuadamente.

Componentes del grafo de flujo:

- **Nodo:** son los círculos representados en el grafo de flujo, el cual representa una o más secuencias del procedimiento, donde un nodo corresponde a una secuencia de procesos o a una sentencia de decisión. Los nodos que no están asociados se utilizan al inicio y final del grafo.
- **Aristas:** son constituidas por las flechas del grafo, son iguales a las representadas en un diagrama de flujo y constituyen el flujo de control del procedimiento. Las aristas terminan en un nodo, aun cuando el nodo no representa la sentencia de un procedimiento.
- **Regiones:** son las áreas delimitadas por las aristas y nodos donde se incluye el área exterior del grafo, como una región más. Las regiones se enumeran siendo la cantidad de regiones equivalente a la cantidad de caminos independientes del conjunto básico de un procedimiento.

Para realizar la técnica de prueba de caja blanca, específicamente la prueba del camino básico es necesario calcular antes la complejidad ciclomática del algoritmo o fragmento de código a analizar. A continuación se enumera las sentencias de código del procedimiento realizado sobre el método `adicionarMovalta ($arrModelomov, $arrTrabajador, $arrPagosadicionales)` el cual se encarga de emitir un movimiento de alta a una persona.

```

function adicionarMovalta($arrModelomov, $arrTrabajador, $arrPagosadicionales)
{
    if($arrTrabajador != 0)
    {
        $modelModelomovimiento = new DatModelomovnominaModel();
        $modelModelomovimiento->Insertar($arrModelomov);

        $modelTrabajador = new DatTrabajadorModel();
        $modelTrabajador->Insertar($arrTrabajador);

        $idpuestotrabajo = $arrTrabajador->idpuestotrabajo;
        $this->pIntegrator->puesto_trabajo->ModificarEstadoPuestoTrab ( $idpuestotrabajo, 1 );

        if($arrPagosadicionales != 0)
        {
            $modelRegistroPagos = new DatRegistropagosadicionalestrabajadorModel ( );
            $datRegistroPagos = null;
            for($i = 0; $i < count($arrPagosadicionales); $i++)
            {
                $datRegistroPagos = new DatRegistropagosadicionalestrabajador ( );
                $datRegistroPagos->idpagosadicionales = $arrPagosadicionales[$i]->idpagosadicionales;
                $datRegistroPagos->importe = $arrPagosadicionales[$i]->valor;
                $datRegistroPagos->idmodelomovnomina = $arrModelomov->idmodelomovnomina;
                $datRegistroPagos->idtrabajador = $arrTrabajador->idtrabajador;
                $modelRegistroPagos->Insertar ( $datRegistroPagos );
                $datRegistroPagos = null;
            }
        }
        return true;
    }
    else
    return false;
}

```

Figura 19 Representación del algoritmo adicionarMovalta(\$idpagosadicionales)

A continuación se representa el Grafo de flujo asociado al algoritmo anteriormente descrito:

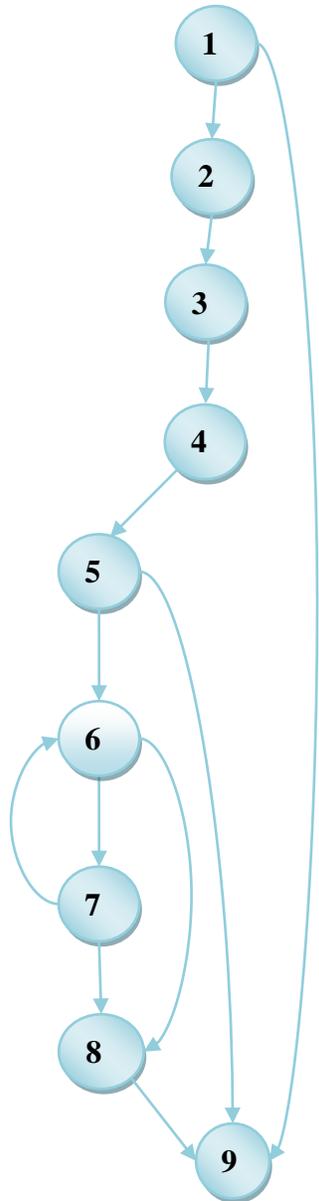


Figura 20 Grafo de flujo asociado al algoritmo `adicionarMovalta($arrModelomov, $arrTrabajador, $arrPagosadicionales)`

#### 4.2.2.1 Cálculo de la complejidad ciclomática a partir de un segmento de código

La complejidad ciclomática es una métrica de software extremadamente útil pues proporciona una medición cuantitativa de la complejidad lógica de un programa. El valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y da un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez.

Para efectuar el cálculo de la complejidad ciclomática del código es necesario tener varios parámetros como son la cantidad total de aristas del grafo, cantidad total de nodos para la siguiente fórmula:

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

$$V(G) = (11 - 9) + 2$$

$$V(G) = 4$$

Siendo "A" la cantidad total de aristas y "N" la cantidad total de nodos.

Se puede usar también:

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

$$V(G) = 3 + 1$$

$$V(G) = 4$$

Siendo "P" la cantidad total de nodos predicados (son los nodos de los cuales parten dos o más aristas).

$$V(G) = R$$

$$V(G) = 4$$

Siendo "R" la cantidad total de regiones, para cada fórmula "V(G)" representa el valor del cálculo.

El cálculo efectuado mediante las tres fórmulas ha dado el mismo valor, por lo que se puede plantear que la complejidad ciclomática del código es de 4, lo que significa que existen cuatro posibles caminos por donde el flujo puede circular, este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento tratado.

**Tabla 21 Caminos básicos del flujo**

Número	Camino básico
1	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9
2	1 – 2 – 3 – 4 – 5 - 6 – 8 – 9
3	1 – 2 – 3 – 4 – 5 – 9
4	1 – 9

Después de haber extraído los caminos básicos del flujo, se procede a ejecutar los casos de pruebas para este procedimiento, se debe realizar al menos un caso de prueba por cada camino básico.

Para realizarlos es necesario cumplir con las siguientes exigencias:

- Descripción: Se hace la entrada de datos necesaria, validando que ningún parámetro obligatorio pase nulo al procedimiento o no se entre algún dato erróneo.
- Condición de ejecución: Se especifica cada parámetro para que cumpla una condición deseada para ver el funcionamiento del procedimiento.
- Entrada: Se muestran los parámetros que entran al procedimiento.
- Resultados Esperados: Se expone resultado que se espera que devuelva el procedimiento.

**Caso de prueba para el camino básico 1**

Camino 1: [1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9]

**Descripción**

Los datos de entrada cumplirán con los siguientes requisitos:

- El parámetro arrModelomov no está vacío, contiene los datos del movimiento de nómina que se conforman cuando se llenan todos los datos correspondientes al movimiento de alta.
- El arreglo \$arrTrabajador no está vacío, contiene los datos de un trabajador.

- El arreglo \$arrPagosadicionales no está vacío, contiene los datos de un pago adicional.

Entrada:

- \$arrModelomov = array( 'idmodelomovnomina' => 45455555, 'idmotivomovnomina' => 3, 'fechamov' => 29/05/2009, 'fechaefectiva' => 29/05/2009, 'version' => 0, 'fechafin' => 29/05/2009, 'observaciones' => observaciones, 'idestructuracomun' => 100000004).
- \$arrTrabajador = array( 'idpersona' => 500000034, 'idpuestotrabajo' => 500000049, 'idcontrato' => 1, 'numexpediente' => 2344666, 'version' => 0, 'idimpuesto' => 500000, 'idmodelomovnomina' => 45455555, 'idsistemapago' => 2).
- \$arrPagosadicionales = array( 'idpagosadicionales' => 500008, 'denom' => Diez Por ciento CIES VI, 'valor' => 26.00).

Resultados esperados:

Se espera que se le realice el movimiento de alta a la persona.

### **Caso de prueba para el camino básico 2**

Camino 1: [1 – 2 – 3 – 4 – 5 - 6 – 8 – 9]

### **Descripción**

Los datos de entrada cumplirán con los siguientes requisitos:

- El parámetro arrModelomov no está vacío, contiene los datos del movimiento de nómina que se conforman cuando se llenan todos los datos correspondientes al movimiento de alta.
- El arreglo \$arrTrabajador no está vacío, contiene los datos de un trabajador.
- El arreglo \$arrPagosadicionales está vacío, no contiene los datos de ningún pago adicional.

Entrada:

- \$arrModelomov = array( 'idmodelomovnomina' => 45455555, 'idmotivomovnomina' => 3,

```
'fechamov' => 29/05/2009, 'fechaefectiva' => 29/05/2009, 'version' => 0,  
'fechafin' => 29/05/2009, 'observaciones' => observaciones,  
'idestructuracomun' => 100000004).
```

- \$arrTrabajador = array( 'idpersona' => 500000034, 'idpuestotrabajo' => 500000049,  
    'idcontrato' => 1, 'numexpediente' => 2344666, 'version' => 0,  
    'idimpuesto' => 500000, 'idmodelomovnomina' => 45455555,  
    'idsistemapago' => 2).
- \$arrPagosadicionales = array().

Resultados esperados:

Se espera que se le realice el movimiento de alta a la persona, a pesar de que no se inserta el trabajador con pagos adicionales, estos pueden o no asignar al trabajador.

### **Caso de prueba para el camino básico 3:**

Camino 1: [1 – 2 – 3 – 4 – 5 – 9]

#### **Descripción:**

Los datos de entrada cumplirán con los siguientes requisitos:

- El parámetro arrModelomov no está vacío, contiene los datos del movimiento de nómina que se conforman cuando se llenan todos los datos correspondientes al movimiento de alta.
- El arreglo \$arrTrabajador no está vacío, contiene los datos de un trabajador.
- El arreglo \$arrPagosadicionales no está vacío, contiene los datos de un pago adicional.

Entrada:

- \$arrModelomov = array( 'idmodelomovnomina' => 45455555, 'idmotivomovnomina' => 3,  
    'fechamov' => 29/05/2009, 'fechaefectiva' => 29/05/2009, 'version' => 0,  
    'fechafin' => 29/05/2009, 'observaciones' => observaciones,  
    'idestructuracomun' => 100000004)

- \$arrTrabajador = array( 'idpersona' => 500000034, 'idpuestotrabajo' => 500000049, 'idcontrato' => 1, 'numexpediente' => 2344666, 'version' => 0, 'idimpuesto' => 500000, 'idmodelomovnomina' => 45455555, 'idsistemapago' => 2)
- \$arrPagosadicionales = array()

Resultados esperados:

Se espera que se le realice el movimiento de alta a la persona, a pesar de que no se inserta el trabajador con pagos adicionales, estos pueden o no asignar al trabajador.

### **Caso de prueba para el camino básico 4:**

Camino 1: [1 – 9]

#### **Descripción:**

Los datos de entrada cumplirán con los siguientes requisitos:

- El parámetro arrModelomov está vacío, no contiene los datos del movimiento de nómina.
- El arreglo \$arrTrabajador está vacío, no contiene los datos de un trabajador.
- El arreglo \$arrPagosadicionales no está vacío, contiene los datos de un pago adicional.

Entrada:

- \$arrModelomov = array().
- \$arrTrabajador = array().
- \$arrPagosadicionales = array( 'idpagosadicionales' => 500008, 'denom' => Diez Por ciento CIES VI, 'valor' => 26.00).

Resultados esperados:

Se espera que no se le realice el movimiento de alta a la persona, ya que no se le incorporaron los datos al trabajador.

Luego de haberse aplicado pruebas de caja blanca, en específico las pruebas del camino básico, seleccionando diferentes caminos a través del Cálculo de la complejidad ciclomática a partir de un segmento de código, se ha arribado a la conclusión de que los resultados obtenidos fueron aceptables ya que se pudo comprobar que el flujo de trabajo de la función está correcto ya que cumple con las condiciones necesarias que se habían planteado.

### **Conclusiones**

En el capítulo se ha realizado la descripción de la implementación, se muestra el diagrama de componentes el cual contiene paquetes que representan las integraciones que existen dentro del sistema desarrollado. Además se le realizan pruebas de caja blanca, en específico la prueba del camino en las cuales se obtienen resultados favorables, demostrándose con ello la calidad y eficiencia del componente Trabajador del subsistema Capital Humano del sistema CedruX.

### CONCLUSIONES

La realización de este trabajo, propició un estudio referente a la gestión de los trabajadores como parte de los procesos de gestión de Capital Humano. Se estableció además una investigación respecto a varios sistemas que gestionan estos procesos analizándose sus características funcionales y bases tecnológicas. Además se desarrolló la modelación del proceso de negocio permitiendo una descripción de cada uno de sus procesos. Se hizo una descripción de los requisitos funcionales definidos. Quedó representado el modelo de datos que muestra la interrelación entre las tablas de la base de datos. Se estudió la documentación referente al marco de trabajo así como la documentación de las herramientas a utilizar para el desarrollo del componente. Para comprobar la calidad y eficiencia del componente se realizaron las pruebas de caja blanca al mismo.

De esta manera se ha logrado dar cumplimiento a los objetivos planteados los cuales mediante la realización de las diferentes tareas de investigación encaminaron el proceso de desarrollo del componente Trabajador del subsistema Capital Humano del sistema Cedrux.

### RECOMENDACIONES

Luego de haber cumplimentado los objetivos propuestos mediante la realización del trabajo, obteniendo así como resultado una solución que permita gestionar los procesos de los trabajadores como parte de los procesos de Capital Humano, se recomienda:

- Proseguir con la investigación e identificar nuevos requisitos para actualizar el sistema desarrollado, garantizando mayor funcionabilidad y aceptación del cliente.
- Introducir mejoras en las vías implementadas para eliminación de los trabajadores.
- Optimizar las funcionalidades de búsqueda de los trabajadores, con nuevos algoritmos más eficientes que los utilizados.
- Implantar el sistema en otras entidades.

REFERENCIA BIBLIOGRÁFICA

7. Bustamante Molina, Vivian. Empleará aeronáutica civil cubana sistema integral de gestión Kewan. [En línea] 2009.  
<http://edicionesanteriores.trabajadores.cu/2006/enero/30/cuba/empleara.htm>.
16. BPMN. [En línea] <http://pana10.files.wordpress.com/2007/12/bpmn1.ppt>.
17. Geek OPEN TI. [En línea] <http://geeknet.com.mx/principal.php?op=detalle1>.
18. Guía Breve de Tecnologías XML. [En línea]  
<http://www.w3c.es/divulgacion/guiasbreves/tecnologiasXML>.
19. JSON, acrónimo de “JavaScript Object Notation”. [En línea] 2009.  
<http://felipengineer.wordpress.com/2009/01/26/json-acronimo-de-javascript-object-notation/>.
20. Glosario. [En línea] <http://www.pixelmio.net/glosario.php>.
21. Guía Breve de CSS. [En línea] <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>.
22. Manuales, Tutoriales y Herramientas. [En línea] <http://max-alva.webs.com/javascript.htm>.
23. New Features. [En línea] 2007. <http://extjs.com/blog/2007/09/28/ext-20-alpha-release/>.
26. El proyecto IDEE. [En línea] <http://www.ineter.gob.ni/sig/docs/Intro%20IDE%20-%20IDEE.pdf>.
27. El ataque de los frameworks. [En línea] 2008. <http://www.webtaller.com/maletin/articulos/el-ataque-de-los-frameworks.php>.
28. ¿Qué es un framework? [En línea] <http://jordisan.net/blog/2006/que-es-un-framework/>.
29. Markiewicz, Marcus Eduardo y de Lucena, Carlos J.P. El Desarrollo del Framework Orientado al . [En línea] <http://www.acm.org/crossroads/espanol/xrds7-4/frameworks.html>.
30. Zend Framework, una introducción. [En línea] 2007.  
<http://www.carlosleopoldo.com/post/zend-framework-una-introduccion/>.
31. What is Doctrine? [En línea] <http://www.doctrine-project.org/>.

33. Guía del Programador de PostgreSQL. [En línea]  
<https://forja.rediris.es/docman/view.php/312/461/Postgres-Programmer.pdf>.
34. EMS SQL Manager para PostgreSQL. [En línea]  
<http://sqlmanager.net/en/products/postgresql/manager>.
36. Collins Sussman, Fitzpatrick, B. y Pilato, W.B. Control de versiones con Subversion. [En línea]  
2008. <http://svnbook.red-bean.com/nightly/es/svn-book.pdf>.
37. Gracia, Joaquín. Patrones de diseño. [En línea] <http://www.ingenierosoftware.com>.
38. Larman, Craig. UML y Patrones. [En línea] <http://bibliodoc.uci.cu>.

### BIBLIOGRAFÍA

1. Lazcano Herrera, MSc. Carlos. Capital Humano, en busca de un contexto. [En línea] [http://www.elprisma.com/apuntes/administracion\\_de\\_empresas/capitalhumano](http://www.elprisma.com/apuntes/administracion_de_empresas/capitalhumano).
2. Norma Cubana 3000 - SISTEMA DE GESTIÓN INTEGRADA DE CAPITAL HUMANO— VOCABULARIO. 2007.
3. Departamento central de ingeniería. [En línea] 2007. <http://www.teleformacion.uci.cu/mod/resource/view.php?id=11402..>
4. Sommerville, Ian. Ingeniería de software. [En línea] 2005.
5. Ariza Rojas, Maribel y Molina García, Juan Carlos. INTRODUCCIÓN Y PRINCIPIOS BÁSICOS DEL DESARROLLO DE SOFTWARE BASADO EN COMPONENTES. [En línea] 2004. <http://pegasus.javeriana.edu.co/~jcpymes/Docs/DSBC.pdf>.
6. ASSETS, Sistema de Gestión Integral. [En línea] <http://assets.co.cu/assets.asp>.
8. Sage MAS 500, Elevar de rendimiento, productividad y rentabilidad. [En línea] <http://www.blytheco.com/mas500/>.
9. [En línea] <http://www.forum.villaclara.cu/UserFiles/forum/PonenciasWORD/0500691.doc>.
10. Un efectivo sistema cubano de contabilidad. [En línea] <http://www.eleconomista.cubaweb.cu/2005/edicionimpresa/premioeconomiamiguel.html>.
11. Camargo, Yolanda y Natividad de Mateo. Exact Software. [En línea] <http://www.exactsoftware.es/index.igw>.
12. Gestión de Recursos Humanos con Exact Globe. [En línea] <http://www.gadisistemas.com/index.php?modulo=20260>.
13. RODAS XXI, Sistema Integral Economico Administrativo. [En línea] <http://www.rodasxxi.cu/rodasxxi.php>.
14. Visual paradigm. [En línea] <http://www.visual-paradigm.com>.

15. UML. [En línea] <http://mayi.polanco.googlepages.com/TRABAJODEINGSOFTWAREII.doc>.
24. Introducción, definición y evolución de PHP. [En línea] [http://php.ciberaula.com/articulo/introduccion\\_php/](http://php.ciberaula.com/articulo/introduccion_php/).
25. Rodas Hinostraza, Raul. Características de PHP. [En línea] 2007. <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
32. WampServer. Instalar PHP5. [En línea] <http://www.desarrolloweb.com/articulos/wampserver-instalar-php5.html>.
35. ¿Qué es Mozilla Firefox? [En línea] <http://firefoxperu.blogspot.com/2005/12/qu-es-mozilla-firefox.html>.
39. Pressman, Roger; Ingeniería de software. Un enfoque práctico. 2002. McGraw.Hill/Interamericana de España.
40. Introducción a Apache.[En línea] 2008. [http://linux.ciberaula.com/articulo/linux\\_apache\\_intro/](http://linux.ciberaula.com/articulo/linux_apache_intro/).
41. El Trabajador. [En línea]. [http://www.uned.es/deahe/alumnos/nuevo2000/apuntes/ade42204\\_derecho-leccion10.doc](http://www.uned.es/deahe/alumnos/nuevo2000/apuntes/ade42204_derecho-leccion10.doc)
42. Concepto De Trabajador, Patrón, Intermediario, Empresa Y Establecimiento. [En línea] <http://www.mitecnologico.com/Main/ConceptoDeTrabajadorPatronIntermediarioEmpresaYEstablecimiento>.

## ANEXOS

## Anexo 1



The screenshot shows a software window titled "Emitir movimiento de nómina". Below the title bar, there is a section titled "Movimiento de la fuerza de Trabajo" containing five buttons: "Buscar", "Alta", "Reubicacion", "Baja", and "Datos contables". Below this is a section titled "Lista de trabajadores" which contains a table with the following columns: "consecutivo", "Exp interno", "No. de carnet", "Nombre(s)", "Primer apellido", "Segundo apellido", "Area", and "Cargo". The table is currently empty.

consecutivo	Exp interno	No. de carnet	Nombre(s)	Primer apellido	Segundo apellido	Area	Cargo

Figura 21 Representa el prototipo de interfaz de usuario Emitir movimiento de nómina

Anexo 2

**Emitir Movimiento de Alta**

**Datos de la persona**

**Buscar**

Nombre(s):       Primer apellido       Segundo apellido       No carnet

**Datos generales**      **Adiciones**

Observaciones(del motivo)

**Cancelar**      **Aplicar**      **Aceptar**

Consecutivo:       Fecha efectiva:       Tipo contrato:       Fecha fin:

Motivo:  ▼      Expediente interno:       Expediente laboral:       Contribucion social:  ▼

Sistema de pago:  ▼      Area:  ▼      Grupo de puesto:  ▼      Puesto:  ▼

**Informacion del trabajador**

Cargo       Categoria       Grupo escala       Salario Escala

**Adicionar**      **Eliminar**

**Adiciones al trabajador**      **Adiciones al puesto**

Descripcion	Importe	Descripcion	Importe

Figura 22 Representa el prototipo de interfaz de usuario Movimiento de altas

Anexo 3

**Emitir Movimiento de Reubicación**

**Datos del trabajador**

Buscar

Nombre(s):  Primer apellido:  Segundo apellido:  Expediente interno:

**Datos actuales**      **Adiciones**

Observaciones(del motivo)

Cancelar    Aplicar    Aceptar

---

**Emitir Movimiento de Reubicación**

Fecha efectiva:  Reubicación:  Fecha fin:  Contribución especial:

Tipo de contrato:  fecha fin:  Motivo:  Consecutivo:

Sistema de pago:

Área:  Grupo de puesto:  Puesto de trabajo:  Cargo:

Categoría ocupacional:  Grupo escala:  Salario Escala:  Salario total:

---

**Emitir Movimiento de Reubicación**

Adicionar    Eliminar

**Adiciones al trabajador**      **Adiciones al puesto de trabajo**

Denominación	Valor	Denominación	Valor

Figura 23 Representa el prototipo de interfaz de usuario Movimiento de reubicación

Anexo 4

**Emitir Movimiento de Baja**

**Datos del trabajador**

Nombre(s):     Primer apellido     Segundo apellido     Expediente interno

**Datos movimiento**    **Datos actuales**

---

**Emitir Movimiento de Baja**

Puesto de trabajo:     Grupo escala:     Salario Escala:     Categoría ocupacional:

**Adiciones asociadas al trabajador**    **Adiciones asociadas al puesto de trabajo**

Denominación	Valor	Denominación	Valor

---

**Emitir Movimiento de Baja**

Fecha efectiva:     Motivo:

Observaciones:

Figura 24 Representa el prototipo de interfaz de usuario Movimiento de bajas

Anexo 5

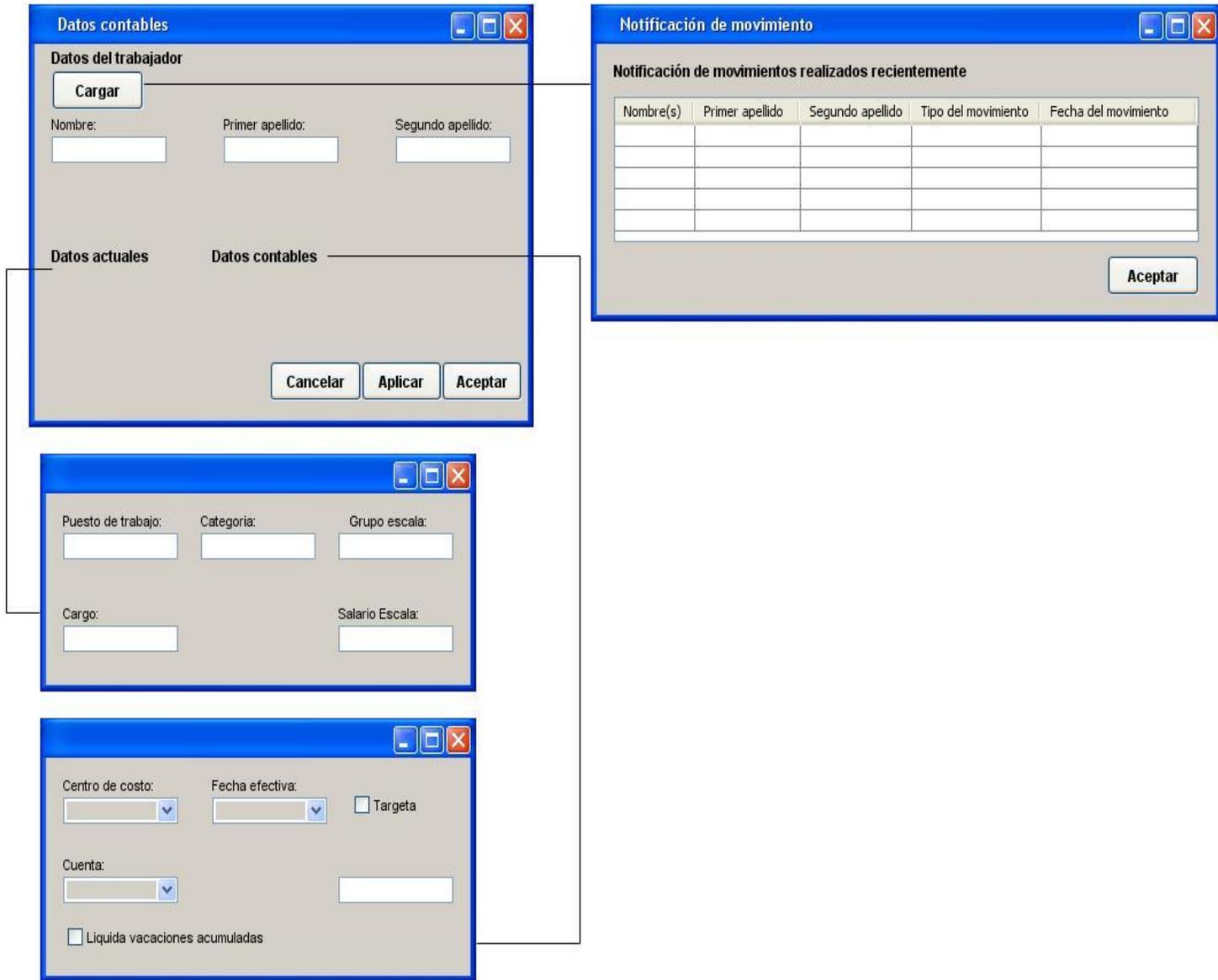


Figura 25 Representa el prototipo de interfaz de usuario Actualizar datos contables

Anexo 6

The image shows two windows from a software application. The top window, titled "Buscar trabajador", has a search form with fields for "No carnet:", "Nombre:", "Primer Apellido:", and "Segundo apellido:". It includes "Buscar" and "Avanzado" buttons. Below the form is a table with columns: "Expediente laboral", "Nombre", "Primer apellido", and "Segundo Apellido". The table is currently empty. At the bottom of the window are "Cancelar" and "Aceptar" buttons. The bottom window, titled "Búsqueda avanzada", contains more search criteria: "Expediente laboral:", "Expediente interno:", "Targeta:", "Nombre:", "Primer apellido:", "Segundo apellido", "Area" (dropdown), "Cargo" (dropdown), "Centro de costo" (dropdown), "Tipo de contrato" (dropdown), "Estado" (checkboxes for "Activo" and "Inactivo"), and "Vinculacion a la produccion" (checkboxes for "Directa" and "Indirecta"). It also has "Cancelar" and "Aceptar" buttons. A line connects the "Avanzado" button in the top window to the "Búsqueda avanzada" window.

Figura 26 Representa el prototipo de interfaz de usuario Buscar trabajador

Anexo 7 - Diagramas de clases del diseño asociados al componente Trabajador.

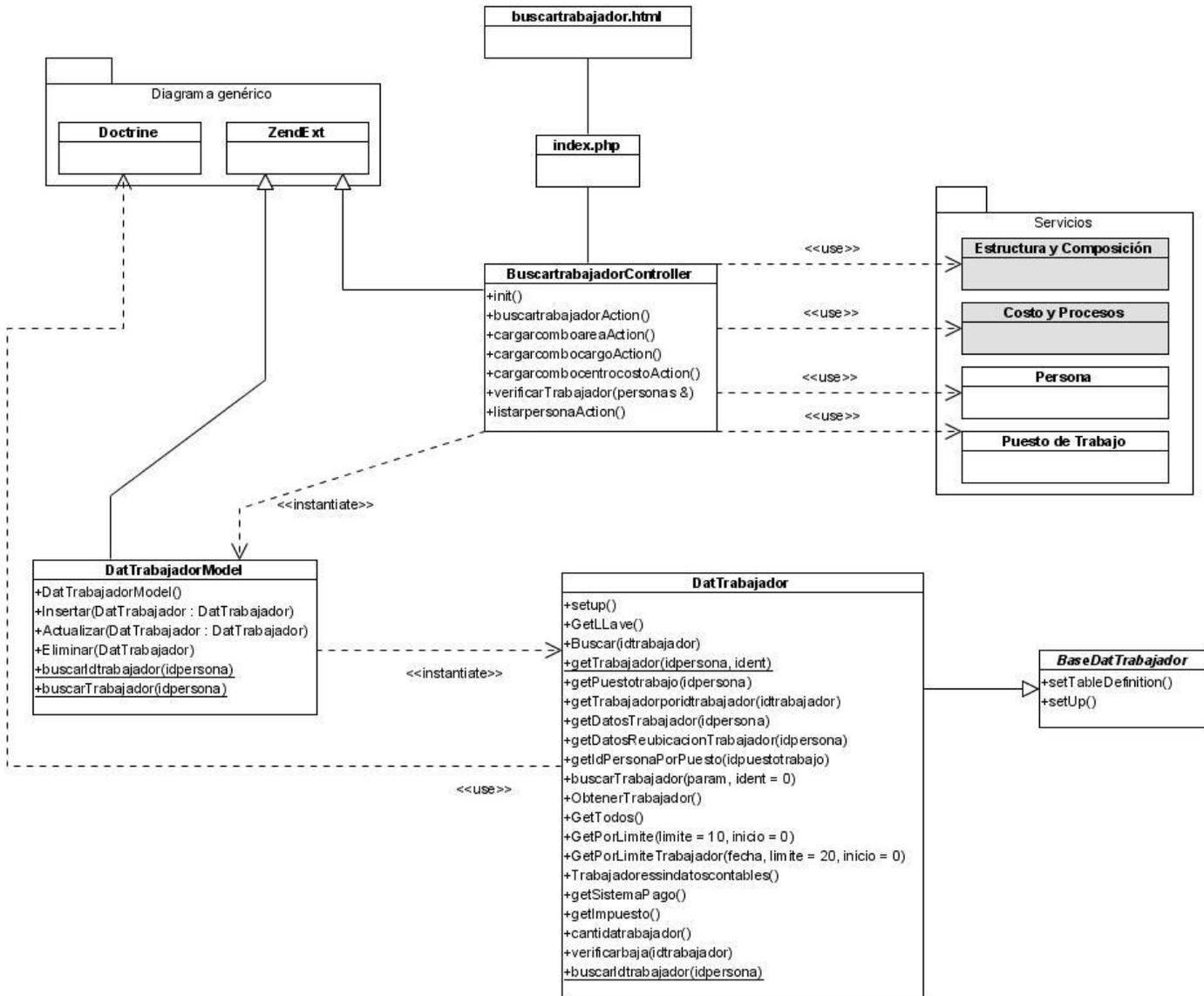


Figura 27 Diagrama de clases de diseño asociado al requisito Buscar trabajador



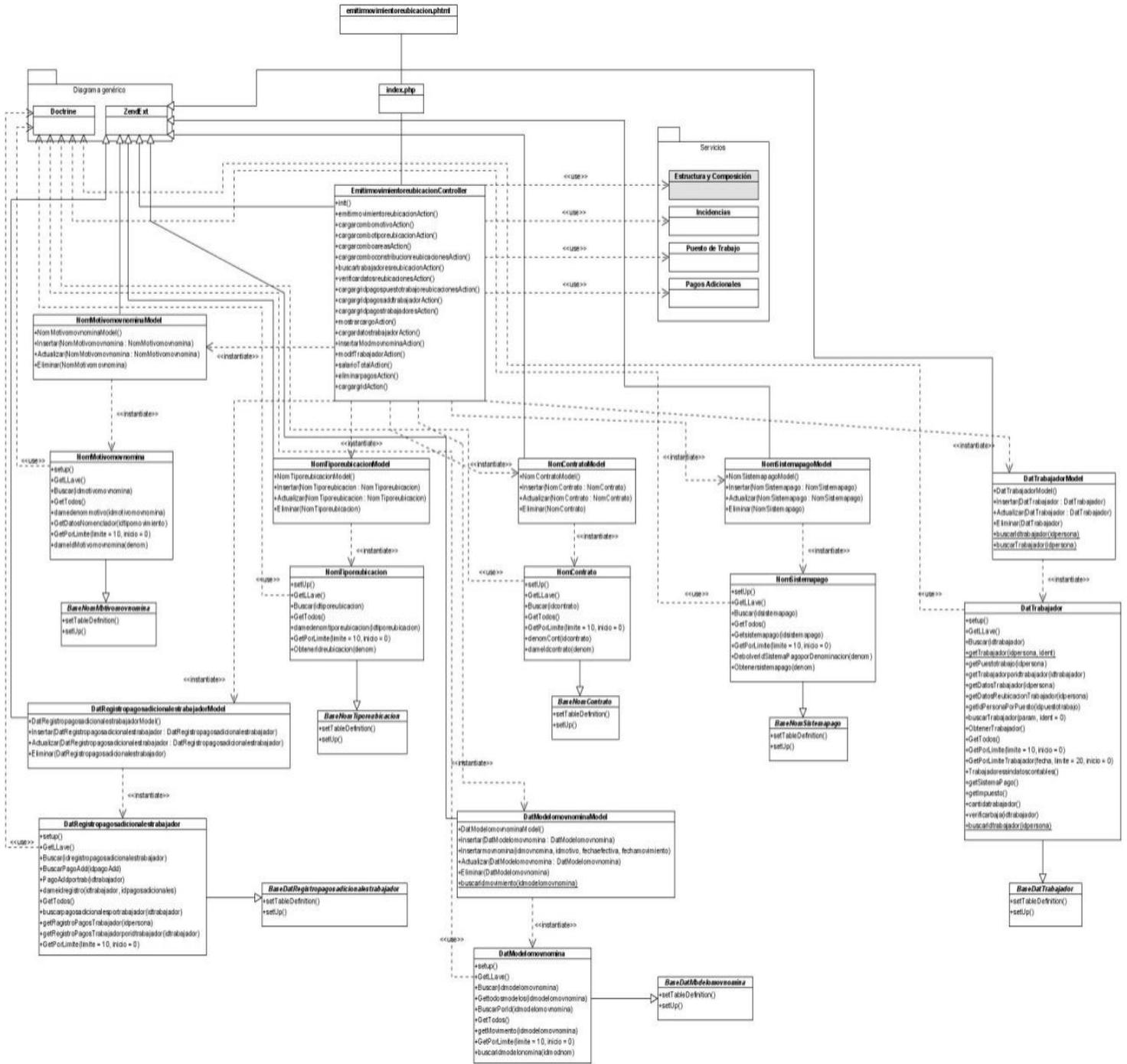


Diagrama de clases de diseño asociado al requisito Movimiento de reubicación.





Anexo 8 - Instrumento de medición de la métrica Tamaño operacional de clase (TOC).

**Tabla 22 Rango de valores de para la evaluación técnica de los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización) relacionados con la métrica TOC.**

	Categoría	Criterio
Responsabilidad	Baja	< =Prom.
	Media	Entre Prom. Y 2* Pom.
	Alta	> 2* Prom.
Complejidad implementación	Baja	< =Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	> 2* Prom.
Reutilización	Baja	> 2*Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	<= Prom.

**Tabla 23 6 Resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización).**

No	Componente	Clase	Cantidad de Procedimientos	Responsabilidad	Complejidad	Reutilización
----	------------	-------	----------------------------	-----------------	-------------	---------------

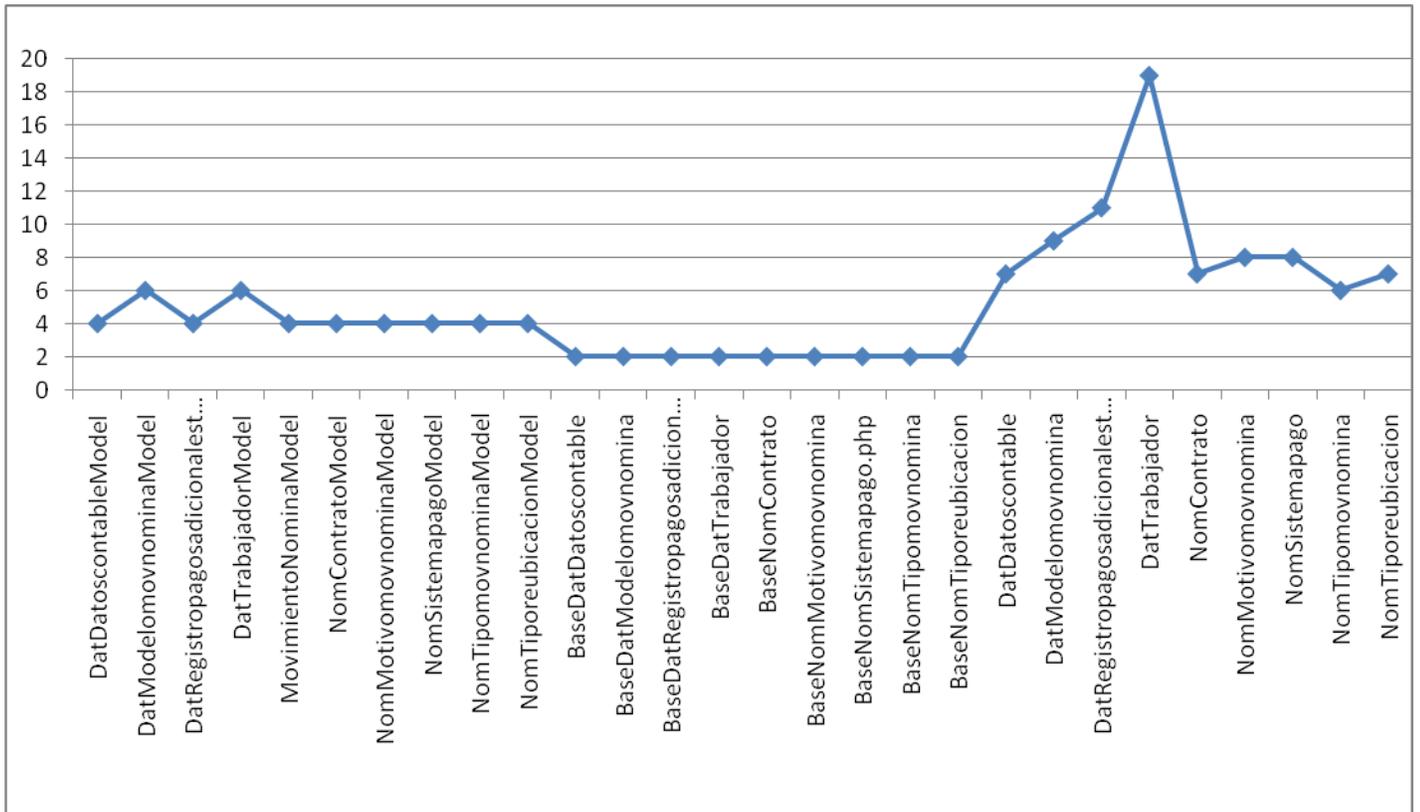
ANEXOS

1	Trabajador	DatDatoscontableModel	4	Baja	Baja	Alta
2	Trabajador	DatModelomovnominaModel	6	Baja	Baja	Alta
3	Trabajador	DatRegistropagosadicionalestrabajadorModel	4	Baja	Baja	Alta
4	Trabajador	DatTrabajadorModel	6	Baja	Baja	Alta
5	Trabajador	MovimientoNominaModel	4	Baja	Baja	Alta
6	Trabajador	NomContratoModel	4	Baja	Baja	Alta
7	Trabajador	NomMotivomovnominaModel	4	Baja	Baja	Alta
8	Trabajador	NomSistemapagoModel	4	Baja	Baja	Alta
9	Trabajador	NomTipomovnominaModel	4	Baja	Baja	Alta
10	Trabajador	NomTiporeubicacionModel	4	Baja	Baja	Alta
11	Trabajador	BaseDatDatoscontable	2	Baja	Baja	Alta
12	Trabajador	BaseDatModelomovnomina	2	Baja	Baja	Alta
13	Trabajador	BaseDatRegistropagosadicionalestrabajador	2	Baja	Baja	Alta
14	Trabajador	BaseDatTrabajador	2	Baja	Baja	Alta
15	Trabajador	BaseNomContrato	2	Baja	Baja	Alta
16	Trabajador	BaseNomMotivomovnomina	2	Baja	Baja	Alta

ANEXOS

---

17	Trabajador	BaseNomSistemapago.php	2	Baja	Baja	Alta
18	Trabajador	BaseNomTipomovnomina	2	Baja	Baja	Alta
19	Trabajador	BaseNomTiporeubicacion	2	Baja	Baja	Alta
20	Trabajador	DatDatoscontable	7	Baja	Baja	Alta
21	Trabajador	DatModelomovnomina	9	Media	Media	Media
22	Trabajador	DatRegistropagosadicional strabajador	11	Media	Media	Media
23	Trabajador	DatTrabajador	19	Alta	Alta	Baja
24	Trabajador	NomContrato	7	Baja	Baja	Alta
25	Trabajador	NomMotivomovnomina	8	Baja	Baja	Alta
26	Trabajador	NomSistemapago	8	Baja	Baja	Alta
27	Trabajador	NomTipomovnomina	6	Baja	Baja	Alta
28	Trabajador	NomTiporeubicacion	7	Baja	Baja	Alta



**Figura 31 Gráfica de los resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización)**

Anexo 9 - Instrumento de medición de la métrica Relaciones entre clases (RC).

**Tabla 24 Rango de valores de para la evaluación técnica de los atributos de calidad (Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas) relacionados con la métrica RC.**

	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	>2

ANEXOS

	Categoría	Criterio
Complejidad Mantenimiento	Baja	$\leq$ Prom.
	Media	Entre Prom. y 2*Prom.
	Alta	$> 2*Prom.$
	Categoría	Criterio
Reutilización	Baja	$>2* Prom.$
	Media	Entre Prom. y 2*Prom.
	Alta	$\leq$ Prom.
	Categoría	Criterio
Cantidad de Pruebas	Baja	$\leq$ Prom.
	Media	Entre Prom. y 2*Prom.
	Alta	$> 2*Prom.$

**Tabla 25 Resultados de la evaluación de la métrica RC y su influencia en los atributos de calidad (Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas).**

No	Componente	Clase	Cantidad de Relaciones	Acoplamiento	Complejidad	Reutilización	Cantidad de Pruebas
----	------------	-------	------------------------	--------------	-------------	---------------	---------------------

ANEXOS

			de Uso		Mantenimiento	n	
1	Trabajador	DatDatoscontableModel	0	Ninguno	Baja	Alta	Baja
2	Trabajador	DatModelomovnominaModel	1	Bajo	Baja	Alta	Baja
3	Trabajador	DatRegistropagosadicionalestrabajadorModel	0	Ninguno	Baja	Alta	Baja
4	Trabajador	DatTrabajadorModel	1	Bajo	Baja	Alta	Baja
5	Trabajador	MovimientoNominaModel	0	Ninguno	Baja	Alta	Baja
6	Trabajador	NomContratoModel	0	Ninguno	Baja	Alta	Baja
7	Trabajador	NomMotivomovnominaModel	0	Ninguno	Baja	Alta	Baja
8	Trabajador	NomSistemapagoModel	0	Ninguno	Baja	Alta	Baja
9	Trabajador	NomTipomovnominaModel	0	Ninguno	Baja	Alta	Baja
10	Trabajador	NomTiporeubicacionModel	0	Ninguno	Baja	Alta	Baja
11	Trabajador	BaseDatDatoscontable	0	Ninguno	Baja	Alta	Baja

ANEXOS

12	Trabajador	BaseDatModelomov nomina	0	Ninguno	Baja	Alta	Baja
13	Trabajador	BaseDatRegistropag osadicionalestrabaj ador	0	Ninguno	Baja	Alta	Baja
14	Trabajador	BaseDatTrabajador	0	Ninguno	Baja	Alta	Baja
15	Trabajador	BaseNomContrato	0	Ninguno	Baja	Alta	Baja
16	Trabajador	BaseNomMotivomov nomina	0	Ninguno	Baja	Alta	Baja
17	Trabajador	BaseNomSistemapa go.php	0	Ninguno	Baja	Alta	Baja
18	Trabajador	BaseNomTipomovno mina	0	Ninguno	Baja	Alta	Baja
19	Trabajador	BaseNomTiporeubic acion	0	Ninguno	Baja	Alta	Baja
20	Trabajador	DatDatoscontable	1	Bajo	Baja	Alta	Baja
21	Trabajador	DatModelomovnomi na	1	Bajo	Baja	Alta	Baja
22	Trabajador	DatRegistropagosadi cionalestrabajador	1	Bajo	Baja	Alta	Baja
23	Trabajador	DatTrabajador	1	Bajo	Baja	Alta	Baja
24	Trabajador	NomContrato	1	Bajo	Baja	Alta	Baja

ANEXOS

25	Trabajador	NomMotivomovnomina	1	Bajo	Baja	Alta	Baja
26	Trabajador	NomSistemapago	1	Bajo	Baja	Alta	Baja
27	Trabajador	NomTipomovnomina	1	Bajo	Baja	Alta	Baja
28	Trabajador	NomTiporeubicacion	1	Bajo	Baja	Alta	Baja

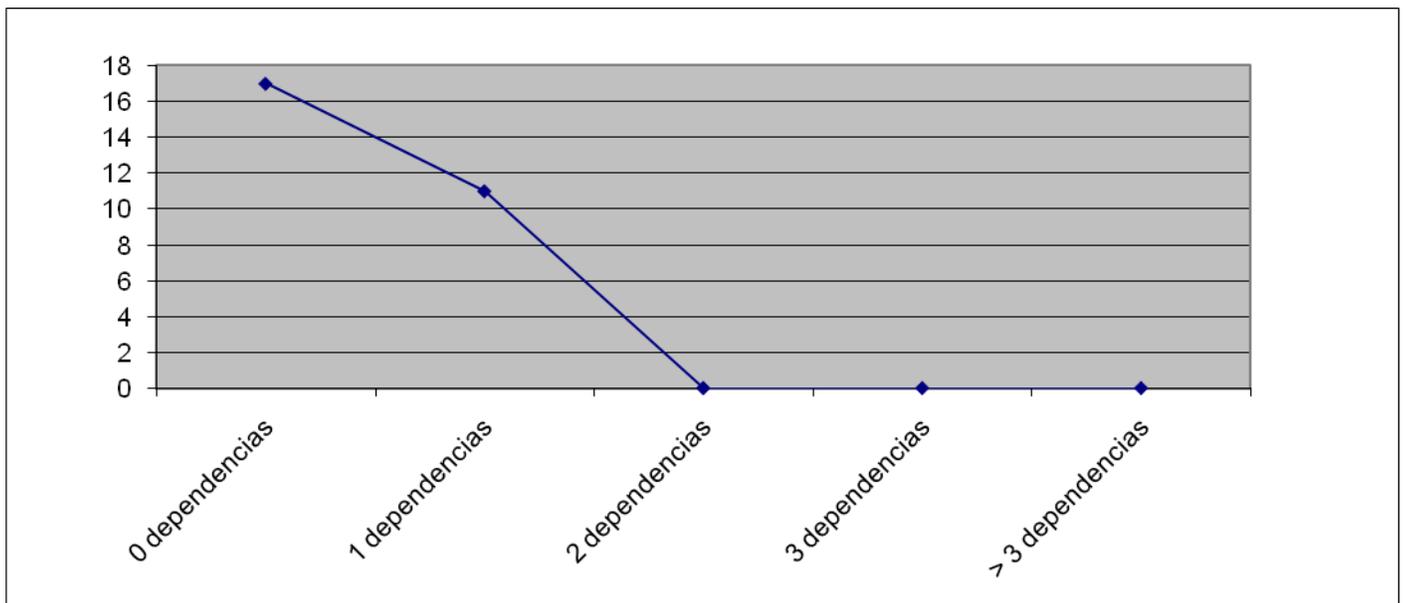


Figura 32 Gráfica de los resultados de la evaluación de la métrica RC agrupados por la tendencia de los valores

## GLOSARIO

**BPD:** Diagrama de procesos de Negocio.

**Capacitación:** Proceso permanente y planificado concebido como una inversión para el desarrollo que llevan a cabo las organizaciones empresariales con el objetivo de que sus trabajadores adquieran y perfeccionen su competencia laboral, de modo que estén aptos para responder a las exigencias constantemente cambiantes del proceso productivo de prestación de servicios del que forman parte.

**Cargos:** Según lo establecido en la Norma Cubana 3000 del 2007 Los que aparecen recogidos en los calificadores comunes, de rama o actividad y propios de organismos, pertenecientes a las diferentes categorías ocupacionales, así como en resoluciones. En ellos se definen la denominación, contenido de trabajo y los requisitos para ocuparlos y expresan el empleo u oficio que desempeñan, los trabajadores en la organización.

**Categoría ocupacional:** Según la norma cubana 3000 del 2007 la categoría ocupacional es la clasificación de los diferentes cargos, de acuerdo con las funciones y tareas que desarrollan los trabajadores, tomando en cuenta sus características disímiles, la complejidad y su vinculación con la producción y la prestación de servicios. Se agrupan en las categorías de operarios, trabajadores de servicios, trabajadores administrativos, técnicos y dirigentes.

**Centro de Costo:** Es la subdivisión mínima en el proceso de registro contable de los gastos en la entidad con el objetivo de permitir la medición de los recursos utilizados y los resultados económicos obtenidos.

**CIMATEL:** Empresa de Tecnologías de la Información y Servicios Telemáticos

**Contratación:** Es formalizar con apego a la ley la futura relación de trabajo para garantizar los intereses, derechos, tanto del trabajador como la empresa. Cuando ya se aceptaron las partes es necesario integrar su expediente de trabajo. La contratación se llevará a cabo entre la organización y el trabajador. La duración del contrato será por tiempo indeterminado o determinado. El contrato deberá ser firmado por el director general, el responsable directo y el trabajador.

**Crystal Reports 7.0:** Es una aplicación de inteligencia empresarial, o bien, inteligencia de negocios, utilizada para diseñar y generar informes desde una amplia gamas de fuentes de datos (bases de datos).

**Delphi:** Es un entorno de desarrollo de software diseñado para la programación de propósito general con énfasis en la programación visual. Se utiliza como lenguaje de programación. Se usa para el desarrollo de aplicaciones visuales y de base de datos.

**Evaluación del desempeño:** Según lo establecido en la Norma Cubana 3000 del 2007 es la medición sistemática e integral del grado de eficacia y eficiencia de la labor realizada por los trabajadores en la actividad que ejecutan, durante un período de tiempo y de su potencial de desarrollo en el ámbito de la empresa, identifica los tipos de insuficiencias y problemas del personal evaluado, sus fortalezas, posibilidades y capacidades, los caracteriza y constituye la base para la elaboración del plan de desarrollo, acorde a las necesidades. Repercute sobre la permanencia, promoción, formación, reconocimiento moral, idoneidad.

**Forma de pago:** Según lo establecido en la Norma Cubana 3000 del 2007 las formas de pago son las vías para remunerar el trabajo en función de su naturaleza. Sistemas de pago son las modalidades de las formas de pago que se adoptan en correspondencia con las características técnico-organizativas y la naturaleza del trabajo.

**Fuentes de reclutamiento:** Son los medios de los que se vale una organización para atraer candidatos adecuados a sus necesidades.

**HTML:** HyperText Markup Language en sus siglas en inglés (HTML), es el Lenguaje de Marcas de Hipertexto predominante para la construcción de páginas web. Usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

**Interfaz grafica:** La interfaz gráfica de usuario es el artefacto tecnológico de un sistema interactivo que posibilita, a través del uso y la representación del lenguaje visual, una interacción amigable con un sistema informático.

**Jubilación:** La jubilación es el nombre que recibe el acto administrativo por el cual un trabajador en activo, ya sea por cuenta propia o ajena, pasa a una situación pasiva (de inactividad laboral), tras haber alcanzado una determinada edad máxima legal para trabajar.

**Ley Federal del Trabajo:** La Ley Federal del Trabajo surge el 18 de agosto de 1931 durante el gobierno del presidente mexicano Pascual Ortiz Rubio, la importancia de sus normas es que estas regulan el equilibrio y la justicia social en la relación obrero-patronal. Son las que garantiza los derechos del obrero compensando su desigualdad respecto al patrón.

**Microsoft Excel:** Excel es una aplicación para manejar hojas de cálculos. Este programa sigue siendo desarrollado y distribuido por Microsoft, y es utilizado normalmente en tareas financieras y contables.

**Microsoft Office:** Esta compuesta básicamente por aplicaciones de procesamiento de textos, planilla de cálculo y programa para presentaciones. Fue desarrollada por la empresa Microsoft.

**Microsoft Word:** Microsoft Word es un software destinado al procesamiento de textos. Fue creado por la empresa Microsoft y actualmente viene integrado a Microsoft Office.

**Ministerio de Finanzas y Precios:** Organismo de la Administración Central del Estado que regula la políticas financieras y de precio para Cuba.

**NC:** Norma Cubana.

**Nóminas:** Según la Resolución No. 13-2007 del Ministerio De Finanzas y Precios de nuestro país el objetivo de la nómina es relacionar a todos los trabajadores de la entidad que perciban salarios y que les correspondan haberes por concepto de: sueldos, jornales, primas, vinculación, vacaciones, licencias y subsidios, obteniéndose la conformidad del cobro efectuado mediante la firma en este documento, siempre y cuando no se ejecute por Tarjetas Magnéticas.

**Pago adicional:** Según lo establecido en la Norma Cubana 3000 del 2007 son los pagos por trabajar en determinadas condiciones, cargos, actividades, ramas o sectores debido a su importancia económica, o del servicio que se presta, o como reconocimiento a trabajadores por sus aportes excepcionales y significativos al desarrollo económico y social del país y otros factores extracalificatorios.

**Proceso:** Según lo establecido en el apartado 3.4.1 de la NC ISO 9000:2005, el proceso es un conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados.

**Puestos de trabajo:** Según lo establecido en la Norma Cubana 3000 del 2007 es una zona o lugar en que se ejecuta la actividad laboral por un trabajador o grupo de trabajadores equipados con instrumentos y medios de trabajo necesarios para su realización.

**RAM:** Random Access Memory en sus siglas en ingles (RAM), es la memoria de acceso aleatorio desde donde el procesador recibe las instrucciones y guarda los resultados. Es el área de trabajo para la mayor parte del software de un computador.

**Remuneración y motivación de los trabajadores:** Se entiende por remuneración a los beneficios que adquiere el trabajador dado su desempeño, rendimiento, y cumplimiento por causa del Contrato de Trabajo. Como ejemplo de estos esta el salario.

**Salario:** Según lo establecido en la Norma Cubana 3000 del 2007 se considera salario la parte del producto nacional que se distribuye a los trabajadores de forma individual, atendiendo a la cantidad y calidad del trabajo aportado, según las condiciones económicas de cada momento histórico. Comprende lo percibido por el trabajador, por rendimiento, unidad de tiempo, pagos adicionales, trabajo extraordinario, laborar en día de conmemoración nacional y feriados y vacaciones anuales pagadas.

**SQL Server 2000:** Es un sistema de manejo de base de datos relacional producido por Microsoft. Tiene como finalidad analizar y administrar datos, dar mayor escalabilidad, y seguridad a las aplicaciones de análisis y los datos empresariales y potenciar las aplicaciones dando una mayor prestabilidad.

**Subsistema:** Cada uno de los componentes principales de un sistema que este dividido en componentes. Cada subsistema abarca aspectos del sistema que comparten alguna propiedad común.

**Visual Basic:** Visual Basic es un lenguaje de programación desarrollado por Alan Cooper para Microsoft. El cual integra el diseño e implementación de formularios de Windows. Permite usar con suma facilidad la plataforma de los sistemas Windows.

**Windows 98:** Es un sistema operativo, y se encarga de gestionar todos los procesos que ocurren en el ordenador. Es incluso mas rápido y eficiente que las versiones anteriores a el ya que entres sus ventajas permite almacenar mayor cantidad de datos, Los programas se ejecutan más rápidamente y el equipo utiliza menos recursos de sistema.