

**Universidad de las Ciencias Informáticas**



**Facultad 3**

**“Diseño e implementación del componente Nómina  
del subsistema Capital humano del Sistema Integral  
de Gestión Cedrux”**

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

Autores: Yarenis Echemendia González

Analina Benítez Guzmán

Tutora: Ing. Tania Teresa Loureiro Valladares

Co-Tutor: Ing. William González Obregón

La Habana, 2012

## **DECLARACIÓN DE AUTORÍA**

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

**Analina Benítez Guzmán**

**Autor**

---

**Yarenis Echemendia González**

**Autor**

---

**Ing. William González Obregón**

**Co-Tutor**

---

**Ing. Tania Teresa Loureiro Valladares**

**Tutor**

**Ing. Tania Teresa Loureiro Valladares.**

Ingeniera en Ciencias Informáticas, graduada de la Universidad de las Ciencias Informáticas en el 2009. Actualmente se desempeña como Jefa del proyecto Caja de la línea Finanzas del departamento Desarrollo de productos del centro CEIGE en la UCI.

Correo electrónico: [ttloureiro@uci.cu](mailto:ttloureiro@uci.cu).

**Ing. William González Obregón.**

Ingeniero en Ciencias Informáticas, graduado con título de oro en la Universidad de las Ciencias Informáticas en el 2010. Se desempeña como Arquitecto de la Línea Capital Humano del Departamento Desarrollo de Productos del centro CEIGE.

Correo electrónico: [wobregon@uci.cu](mailto:wobregon@uci.cu).

### **Analina**

*Este trabajo tiene detrás un gran número de personas que hicieron posible su realización, a todos ellos les debo mi agradecimiento.*

*Primeramente quisiera agradecer a mi tutor William porque siempre nos ayudó incluso en sus tiempos libres mostrándose siempre preocupado y atento.*

*A mi compañera de tesis que hizo que el trabajo se simplificara y me dio muchos dolores de cabeza.*

*A Soto por ser buena compañera.*

*A Donel por unirme a la familia que ha sido para mí Capital Humano.*

*A todos los integrantes de este proyecto por lo bien que me siento entre ellos.*

*A todos los que compartieron conmigo ya sea en el aula o en la residencia.*

*A todos los que hicieron mi estancia en esta universidad más alegre y amena.*

*A todos los que han formado parte de mi vida en estos últimos años muchas gracias.*

### **Yarenis**

*Quiero agradecer a todos los que me han ayudado de una forma u otra en estos cinco años.*

*Primeramente a mi mamá y a mi papá, por darme su apoyo incondicional y guiarme por el camino correcto. Sin sus consejos no hubiese logrado este sueño.*

*A mi hermano por servirme de ejemplo y motivación.*

*A mi tutor William, por tener mucha paciencia con nosotras y ayudarnos como lo ha hecho.*

*A mis amigos Mati, Pedro Luis y Leidy, por aguantar todas mis locuras sin protestar.*

*A Analina, por ser mi compañera de tesis y soportarme todo este tiempo.*

*A Donel por darme la oportunidad de formar parte de Capital Humano.*

*A mi familia, especialmente a tía Dori, Yelo y Dorita por ayudarme a confiar más en mí.*

*A Sergio por apoyarme y alegrar mi vida.*

*A todos mis compañeros de grupo, porque a pesar de que algunos me dieron muchos dolores de cabeza hicieron estos cinco años inolvidables.*

*Por último y no por eso menos importante al Comandante, por ser el impulsor del Proyecto Futuro y permitirme formarme como profesional.*

### **Analina**

*Quisiera dedicar esta tesis de forma muy especial a mis dos hermanitas por ese cariño que siempre encuentro en ellas aún cuando me he perdido gran parte de sus vidas, a ustedes porque las quiero y por ser las hermanas que todos quisieran tener.*

*A mi padre por estar tan orgulloso de mí, por apoyarme y por escucharme.*

*A mi mamá por ser la que me trajo a este mundo.*

*A mi abuela cascarrabias que se preocupa mucho por mí.*

*A William por el tiempo dedicado, el interés y empeño que puso en que saliera bien.*

*A Adolfo porque despertó mis deseos de hacer y batallar para lograr lo que quiero.*

*... A todos ustedes por lo agradecida que estoy de lo que cada uno aporta a mi vida es este trabajo.*

### **Yarenis**

*Quiero dedicar este trabajo especialmente a mis padres, por ser los principales impulsores de mi formación, por confiar en mí y sentirse tan orgullosos de la hija que tienen.*

*A mi hermano, por apoyarme mucho en mis primeros años de universidad y servirme de ejemplo y motivación.*

*A William, por formar parte de este trabajo, sin su ayuda no lo hubiésemos logrado.*

*A mi prima Dorita, por ser tan especial y seguir mis pasos. Me enorgullece saber que soy para ti un ejemplo a seguir.*

## **RESUMEN**

Siguiendo una estrategia del país, en la Universidad de las Ciencias Informáticas se desarrolla el Sistema Integral de Gestión Cedrux. Este contiene un módulo de gestión de Capital humano el cual incluye la implementación del proceso Estimulación Moral y Material. Una de las responsabilidades de este es garantizar el pago del salario de los trabajadores, siendo imprescindible para ello la generación de la nómina. El presente trabajo describe el diseño y la implementación de un sistema informático responsable de generar la nómina, para lo cual se realizó un estudio de otros sistemas que efectuaban dicho proceso, concluyendo que era necesario construir una solución propia. Las herramientas y tecnologías utilizadas para su confección fueron las definidas por el CEIGE<sup>1</sup>.

La nueva solución informática contribuye al cumplimiento de las políticas de migración hacia el software libre, permite llevar a cabo la remuneración del trabajador mediante el sistema Cedrux, mantener actualizados los registros de los submayores de vacaciones y retenciones y los datos contables de los que dependen otros subsistemas, así como garantizar la integridad de la información.

**Palabras claves:** capital humano, Cedrux, estimulación material, gestión de capital humano, nómina.

---

<sup>1</sup> CEIGE: Centro de Informatización de la Gestión de Entidades.

## ÍNDICE

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>5</b>
1.1 INTRODUCCIÓN .....	5
1.2 PRINCIPALES CONCEPTOS .....	5
1.3 SISTEMAS QUE INCLUYEN EL PROCESO DE LA NÓMINA. ....	6
1.3.1 SAGE MAS 500 .....	6
1.3.2 Seven2000 .....	7
1.3.3 Assets.....	8
1.3.4 SAP .....	8
1.3.5 Versat-Sarasola.....	9
1.3.6 Rodas XXI .....	10
1.4 MODELO DE DESARROLLO DEL SOFTWARE.....	12
1.5 TECNOLOGÍAS Y HERRAMIENTAS PARA EL DESARROLLO .....	13
1.5.1 Marco de trabajo.....	14
1.5.2 Lenguajes de programación .....	15
1.5.3 Herramientas para el modelado.....	16
1.5.4 Herramientas para el desarrollo colaborativo.....	16
1.5.5 Gestor de Base de Datos .....	16
1.5.6 Servidor Web.....	16
1.5.7 Navegador Web.....	17
1.6 PATRONES DE DISEÑO.....	17
1.7 PATRONES DE ARQUITECTURA .....	18
1.8 CONCLUSIONES PARCIALES .....	18
<b>CAPÍTULO 2: PROPUESTA DE SOLUCIÓN</b> .....	<b>20</b>
2.1. INTRODUCCIÓN .....	20
2.2. VALORACIÓN DEL DISEÑO ANTERIOR .....	20
2.3. DISEÑO DEL SISTEMA .....	21
2.3.1. Patrones de diseño utilizados .....	21

2.3.2.	Patrones de arquitectura utilizados .....	22
2.3.3.	Diagrama de clases .....	23
2.3.4.	Modelo de datos .....	27
2.4.	IMPLEMENTACIÓN .....	29
2.4.1.	Estructura del componente Nómina.....	29
2.4.2.	Descripción del componente.....	31
2.4.3.	Diagrama de componentes.....	31
2.4.4.	Servicios utilizados .....	33
2.4.4.1.	Servicios que consume nómina.....	34
2.4.4.2.	Servicios que brinda nómina .....	35
2.4.5.	Diagrama de despliegue.....	36
2.4.6.	Funcionalidades implementadas.....	37
2.5.	CONCLUSIONES PARCIALES .....	41
<b>CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN .....</b>		<b>42</b>
3.1	INTRODUCCIÓN.....	42
3.2	VALIDACIÓN DEL MODELO DE DISEÑO PROPUESTO .....	42
3.2.1	Métrica Tamaño Operacional de Clase.....	42
3.2.2	Métrica Relaciones entre Clases .....	46
3.3	PRUEBAS DE SOFTWARE .....	50
3.3.1	Pruebas de Caja blanca o Estructurales .....	51
3.3.2	Pruebas de Caja negra.....	57
3.4	CONCLUSIONES PARCIALES .....	63
<b>CONCLUSIONES.....</b>		<b>64</b>
<b>RECOMENDACIONES.....</b>		<b>65</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>		<b>66</b>
<b>GLOSARIO DE TÉRMINOS .....</b>		<b>70</b>



## ÍNDICE DE FIGURAS

FIGURA 1. SECUENCIA DE ACTIVIDADES DEL MODELO DE DESARROLLO DEL SOFTWARE. ....	13
FIGURA 2. ARQUITECTURA DE SAUXE. ....	14
FIGURA 3. REPRESENTACIÓN GENERAL DE PATRÓN MVC EN LOS DIAGRAMAS DE CLASES. ....	23
FIGURA 4. DIAGRAMA DE CLASES DEL DISEÑO ADMINISTRAR NÓMINA. ....	24
FIGURA 5. DIAGRAMA DE CLASES DEL DISEÑO GESTIONAR TIPO NÓMINA. ....	25
FIGURA 6. DIAGRAMA DE CLASES DEL DISEÑO REVISAR NÓMINA. ....	26
FIGURA 7. MODELO DE DATOS DEL COMPONENTE NÓMINA. ....	28
FIGURA 8. CONTENIDO DE LAS CARPETAS APPS Y WEB PARA CAPITAL HUMANO. ....	29
FIGURA 9. ESTRUCTURA DE LA CARPETA NÓMINA. ....	29
FIGURA 10. ESTRUCTURA DE LA CARPETA NÓMINA. ....	30
FIGURA 11. DIAGRAMA DE COMPONENTES INTERNOS. ....	32
FIGURA 12. DIAGRAMA DE COMPONENTES EXTERNOS. ....	32
FIGURA 13. CONTENIDO DEL FICHERO IOC.XML INTERNO. ....	33
FIGURA 14. CONTENIDO DEL FICHERO IOC.XML EXTERNO. ....	34
FIGURA 15. DIAGRAMA DE DESPLIEGUE DE ESCENARIO PARA PC CLIENTE SIN DISCO. ....	37
FIGURA 16. DIAGRAMA DE DESPLIEGUE DE ESCENARIO PARA PC CLIENTE CON DISCO. ....	37
FIGURA 17. INTERFAZ ADMINISTRAR NÓMINA. ....	38
FIGURA 18. VISIBILIDAD DE LOS CAMPOS SEGÚN EL TIPO DE NÓMINA. ....	38
FIGURA 19. INTERFAZ REVISAR NÓMINA. ....	39
FIGURA 20. INTERFAZ GESTIONAR TIPO DE NÓMINA. ....	40
FIGURA 21. RESULTADOS OBTENIDOS DE LA APLICACIÓN DE LA MÉTRICA TOC EN EL INSTRUMENTO AGRUPADOS EN LOS INTERVALOS DEFINIDOS. ....	45
FIGURA 22. REPRESENTACIÓN EN % DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA TOC EN EL ATRIBUTO RESPONSABILIDAD. ....	45
FIGURA 23. REPRESENTACIÓN EN % DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA TOC EN EL ATRIBUTO COMPLEJIDAD DE IMPLEMENTACIÓN. ....	46
FIGURA 24. REPRESENTACIÓN EN % DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA TOC EN EL ATRIBUTO REUTILIZACIÓN. ....	46
FIGURA 25. RESULTADOS OBTENIDOS DE LA APLICACIÓN DE LA MÉTRICA RC EN EL INSTRUMENTO AGRUPADOS EN LOS INTERVALOS DEFINIDOS. ....	49
FIGURA 26. REPRESENTACIÓN EN % DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA RC EN EL ATRIBUTO ACOPLAMIENTO. ....	49
FIGURA 27. REPRESENTACIÓN EN % DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA RC EN EL ATRIBUTO COMPLEJIDAD DE MANTENIMIENTO. ....	49
FIGURA 28. REPRESENTACIÓN EN % DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA RC EN EL ATRIBUTO REUTILIZACIÓN. ....	50
FIGURA 29. REPRESENTACIÓN EN % DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA RC EN EL ATRIBUTO CANTIDAD DE PRUEBAS. ....	50
FIGURA 30. REPRESENTACIÓN DE TÉCNICA DE PRUEBAS DE CAJA BLANCA. ....	51
FIGURA 31. NOTACIÓN DE GRAFOS DE FLUJO PARA LAS INSTRUCCIONES: SECUENCIALES, SI, MIENTRAS. ....	52
FIGURA 32. MÉTODO QUE ADICIONA TRABAJADORES A LA NÓMINA. ....	53
FIGURA 33. GRAFO DE FLUJO ASOCIADO AL MÉTODO. ....	53
FIGURA 34. REPRESENTACIÓN DE TÉCNICA DE PRUEBA DE CAJA NEGRA. ....	57

## ÍNDICE DE TABLAS

TABLA 1. RESULTADOS DEL ESTUDIO DE LOS SISTEMAS ANALIZADOS.....	11
TABLA 2. MÉTRICA TAMAÑO OPERACIONAL DE CLASE. ....	43
TABLA 3. RANGO DE VALORES PARA LA EVALUACIÓN TÉCNICA DE LOS ATRIBUTOS DE CALIDAD RELACIONADOS CON LA MÉTRICA TOC.....	43
TABLA 4. RESULTADOS DE LA APLICACIÓN DE LA MÉTRICA TOC PARA CADA CLASE DEL SISTEMA. ....	44
TABLA 5. RELACIONES ENTRE CLASES. ....	46
TABLA 6. RANGO DE VALORES PARA LA EVALUACIÓN TÉCNICA DE LOS ATRIBUTOS DE CALIDAD RELACIONADOS CON LA MÉTRICA RC. ....	47
TABLA 7. RESULTADOS DE LA APLICACIÓN DE LA MÉTRICA RC PARA CADA CLASE DEL SISTEMA. ....	47
TABLA 8. CASO DE PRUEBA PARA EL CAMINO BÁSICO 1 .....	55
TABLA 9. CASO DE PRUEBA PARA EL CAMINO BÁSICO 2 .....	55
TABLA 10. CASO DE PRUEBA PARA EL CAMINO BÁSICO 3 .....	56
TABLA 11. POSIBLES ESCENARIOS AL ADICIONAR UNA NÓMINA. ....	59
TABLA 12. DESCRIPCIÓN DE LAS VARIABLES DEL ADICIONAR NÓMINA.....	60
TABLA 13. RESPUESTAS DEL SISTEMA PARA LOS POSIBLES ESCENARIOS AL ADICIONAR UNA NÓMINA. ....	60
TABLA 14. RESULTADO DE LAS ITERACIONES DE LAS PRUEBAS DE CAJA NEGRA REALIZADAS POR CALISOFT. ....	63

## INTRODUCCIÓN

El capital humano constituye un elemento fundamental e indispensable en una empresa. Su adecuada gestión influye de manera decisiva en las posibilidades de alcanzar las metas y objetivos propuestos por una entidad. En la actualidad se requiere de un personal más preparado, capaz de adaptarse fácilmente a los cambios de los ambientes en los que se desarrolla. Todo esto representa un reto y una motivación para adoptar nuevas estrategias que permitan perfeccionar la gestión del capital humano, orientada esencialmente al desarrollo del valor intangible de los recursos humanos. Esta gestión se concibe como la integración de varios procesos: Captación, selección, contratación del personal, evaluación del desempeño, capacitación, motivación, remuneración, estimulación y promoción. (1)

La gestión del capital humano ha cobrado mayor importancia cada día en las empresas. La necesidad de aprovechar al máximo su potencial, se ha convertido en una prioridad en estos tiempos. Existen sistemas de información que integran y manejan todas las áreas funcionales tradicionales de una organización, conocidos por sus siglas en inglés como ERP (*Enterprise Resource Planning*) (2). Estos incluyen entre sus funcionalidades algunas encargadas de la gestión del capital humano.

Siguiendo una estrategia del país, en la Universidad de las Ciencias Informáticas se desarrolla el Sistema Integral de Gestión Cedrux, basado en los principios de independencia tecnológica y con particularidades de la economía cubana. Este cuenta con un subsistema dedicado a la gestión del capital humano, que incluye el macro proceso Estimulación Moral y Material como parte de los procesos que velan por la adecuada gestión del personal implicado.

La estimulación moral de los trabajadores describe el sistema de acciones realizadas en una entidad con el fin de reconocer el aporte laboral del trabajador en el logro de los objetivos y propiciar el sentido de pertenencia. Mientras que la estimulación material es la encargada de registrar los procesos relacionados con beneficios materiales que reciben los trabajadores, para de esta forma remunerarlos por las labores realizadas dentro de la entidad. El pago de los salarios que les correspondan a los trabajadores de acuerdo con la escala salarial y cargo que desempeñen, es una de las formas mediante la cual se lleva a cabo la estimulación material en las empresas, siendo necesario para esto realizar el procesamiento de las nóminas, responsables de mostrar los salarios, retenciones, tiempo de trabajo y demás datos de cada trabajador en un período de pago. Como resultado de dicho procesamiento se obtiene un grupo de salidas, que además de garantizar el pago de los salarios permiten mantener actualizado todo el proceso

de la relación monetaria empresa-empleado y una serie de informaciones que se depositan periódicamente al subsistema de Contabilidad. Estas salidas pueden resumirse en: nóminas en sus diferentes tipos, datos necesarios para la actualización del submayor de vacaciones y el submayor de retenciones, el impuesto por la utilización de la fuerza de trabajo, la contribución y la contribución especial a la seguridad social.

En las entidades cubanas, los procesos mencionados con anterioridad, se realizan de diversos modos atendiendo a las particularidades de cada una. Algunas los realizan de forma manual y otras a través de diferentes sistemas informáticos. Cedrux es una alternativa del país para erradicar la diversidad de formas de realizarlos, pero actualmente llevar a cabo el proceso de remuneración del trabajador mediante el mismo se dificulta, ya que para el procesamiento de la nómina deben tomarse los datos necesarios del sistema y hacer las operaciones que implican su cálculo manualmente. Luego de obtener las salidas resultantes del procesamiento, se deben introducir los datos necesarios al sistema para actualizar los submayores y registrar los valores de los impuestos por la utilización de la fuerza de trabajo y las contribuciones especiales, debido a que son datos que el subsistema Capital humano debe manejar, para finalmente registrarlos en un comprobante de operaciones que es emitido a Contabilidad, pues este último es el subsistema que maneja la información contable, que incluye las cuentas y las operaciones que se realizan sobre estas. La realización de estas acciones puede conducir a la pérdida de datos, lo que trae como consecuencias errores en el descuento de las retenciones provocando que el trabajador reciba un salario inferior al que le corresponde, el cálculo incorrecto del aporte a la seguridad social ocasionando que no se ingrese en la cuenta del estado el monto adecuado, deducción errónea del acumulado de vacaciones que le pertenece al trabajador propiciando la violación de uno de sus derechos y un mal procesamiento de la información relacionada con los subsidios que puede inducir perjuicios en los empleados y demás personas implicadas. Todo esto produce un escenario favorable para las ilegalidades afectando la integridad de la información, comprometiendo su corrección y completitud, causando que la empresa incurra en procesos penales, además de lo engorroso que resulta su ejecución.

Por la situación antes expuesta se plantea como **problema a resolver**: La gestión manual de las nóminas afecta la integridad de su información como entrada del proceso de Estimulación Moral y Material del subsistema Capital humano del Sistema Integral de Gestión Cedrux.

Con el objetivo de solucionar el problema planteado anteriormente el **objeto de estudio** queda enmarcado en los Sistemas de Gestión del Capital humano, y delimitado el **campo de acción** en los Sistemas de Gestión de la Nómina.

Para llevar a cabo este trabajo se planteó como **objetivo general**: Realizar el diseño e implementación del componente Nómina de manera que se garantice la integridad de su información como entrada del proceso de Estimulación Moral y Material en el subsistema Capital humano del Sistema Integral de Gestión Cedrux.

Para dar cumplimiento al objetivo general se definieron los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación que permita identificar los principales logros y limitaciones en cuanto a la gestión de la nómina.
- Diseñar el componente Nómina basado en los requisitos previamente identificados.
- Validar el diseño propuesto mediante la utilización de métricas.
- Implementar el componente Nómina teniendo en cuenta el diseño realizado.
- Validar el componente obtenido mediante la realización de pruebas de caja blanca y caja negra.

Las **tareas a cumplir** son las siguientes:

- Actualización de los principales conceptos asociados a la gestión de la nómina como parte del proceso de Estimulación Moral y Material.
- Análisis de los sistemas nacionales e internacionales existentes para la gestión de la nómina identificando sus características y deficiencias fundamentales.
- Caracterización de las tecnologías, lenguajes y herramientas propuestas para el desarrollo del componente.
- Valoración del diseño anterior teniendo en cuenta los cambios ocurridos en los requisitos.
- Diseño del componente Nómina basado en los requisitos previamente identificados y la valoración realizada.
- Validación del diseño mediante la aplicación de métricas.
- Implementación del componente Nómina teniendo en cuenta el diseño realizado.
- Realización de pruebas de caja blanca a las principales funcionalidades implementadas.
- Validación del componente obtenido mediante la realización de pruebas de caja negra.

Formulándose como **idea a defender** que si se realiza el diseño e implementación del componente Nómina entonces se logrará la integridad de su información que constituye una entrada del proceso de Estimulación Moral y Material del subsistema Capital humano del Sistema Integral de Gestión Cedrux.

Con la realización de este trabajo se espera obtener los siguientes resultados:

- El componente Nómina del subsistema Capital humano del Sistema Integral de Gestión Cedrux.
- La documentación generada en los flujos de trabajo diseño e implementación del componente Nómina.

**La estructura del trabajo de diploma será la siguiente:**

**CAPÍTULO 1: Fundamentación teórica:** Describe el estado del arte de los sistemas que gestionan el capital humano, los principales conceptos relacionados, así como la fundamentación del uso de los lenguajes, herramientas y modelo de desarrollo.

**CAPÍTULO 2: Propuesta de solución:** Realiza una valoración crítica al diseño existente teniendo en cuenta su importancia para la implementación, propone un nuevo diseño y describe la implementación de la solución.

**CAPÍTULO 3: Validación de la solución:** Se muestran los resultados obtenidos de la validación del diseño, así como del componente.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

En este capítulo se abordan los fundamentos teóricos asociados a la investigación, se enuncian las definiciones de los conceptos relacionados con la administración del capital humano, estimulación moral y material y nómina, así como la relación existente entre ellos. Se realiza un estudio de diferentes sistemas existentes en Cuba y el mundo que llevan a cabo la gestión de la nómina y una descripción de las tecnologías utilizadas para el desarrollo de la solución propuesta.

### 1.2 Principales conceptos

La Administración del capital humano consiste en la planeación, organización, desarrollo y coordinación, así como el control de técnicas, capaces de promover el desempeño eficiente del personal (1). La estimulación moral y material recogida dentro de dicha administración es realizada con el objetivo de conquistar y mantener a las personas en la organización, trabajando y dando el máximo de sí. La norma cubana 3000 de 2007 define estimulación material como: un “sistema de acciones que interactúan y se integran con la estimulación moral, para motivar a los trabajadores en el logro de los objetivos estratégicos de la organización. El pago con arreglo al trabajo, por cantidad y calidad, es el elemento principal de esta”, mientras que, la estimulación moral es el “sistema de acciones que se realizan para propiciar el desarrollo de la moral socialista en el trabajo y el sentido de pertenencia; reconocer y promover el aporte laboral de los trabajadores en la obtención de los objetivos estratégicos y la elevación de la cultura de la organización, así como la satisfacción individual y colectiva de los trabajadores” (3).

Como elemento esencial dentro de este proceso se encuentra la nómina, herramienta a través de la cual se retribuyen al trabajador los servicios prestados dentro de una entidad. Esta palabra, que tiene su origen en el término latino *nomina*, es recogido por la Real Academia Española como: la relación nominal de los individuos que, en una oficina ya sea pública o particular, pueden ver los salarios y justificar con su firma haberlos recibido (4). Este término ha evolucionado y lo que en un principio fue una “lista conformada por el conjunto de trabajadores a los cuales se les va a remunerar por los servicios que estos le prestan al patrono” (5) se convirtió en un “instrumento que permite de una manera ordenada, realizar el pago de sueldos o salarios a los trabajadores, así como proporcionar información contable y estadística, tanto para la empresa como para el ente encargado de regular las relaciones laborales” (5).

Una nómina también se define como un documento que, a modo de recibo de salario individual, en el que la empresa acredita el pago de las diferentes cantidades que forman el salario. En ella quedan registradas también las deducciones que se realizan sobre el salario, básicamente las cuotas al obrero y las retenciones a la cuenta de este (6).

Luego del análisis de varios conceptos de nómina se define esta como: una lista conformada por los salarios, retenciones y otros datos relacionados con los trabajadores de una entidad, que ofrecerá información contable y estadística para la empresa y el organismo encargado de regular las relaciones laborales. Permite realizar el pago de los salarios y justificar por medio de la firma del trabajador el haberlos recibido.

### **1.3 Sistemas que incluyen el proceso de la nómina.**

En la actualidad existen diferentes sistemas informáticos que se encargan de la realización del proceso de la nómina, por lo que es importante valorar sus particularidades y analizar su reutilización atendiendo a las siguientes características:

1. Software que permita el acceso a las funcionalidades, al código o parte de este.
2. Acatamiento de las normas cubanas encargadas de regular el sistema de gestión del capital humano.
3. Dependencias de otros componentes o módulos para efectuar las funciones de la nómina.
4. Ejecución de todas las funcionalidades necesarias para cumplir los requerimientos detectados para el correcto funcionamiento del componente Nómina que se desea desarrollar.
5. Cumplimiento con el paradigma de independencia tecnológica del país.

#### **1.3.1 SAGE MAS 500**

Sage MAS 500 es un sistema que ofrece soluciones integradas que automatizan todas las áreas de administración de empresas, para ello separa las soluciones en los siguientes componentes:

- Recursos Humanos y Nómina
- Negocios Electrónicos
- Manufactura Discreta.
- Inteligencia de Negocios
- Distribución al por Mayor
- Contabilidad del Proyecto
- Aspectos de Finanzas Centrales y Avanzados.
- Administración de Relaciones con los Clientes (CRM).



- Administración de Centros de Distribución

Dentro de sus módulos, Abra nómina es el encargado de realizar el proceso de la nómina. La integración de este módulo con Recursos Humanos le permite la gestión de asistencia necesaria para el procesamiento de la nómina y que los empleados vean los beneficios adquiridos y saldos de tiempo libre. Su construcción se realizó desde cero con Microsoft SQL Server, Visual Basic y Microsoft Office. (7)

Para el correcto funcionamiento de este software es preciso comprar además de los módulos deseados los que guardan relación con estos, como por ejemplo: Abra nómina necesita de Recursos Humanos, por lo que si solo se adquiere el primero quedarían funcionalidades sin utilizarse (8). En caso de obtener solamente el Abra nómina es imprescindible conocer cómo interactuar con él, qué estructura tienen los datos enviados de Recursos Humanos y cuáles son, con el objetivo de lograr que todas las funciones de la nómina se ejecuten correctamente.

La documentación del sitio blytheco.com (9) permite acceder al demo del software solamente para personas residentes en Estados Unidos de América (EEUU) y el sitio oficial sagemas.com (7) tampoco ofrece acceso al producto, ni a información del precio de adquisición del software. Las documentaciones encontradas en los sitios antes mencionados describen que las funcionalidades de los productos no permiten ser cambiadas excepto cuando sean actualizadas o mejoradas por los propios proveedores, que el componente Nómina es capaz de asignar un código único a cada nómina, hacer nóminas de prueba, descontar impuestos, eliminar la entrada de datos duplicados e incluye además una gama completa de gestión de impuestos; pero como es un producto comercial no ofrece detalles de cómo realiza cada una de estas funcionalidades.

### **1.3.2 Seven2000**

Seven2000 es un sistema integrado, compuesto por módulos que pueden actuar separados o integrados. Entre sus principales características cuenta con el subsistema de pagos de nóminas que incluye liquidación de nómina normal y de aportes a la seguridad social, control de descuentos de diversos tipos de embargos y liquidación de impuestos a empleados (retenciones). (10)

Permite procesar un número ilimitado de conceptos, la definición de la forma de calcular estos es totalmente parametrizado y no requiere intervención de personal de informática (los cambios los realiza el usuario del sistema). Posibilita procesar múltiples contratos de nómina siendo la definición de estas determinada por el usuario (semanal, quincenal, mensual y contratado). Maneja diferentes turnos de

trabajo los cuales son igualmente determinados por el usuario (fijos, rotativos, eventuales, planificada por mes/año). (10)

Este software está diseñado para evitar la acumulación de documentos dentro del departamento de personal, de forma tal que permita incluir variaciones de nómina a futuro o en el momento que ocurren y se procesarán dependiendo de la fecha. Algunos de los requerimientos de este software expresa que el cliente debe tener como sistema operativo Windows XP o superior, no permitiendo la utilización de otro software como Linux, (11) requerimiento que contrasta con la política de migración y soberanía tecnológica del país. La documentación de este software es estrictamente comercial por lo que no ofrece detalles de funcionalidades de la nómina y no permite acceder a él.

### 1.3.3 Assets

ASSETS NS es un Sistema de Gestión Integral creado por Assets S.A.: una compañía de desarrollo de aplicaciones que posee representación en Cuba. Todos sus módulos trabajan en estrecha relación (12) como por ejemplo el módulo de Recursos Humanos concebido para calcular nóminas de vacaciones, subsidios, nominillas de diferentes tipos (salario, vacaciones, subsidios, reintegros, estimulación, entre otras) y nóminas de salario según su tipo de pago (sueldo fijo, jornal, pago por rendimiento) incluyendo el pago de horas extras, interrupciones y condiciones laborales anormales, necesita del componente Expedientes de retenciones para realizar el pago a los trabajadores.

Este sistema está desarrollado sobre Visual Basic y su código es invisible para los clientes, por lo que no puede utilizarse un componente que dependa de otro ya que no permite conocer con exactitud los datos y estructura que estos deben tener para que trabaje correctamente de forma independiente. Su desarrollo se realizó con herramientas privativas.

### 1.3.4 SAP

El software ERP de SAP comprende cuatro soluciones independientes que brindan soporte a procesos de negocio clave a través de su sistema ERP específico: SAP ERP Financials, SAP ERP Human Capital Management, SAP ERP Operations y SAP ERP Corporate Services. (13)

La solución SAP ERP *Human Capital Management* (SAP ERP HCM) optimiza la contribución de cada empleado coordinando aptitudes, actividades e incentivos de los empleados con los objetivos empresariales y las estrategias necesarias para alcanzarlos. También proporciona herramientas para

gestionar, medir y recompensar las contribuciones individuales y colectivas y ofrece funciones de cálculo de nómina. (13)

Este componente puede ser utilizado para calcular la remuneración por el trabajo realizado de cada empleado así como realizar otra serie de procesos encaminados al suministro de beneficios y el bienestar del trabajador. Está integrado con otros que son imprescindibles para su correcto funcionamiento por ejemplo el componente Contabilidad y finanzas encargado de llevar toda la información de los gastos y deudas de la nómina los cuales deben ser contabilizados directamente en este otro componente así como asignar los costos para el centro de coste adecuado.

Algunas de las principales características de este sistema es que calcula el salario bruto y neto, que comprende los pagos individuales y las deducciones que se calculan durante un período de nómina, y son recibidos por un empleado. Estos pagos y deducciones se incluyen en el cálculo de la nómina de la cual luego se puede ver el pago de los salarios, permite hacer evaluaciones de esta así como otros datos relacionados.

El módulo de nóminas de este componente es el más específico a cada país ya que las leyes que rigen las relaciones laborales difieren mucho de un país a otro. Es por ello que SAP crea un componente diferente para cada país para cumplir con las normativas fiscales de este. Por tal motivo en algunos países que implementan el componente, no están disponibles todas las funciones del sistema. (14) Cedrux para un correcto funcionamiento del resto de los componentes ya desarrollados necesita un componente Nómina que incluya en los tipos de nómina los conceptos de pago asociados a este, opción que no está disponible en dicho software.

SAP utiliza para su creación software propietario lo que contrasta con la política de migración a software libre del país y además se observa que este posee gran cantidad de las funcionalidades requeridas por el componente Nómina que se desea desarrollar, pero no recoge todos los requisitos de las Normas cubanas.

### **1.3.5 Versat-Sarasola**

Es un sistema cubano de contabilidad diseñado para ser empleado en cualquier tipo de entidad empresarial o presupuestada, permite enviar información de forma inmediata, desde lugares apartados, a la vez que ofrece mayor organización, control y disciplina en cada gestión. Fue desarrollado por la Empresa de Tecnologías de la Información y el Conocimiento (TEICO) del Ministerio del Azúcar (actualmente Azcuba) en una plataforma de Delphi 5 con servidor SQL Server. Está constituido por 12

módulos o subsistemas que incluyen configuración y seguridad, contabilidad general y de gastos, costos y procesos, análisis económico empresarial y control de activos fijos. Además, interviene finanzas y cajas, planificación y presupuestos, control de inventarios, de productos terminados, pago de salario (nómina), paquete de gestión, contratación y facturación. (15)

Dentro de los Subsistemas del Versat se puede encontrar el subsistema Nómina de Salarios. Este se adecua a diversos sistemas de pago, estimulaciones y demás regulaciones laborales que existen en el país. Posee además todo lo relacionado con el descuento, liquidación y submayor de retenciones, así como vacaciones acumuladas. Se puede configurar el sistema definiendo de antemano todas las variantes de impuestos, bonificaciones, condiciones anormales, retenciones, descuentos, penalizaciones, entre otros con las denominaciones y los parámetros numéricos que se requieran. Mediante los conceptos de pago puede definir todas las variantes de cálculos y procedimientos contables que se presenten en su entidad y designarlas con las denominaciones más naturales. (16)

Este sistema presenta varias funciones útiles que sirvieron para conocer que era necesario descontar el 5% del salario para su aporte al presupuesto del estado y seguridad social, hacer variable el número de retenciones y actualizarlas con cada pago del trabajador haciendo así el componente más completo. Aunque aportó información para el componente que se desea desarrollar, no es una solución viable ya que está desarrollado con software propietario no cumpliendo así con las políticas de migración hacia software libre del país, además no cumple con la estructura que necesita el componente Nómina para enviar datos a otros subsistemas de Cedrux.

### **1.3.6 Rodas XXI**

Sistema Integral Económico- Administrativo creado por CITMATEL para la automatización de la gestión empresarial. Es un sistema multiempresa que cuenta actualmente con diferentes módulos: Recursos Humanos, Finanzas, Contabilidad, Activos Fijos, Nóminas, Inventario y Facturación. Estos módulos pueden emplearse integrados en su totalidad, formando cualquier subconjunto entre ellos, o cada uno de forma independiente. (17)

El módulo de Nóminas de RODAS XXI brinda grandes facilidades a las entidades para el cálculo y emisión de sus nóminas, permitiendo calcular y emitir todos los tipos de nóminas que se utilizan en Cuba. Para la obtención de las nóminas en cada uno de los períodos del año, el sistema divide las operaciones en dos pasos fundamentales, el cálculo de las nóminas y su emisión. Como resultado del cálculo de cada nómina

y su comprobante, el sistema muestra cómo quedarían en caso de emitirse dicha nómina, si se detecta algún error u omisión en las incidencias agregadas a los trabajadores que les corresponda o en el comprobante calculado, se pueden realizar las correcciones pertinentes, calculándola nuevamente hasta que todo está correcto y una vez logrado esto puede procederse a emitirla. Esto implica, que en el sistema a nivel de módulo, se registrará la nómina emitida en el período que se está trabajando, así como su comprobante correspondiente. Una vez emitida la nómina el comprobante puede ser exportado al módulo de contabilidad facilitando en gran medida el trabajo del contador.

El sistema es capaz de guardar, por trabajador, los pagos y retenciones fijas que se realizan a cada uno de ellos por lo que para la confección de las nóminas cada mes solo es necesario actualizar las incidencias que correspondan y todo el trabajo posterior de cálculo es realizado de forma automática. Siempre que se desee se pueden ver los reportes correspondientes a cada una de las nóminas emitidas, la nómina en sí, su comprobante, el desglose de efectivo y las retenciones realizadas. Este módulo permite además visualizar información correspondiente a períodos anteriores, tan solo con cambiar de período contable a períodos anteriores ya cerrados, aunque en dichos períodos no podrá realizar ninguna operación. Esta característica es compartida por todos los módulos de Rodas XXI. (18)

El sistema contable Rodas XXI tiene funcionalidades muy útiles pero está desarrollado para plataforma de software propietario, además no se conoce la estructura de los datos que envía al módulo de contabilidad y tampoco cuáles son estos datos.

### Resultados del estudio de los sistemas informáticos

Después del estudio de los sistemas anteriores se refleja en la siguiente tabla las características que estos deben cumplir para ser utilizados. Las características deseadas aparecen por las filas y por las columnas los sistemas analizados.

Tabla 1.Resultados del estudio de los sistemas analizados.

Características	Sage Mas 500	Seven 2000	Assets	SAP	Versat Sarasola	Rodas XXI
Software accesible.	No	No	No	No	Sí	No
Cumplimiento de las normas cubanas.	No	No	Sí	Sí	Sí	Sí

Dependencias para efectuar las funciones de la nómina.	Sí	No	Sí	Sí	Sí	No
Cumplimiento con el paradigma de independencia tecnológica del país.	No	No	No	No	No	No
Ejecución de todas las funcionalidades necesarias.	No	Se desconoce	Se desconoce	No	Sí	No

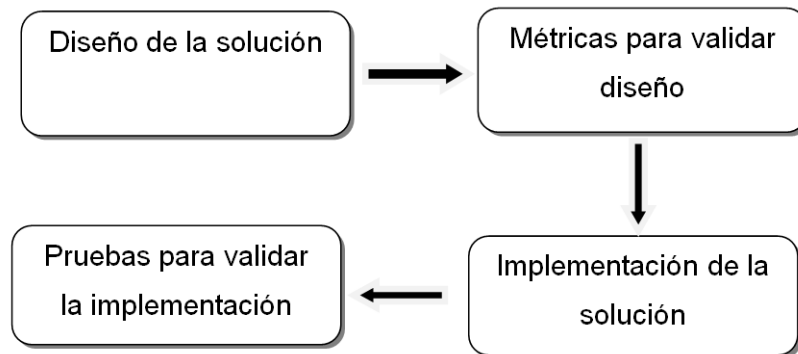
Luego de haber estudiado y analizado las características de los componentes seleccionados se puede concluir que ninguno cumple con todas las características deseadas, por lo que sería más factible crear un sistema que responda a las particularidades de la economía cubana, que incluya las características positivas de los estudiados y sea consecuente con el paradigma de independencia tecnológica por el cual apuesta el país, para lo cual sería imprescindible la utilización de herramientas y tecnologías libres para el desarrollo de la solución.

### 1.4 Modelo de desarrollo del software

El modelo de desarrollo de software propuesto por el Centro de Informatización de la Gestión de Entidades (CEIGE) describe la secuencia de actividades de alto nivel para la construcción y desarrollo de soluciones. Se logra con la combinación entre los modelos Orientado a componentes e Iterativo e incremental (19), el mismo presenta las siguientes características:

- ✓ Centrado en la arquitectura
- ✓ Iterativo e Incremental
- ✓ Orientado a componentes
- ✓ Ágil y adaptable al cambio

En la figura 1 se muestran las actividades que se aplicarán en el modelo de desarrollo de la propuesta de solución del presente trabajo.



**Figura 1: Secuencia de actividades del modelo de desarrollo del software.**

A continuación se muestran los artefactos que se generan en el modelo de desarrollo, teniendo en cuenta la estructura del presente trabajo.

### **Diseño de la solución**

- Modelo de datos.
- Diagramas de clases del diseño.

### **Métricas para validar diseño**

- Resultados de la aplicación de la métrica Relación entre clases.
- Resultados de la aplicación de la métrica Tamaño operacional de clase.

### **Implementación de la solución**

- Diagrama de componentes.
- Modelo de despliegue.
- Código fuente del componente.

### **Pruebas para validar la implementación**

- Resultados de las pruebas de caja blanca.
- Resultados de las pruebas de caja negra.

## **1.5 Tecnologías y herramientas para el desarrollo**

Cuando se inicia el desarrollo de un sistema se hace un estudio de las principales soluciones que existen y que van a darle solución a una determinada problemática. Muchos de los escenarios que pueden

encontrarse en el transcurso del desarrollo son resueltos por marcos de trabajo que facilitan obtener resultados en breves períodos de tiempo.

El desarrollo de la solución se realizará utilizando el marco de trabajo Sauxe, implementado por el Departamento de Tecnología del CEIGE. Está compuesto por un conjunto de componentes reutilizables que provee la estructura genérica, con soporte para entornos multientidad, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo, alejando a los programadores de los detalles arquitectónicos. (20) Cuenta con una arquitectura en capas como se muestra en la figura 2 reutilizando las siguientes tecnologías libres (21):



Figura 2. Arquitectura de Sauxe.

### 1.5.1 Marco de trabajo

**Zend\_Framework 1.7:** Se trata de un marco de trabajo para desarrollo de aplicaciones web y servicios web con PHP, brinda soluciones para construir sitios web modernos, robustos y seguros. Además es de código abierto y trabaja con PHP 5. Está formado por una serie de métodos estáticos y varios componentes que pueden usarse múltiples veces. (22)

**Doctrine 1.2.1:** Potente y completo sistema de mapeo relacional de objetos (por sus siglas en inglés ORM) para PHP 5.2+ con una capa de abstracción de bases de datos incorporado. Brinda la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos.



Al trabajar con este es necesario informar a su motor interno cuál es el modelo de la aplicación haciendo ingeniería inversa de la base de datos existente o si se empieza la aplicación desde cero, crear el modelo en la sintaxis específica que propone Doctrine y luego generar toda la base de datos. (23)

**Extjs 2.2:** es una biblioteca o conjunto de librerías de JavaScript para el desarrollo de aplicaciones web interactivas, usa tecnologías *AJAX*<sup>2</sup>, *DHTML*<sup>3</sup> y *DOM*<sup>4</sup>. Permite realizar completas interfaces de usuario, fáciles de usar, muy parecidas a las conocidas aplicaciones de escritorio, posibilitado a los desarrolladores concentrarse en la funcionalidad de las aplicaciones en vez de en las advertencias técnicas. (24)

### 1.5.2 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 (25).

**JavaScript:** Es un lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web. Gran parte de su programación está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas, entre otros. Permite la programación de pequeños scripts y de programas más grandes orientados a objetos, con funciones y estructuras de datos complejas (26). Además, pone a disposición del programador todos los elementos que forman la página web para poder acceder a ellos y modificarlos dinámicamente. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado, es soportado por Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros. (27)

**PHP 5.2.6:** Es un lenguaje interpretado de propósito general ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como

---

<sup>2</sup> Técnica de desarrollo web para crear aplicaciones interactivas.

<sup>3</sup> HTML Dinámico (del inglés Dynamic HTML).

<sup>4</sup> Interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.

salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. (28)

### 1.5.3 Herramientas para el modelado

**Visual Paradigm 6.4:** Es una herramienta de diseño UML<sup>5</sup> libre y profesional, diseñado para contribuir al desarrollo de software. Soporta los principales estándares como UML, SysML, BPMN y XML. Ofrece un completo conjunto de herramientas a los equipos de desarrollo de software, necesarios para la captura de requisitos, la planificación de software, la planificación de pruebas, el modelado de clases y el modelado de datos. (29)

### 1.5.4 Herramientas para el desarrollo colaborativo

**SVN (Subversion) 1.6.6:** Es un software de sistema de control de versiones, libre bajo una licencia de tipo Apache/BSD y se le conoce también como *svn* por ser ese el nombre de la herramienta de línea de comandos. Una característica importante de Subversion es que los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo. (30)

### 1.5.5 Gestor de Base de Datos

**PostgreSQL 8.3:** Es un servidor de base de datos relacional libre. Tiene soporte total para transacciones, disparadores, vistas, procedimientos almacenados, almacenamiento de objetos de gran tamaño. Se destaca en ejecutar consultas complejas, consultas sobre vistas, subconsultas y *joins* de gran tamaño. Permite la definición de tipos de datos personalizados e incluye un modelo de seguridad completo. Cuenta con una gran comunidad de desarrollo en Internet, su código fuente está disponible sin costo alguno y es multiplataforma. (22)

### 1.5.6 Servidor Web

**Apache 2.2.9:** Es un servidor gratuito de código fuente abierto, potente y que ofrece un servicio estable y sencillo de mantener y configurar. Se ejecuta en varios Sistemas Operativos, haciéndolo esta

---

<sup>5</sup> UML: Lenguaje de Modelado Unificado.

característica prácticamente universal. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es altamente configurable en la creación y gestión de *logs*. (31)

### 1.5.7 Navegador Web

**Mozilla Firefox 2.17:** Presenta una forma rápida y eficiente de navegar por la web, que permite abrir varias páginas en una misma ventana mediante el empleo de pestañas separadas. Contiene un Plugin Firebug que se utiliza para ver los errores del código. Firefox es un navegador multiplataforma y está disponible en varias versiones de Microsoft Windows, Mac OS X, GNU/Linux y algunos sistemas basados en Unix. Su código fuente es software libre, publicado bajo una triple licencia GPL/LGPL/MPL. (32)

## 1.6 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software, por lo que es de suma importancia aplicarlos en la construcción del diseño de un sistema. Con su uso, se pretende establecer un lenguaje común entre los programadores, contribuir a la reutilización, ahorrar tiempo en la implementación y obtener un producto con calidad.

Brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Se deben tener presente los siguientes elementos de un patrón: su nombre, el problema (cuándo aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios). (33)

El patrón es, resumiendo, al mismo tiempo una cosa que tiene su lugar en el mundo, y la regla que dice cómo crear esa cosa y cuándo crearla. (34)

### Patrones GoF

Los patrones *Gang Of Four* (GoF), son patrones de diseño publicados en el libro *Design Patterns: Elements of Reusable Object-Oriented Software* por Gamma, Helm, Jonson y Vlissides conocidos por Banda de los cuatro. Están divididos fundamentalmente en tres grandes grupos: estructurales, creacionales y de comportamiento. Como ventaja de la utilización del marco de trabajo Sauxe se aplican el Fachada, Cadena de responsabilidades e Instancia única.

- Fachada: Es un patrón estructural que define una interfaz de alto nivel que hace que el subsistema sea más fácil de usar, brindando una interfaz unificada simple para el acceso de una o varias interfaces de un subsistema.

- Cadena de responsabilidad: Es un patrón de comportamiento y se encarga de evitar el acoplamiento del remitente de una petición a su receptor, dando a más de un objeto la posibilidad de manejar la petición.
- Instancia única: Es un patrón creacional que garantiza que exista una instancia única para una clase y proporciona un punto de acceso global a ella.

### Patrones GRASP

Otros patrones de diseño son los GRASP (*General Responsibility Assignment Software Patterns*, Patrones Generales de Software para Asignación de Responsabilidades), estos describen los principios fundamentales de la asignación de responsabilidades a objetos, expresada en forma de patrones.

Existen nueve patrones GRASP los cuales son: Experto, Creador, Alta cohesión, Bajo acoplamiento, Controlador, Polimorfismo, Fabricación pura, Indirección y No hables con extraños.

## 1.7 Patrones de arquitectura

El patrón de arquitectura conocido como Modelo-Vista-Controlador (MVC), separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario; es decir separa en tres capas diferentes los datos de una aplicación, la interfaz de usuario, y la lógica de control:

- Modelo: Esta capa administra el comportamiento y los datos del dominio de la aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y a instrucciones de cambiar el estado (habitualmente desde el controlador).
- Vista: Esta capa maneja la visualización de la información, es decir que presenta el modelo en un formato adecuado para interactuar, que usualmente es la interfaz de usuario.
- Controlador: Esta capa controla el flujo de datos entre la vista y el modelo; es decir que responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. (35)

Esta separación permite construir y probar el modelo, independientemente de la representación visual.

## 1.8 Conclusiones parciales

En este capítulo con el fin de transmitir las especificidades del negocio se parte de los conceptos y definiciones asociados al proceso. De los sistemas estudiados que incluyen dicho proceso, se recoge una breve descripción que resume el por qué ninguno de estos cumple con los requisitos necesarios para la

gestión de las nóminas de los trabajadores en Cuba, demostrando que es mejor desarrollar un componente propio que responda a las peculiaridades de la economía cubana. La descripción de las tecnologías, el modelo de desarrollo y las herramientas con las cuales se va a desarrollar la solución contribuyen a un mejor entendimiento de esta, mostrando además que su utilización fortalece el paradigma de independencia tecnológica por el cual apuesta el país. Todos los elementos mencionados con anterioridad conforman la fundamentación teórica que sustenta el desarrollo del presente trabajo.

## **CAPÍTULO 2: PROPUESTA DE SOLUCIÓN**

### **2.1. Introducción**

En el presente capítulo se realiza una valoración del diseño anterior del sistema, a partir de la cual se propone el nuevo diseño que da paso a la implementación. Se obtienen los artefactos: diagramas de clases del diseño a los cuales se les aplican patrones que permiten precisar un vocabulario común entre diseñadores, hacer el trabajo más eficiente y entendible para los desarrolladores; el diagrama de componentes, que ofrece una visión más clara de la dependencia que tiene el componente de otros, facilitando la implementación del sistema, y el modelo de datos donde se representan las tablas vinculadas a la propuesta de solución, así como las relaciones existentes entre ellas.

### **2.2. Valoración del diseño anterior**

Para la realización del presente trabajo existían los artefactos: Descripción de los Requisitos Funcionales, Diagrama entidad relación y Diagramas de clases del diseño previamente elaborados. Teniendo en cuenta la importancia del diseño como punto de partida para la implementación, se decide realizar una valoración a la versión existente. En una entrevista con los analistas y arquitectos se concluyó que los requisitos habían cambiado por lo que el diseño anterior no se ajustaba a las nuevas especificaciones causando ambigüedades en los diagramas, malas interpretaciones y falta de información para los desarrolladores. Además, como no reflejaba con claridad la relación existente entre las clases, la probabilidad de conocer con exactitud la repercusión en la modificación de una de ellas para el resto era baja.

De forma general los diagramas de clases del diseño anterior no reflejaban el uso correcto de estereotipos web, carecían de relaciones entre clases esenciales para una correcta comprensión y desarrollo de la solución, así como funcionalidades que estas debían tener.

En el caso del diagrama Administrar nómina se pudo apreciar que no reflejaba las funcionalidades necesarias para llevar a cabo el procesamiento de la nómina, la funcionalidad Utilizar multimoneda fue excluida provocando su eliminación de las operaciones relacionadas con ella en las clases, se realizaron cambios en los requisitos de la funcionalidad Procesar nóminas, derivados de los cambios en el requisito Contabilizar nóminas que no se reflejan en el diseño, por lo que sería necesario adicionar las nuevas relaciones con otras clases que intervienen en el escenario (ejemplo: valores que se deben procesar y guardar para ser utilizados en la actualización de los submayores luego de contabilizar). Se agregaron

nuevas funcionalidades imprescindibles para el escenario de gestión de la nómina en general que el diagrama de clases no reflejaba.

Para el diagrama Tipo de nómina se definió como estático el nomenclador de funciones de la nómina y su prioridad, provocando cambios en funcionalidades de las clases que gestionan el negocio. Fue definida una nueva relación de los tipos de nómina con los conceptos de pagos, generando nuevas tablas y nuevas clases con más funcionalidades, provocando cambios además en las clases que gestionan el negocio ahora con nuevas relaciones.

El Revisar nómina eliminó la funcionalidad ajustar nómina, que implicó cambios en las clases del diseño del escenario con igual nombre, así como la funcionalidad gestionar tipo de ajuste, desechando el diagrama de clases del diseño de este escenario y provocando cambios en las clases que gestionan el negocio del revisar nómina. Los cambios en los requisitos de la funcionalidad Contabilizar nóminas no se reflejaban en el diseño, por lo que se hizo necesario adicionar las nuevas relaciones con otras clases que intervienen en el escenario. También los cambios en los requisitos de Actualizar submayores implicaron nuevas funcionalidades y relaciones con nuevas clases que antes no se encontraban.

Al detectarse errores en dichos artefactos y por la importancia que representa poseer un diseño actualizado, se propone la realización del mismo partiendo de los nuevos requerimientos entregados. La lista de estos se puede observar en el anexo 2.

## **2.3. Diseño del sistema**

El diseño es un modelo del sistema o producto que se va a construir. Debe ser suficiente para que la implementación de dicho sistema se realice sin ambigüedades, encontrando la forma para que soporte todos los requisitos y restricciones que se le suponen. Una entrada esencial en el diseño es el resultado del análisis que proporciona una descripción detallada de los requisitos. (36)

### **2.3.1. Patrones de diseño utilizados**

Para contribuir a que el sistema sea más robusto y flexible se utilizaron en la elaboración del diseño los siguientes patrones GRASP:

- Experto: Se pone en práctica con el uso de clases que poseen responsabilidades específicas a cumplir, de acuerdo con la información que manejan. El componente cuenta con clases controladoras, modelos y de entidad que poseen funciones concretas de acuerdo con la información que gestionan.
- Bajo acoplamiento: Consiste en tener las clases lo menos relacionadas entre sí, para que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión en las demás, potenciando la reutilización y disminuyendo la dependencia entre las clases.
- Alta cohesión: Este patrón determina, que la información almacenada en una clase debe ser coherente y estar relacionada con esta en mayor medida, enfocada en sus responsabilidades. Realizando un diseño donde las clases mantengan una alta cohesión, es posible ganar en claridad y facilidad a la hora de entender el diseño, además de simplificar el mantenimiento y soportar mayor capacidad de reutilización.
- Creador: El patrón de creación hizo posible que el diseño pudiera soportar bajo acoplamiento, encapsulación y reutilización, así como mayor claridad. Las clases controladoras son responsables de crear el objeto de las modelos y estas a su vez de las entidades.

### 2.3.2. Patrones de arquitectura utilizados

Los diagramas de clases realizados para el diseño aplican el patrón arquitectónico modelo vista controlador (MVC) el cual separa en tres capas diferentes la interfaz, la lógica de negocio y los datos. Las páginas *.html* y las clases *.js* representan la vista como se muestra en la figura, las clases *controller* el controlador y en el modelo se representan las clases de la lógica del negocio y las de dominio que son las de acceso a datos; logrando que cualquier cambio que se realice en la vista no afecte ni la lógica del negocio, ni el dominio. La figura 3 muestra como se ve la aplicación de este patrón en todos los diagramas de clases de forma general por lo que no se especifican nombres de métodos y clases concretos.



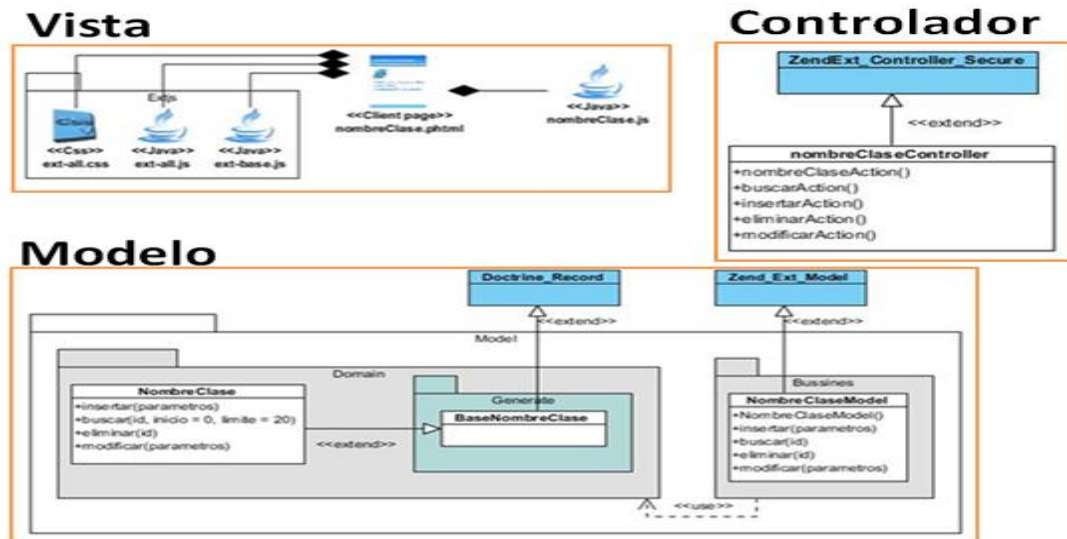


Figura 3. Representación general de patrón MVC en los diagramas de clases.

### 2.3.3. Diagrama de clases

Un diagrama de clases del diseño es un diagrama estático que describe la estructura de un sistema mostrando sus clases, operaciones y las relaciones existentes entre ellas.

En la figura 4 se muestra el diagrama de clases del diseño del escenario Administrar nómina, donde se representan las principales clases, operaciones y relaciones que se necesitan para darle cumplimiento a los requisitos funcionales relacionados con la gestión de los datos de la nómina hasta su procesamiento. Las clases *AdministrarNomina.js* y *AdministrarNomina.phtml* conforman la capa arquitectónica de presentación. La clase *AdministrarNominasController* solo maneja la comunicación entre la vista y el modelo, las clases *Model* son las encargadas de la lógica del negocio, implementando funcionalidades que garantizan el cumplimiento de los requisitos identificados; encargadas del acceso a los datos se encuentran *DatNprocNomina*, *DatNnomina*, *DatNprocConceptoPago*, *DatNprocNominalImpuesto*, *DatNprocRetenciones* y todas las clases Base de las cuales extienden las mencionadas.

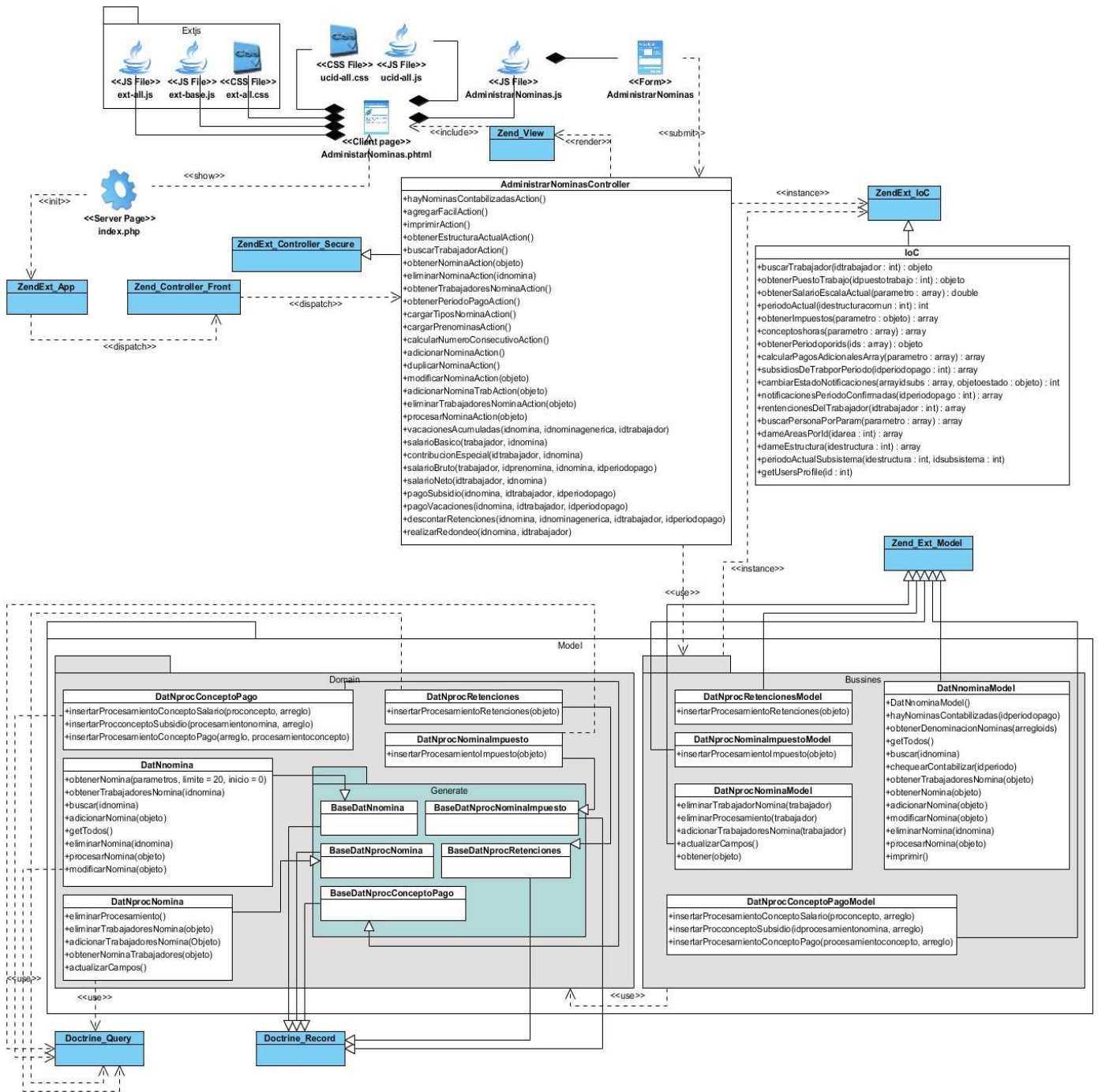


Figura 4. Diagrama de clases del diseño Administrar nómina.

En la figura 5 se muestra el diagrama de clases del diseño del escenario Gestionar tipo nómina, en la cual se le da solución a los requisitos funcionales relacionados con la gestión de los tipos de nómina, se representan las principales clases, operaciones y relaciones que se necesitan para darle cumplimiento a estos requerimientos, donde la capa arquitectónica de presentación está formada por las clases *GestionarTipoNomina.js* y *GestionarTipoNomina.phtml*. La comunicación entre la vista y el modelo es realizada por la clase *GestionarTipoNominaController*, las clases *Model* son las encargadas de la lógica del negocio, implementando funcionalidades que garantizan el cumplimiento de los requisitos identificados, encargadas del acceso a los datos se encuentran *NomTipoNominaConcepto*, *NomTipoNomina*, *DatTipoNominaFuncion* así como todas las clases Base de las cuales extienden.

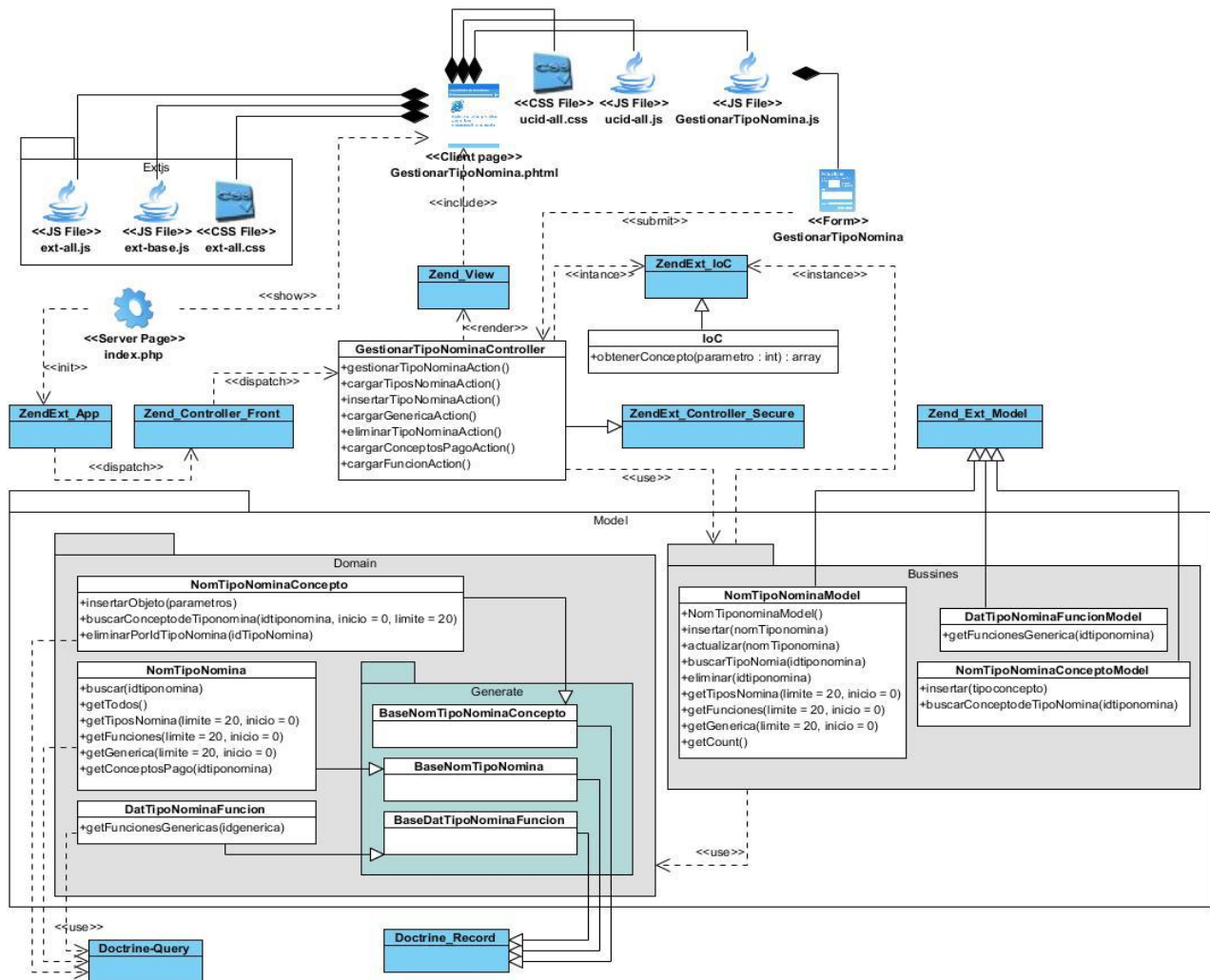


Figura 5. Diagrama de clases del diseño Gestionar tipo nómina.

En la figura 6 se muestra el diagrama de clases del diseño del escenario Revisar nómina, en la cual se le da solución a los requisitos funcionales relacionados con la revisión de la nómina, donde luego de comprobar que estén correctas, se contabilizan, se confirman y se emite un comprobante. Las clases *RevisarNomina.js* y *RevisarNomina.phtml* conforman la capa arquitectónica de presentación. La comunicación entre la vista y el modelo es realizada por la clase *RevisarNominasController*, las encargadas de la lógica del negocio son las clases *Model* y encargadas del acceso a los datos se encuentran *DatNprocNomina*, *DatNnomina* y *DatNprocRetenciones* así como todas las clases Base de las cuales extienden.

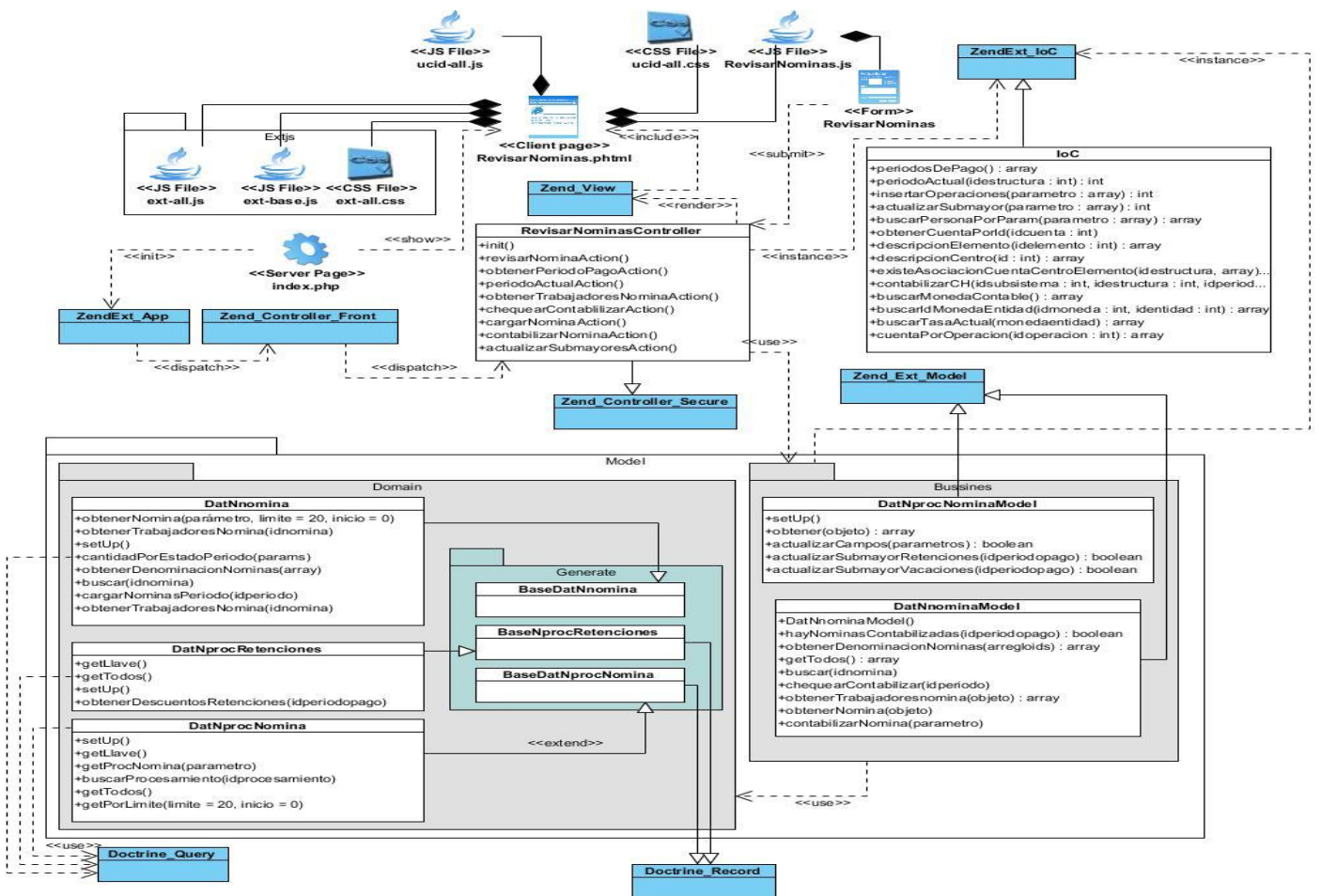


Figura 6. Diagrama de clases del diseño Revisar nómina.

La descripción de los métodos del componente mostrados en los diagramas de clases puede verse en el anexo 1.

#### **2.3.4. Modelo de datos**

Los modelos de datos (MD), son el mecanismo formal para representar los datos de manera general y sistemática, establece una relación entre el mundo real y la información almacenada físicamente en la base de datos (BD). (37)

El objetivo de construir un MD es identificar y representar las tablas (entidades) de importancia para el funcionamiento del negocio, sus propiedades (atributos), y la forma en que estas tablas se comunican entre sí (relaciones). Este modelo se desarrolla para facilitar el diseño de la BD y mostrar los datos que contendrá el sistema.

El modelo de datos propuesto en la solución cuenta con un total de 12 tablas, de ellas 4 son nomencladores encargados de gestionar conceptos específicos del negocio ya predefinidos, ejemplo: la tabla “nom\_tiponmina” gestiona los tipos de nóminas definidos en la entidad y su vínculo a la nómina genérica a la que pertenece (salario, subsidio y vacaciones).

Las ocho tablas restantes se encargan de gestionar los datos que son necesarios tener registrados, por ejemplo, la tabla “dat\_nnmina” almacena los principales datos relacionados con las nóminas como son la denominación que la identifica y el estado en que se encuentra. El estado de la nómina determina las acciones que se pueden realizar sobre esta hasta su contabilización. Otro ejemplo es la tabla “dat\_nprocnomina” encargada de registrar los datos de la asociación de la nómina con sus trabajadores en el período de pago y los principales valores resultantes del procesamiento de la nómina, como el salario a pagar al trabajador (necesarios para la emisión de los comprobantes de operaciones). En “dat\_tiponminafuncion” es donde persiste la relación de los tipos de nóminas con las funciones, estas últimas tienen asociadas una prioridad, necesaria para realizar el procesamiento de la nómina.

De las tablas que se muestran en el modelo de datos, hay tres que no pertenecen directamente al componente, pero son imprescindibles para llevar a cabo el proceso que gestiona la nómina, por ejemplo, a partir de los datos que persisten en la tabla “dat\_prenomina” se crea la nómina de tipo salario, teniendo en cuenta solamente las pre nóminas que se encuentren en estado confirmada. Otro ejemplo lo constituye la tabla “nom\_periodopago” donde se almacenan los períodos en el cual se enmarcan las nóminas y por último la tabla “dat\_trabajador” donde se registran los datos de los trabajadores.

A continuación se muestra el modelo de datos obtenido:

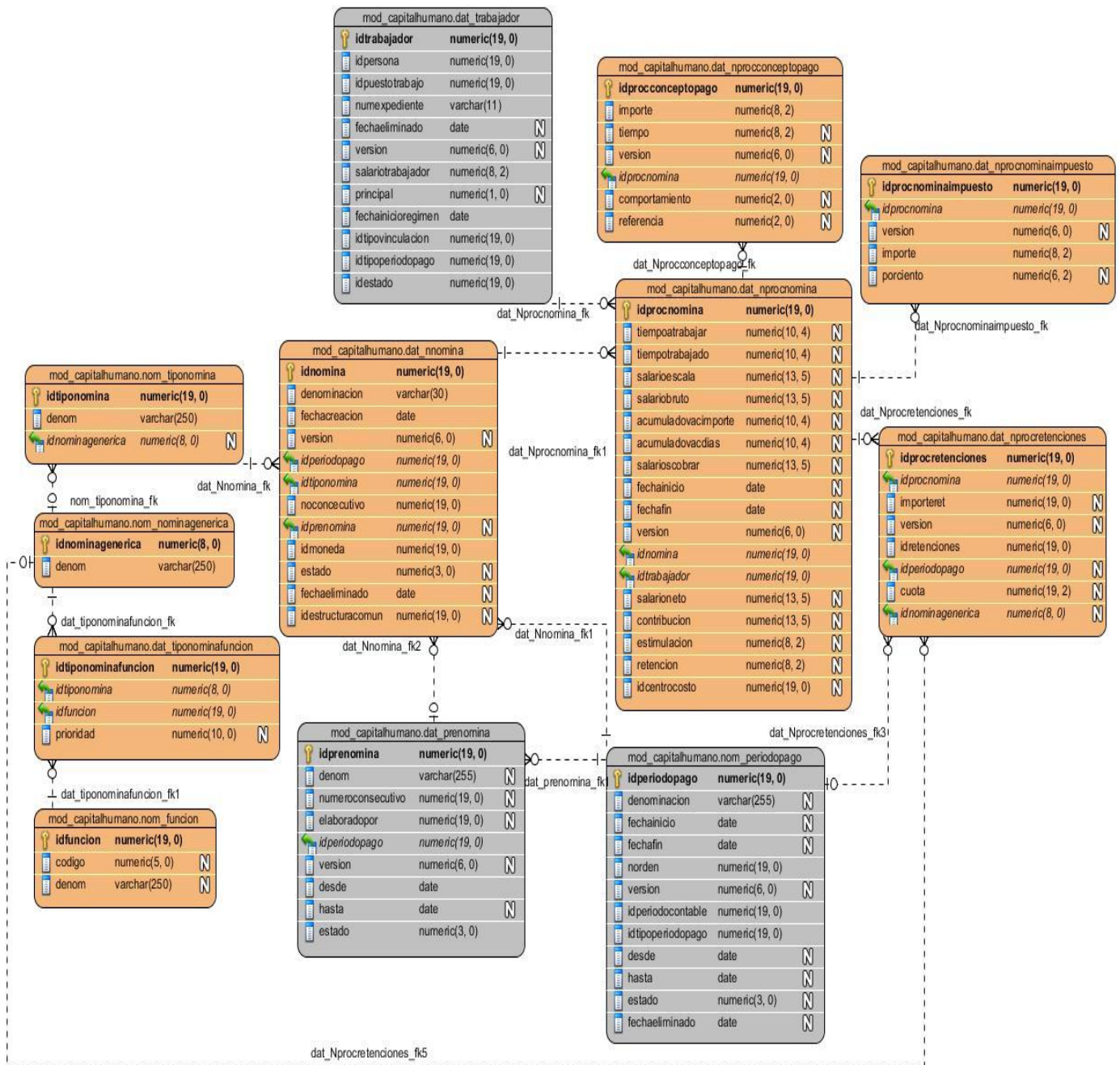


Figura 7. Modelo de datos del componente Nómina.

## 2.4. Implementación

### 2.4.1. Estructura del componente Nómina.

Dentro de la carpeta raíz del sistema Cedrux se encuentran definidas las carpetas *apps* y *web*, que contienen la lógica de negocio y las vistas de los componentes de cada subsistema respectivamente. Las carpetas contenidas en Capital humano llevan igual nombre que el proceso que representan, nómina se encuentra contenida en la carpeta Estimulación moral y material como se observa en la figura 8.

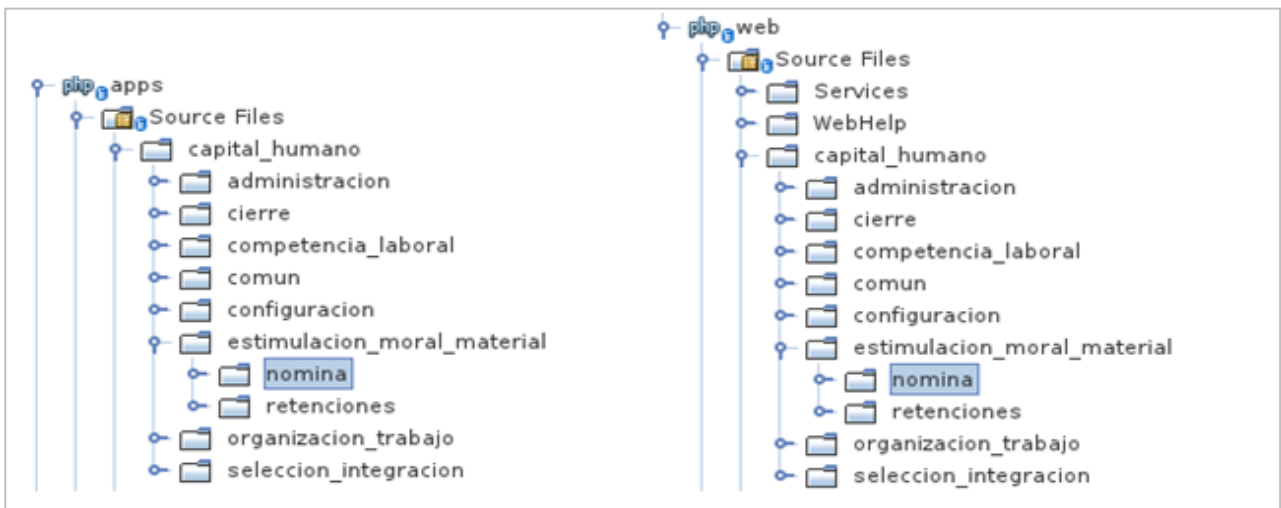


Figura 8. Contenido de las carpetas apps y web para Capital humano.

La carpeta *apps* contiene, al mismo nivel de las carpetas que llevan el nombre de los procesos que identifican, la carpeta *común* que incluye el fichero de tipo xml, nombrado ioc.

#### Nómina en la carpeta apps

El componente Nómina, muestra una estructura como aparece en la figura 9, compartida por el resto de los componentes del subsistema.

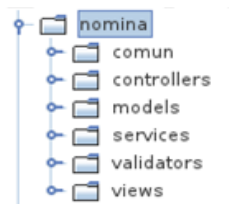


Figura 9. Estructura de la carpeta Nómina.

**controllers:** Almacenan las clases controladoras encargadas de la comunicación entre las interfaces de usuario y la lógica del negocio de la solución.

**models:** En esta carpeta se encuentran las carpetas *bussines* y *domain* encargadas de la lógica del negocio y acceso a datos del componente respectivamente.

**services:** Esta carpeta contiene la(s) clase(s) que se utiliza(n) para interactuar con los servicios que brinda el componente, permitiendo el acceso a las funcionalidades desde otros componentes haciendo uso del fichero *ioc.xml* (ver acápite 2.4.4).

**validators:** Esta carpeta posee las clases de tipo *.php* que van a realizar acciones de validación en el componente, como las precondiciones que se deben cumplir antes de que un determinado método sea ejecutado.

**views:** En esta carpeta se recopilan los ficheros que van a gestionar la capa de presentación, estos ficheros se agrupan en dos carpetas:

**idioma:** Contiene ficheros de tipo *json* que recopilan etiquetas para la gestión de los mensajes vinculados a la presentación.

**scripts:** En este directorio se incluyen todas las vistas, para ello se crea una carpeta para cada clase controladora y dentro se incluye la vista o *script*, archivos de extensión *phtml* donde se especifica el título de la página que se gestiona y se carga el archivo *js* que mostrará la presentación.

### Nómina en la carpeta web

La figura 10 muestra la estructura de la carpeta web perteneciente al componente Nómina.



Figura 10. Estructura de la carpeta Nómina.

**css:** En este directorio se encuentran las plantillas y estilos para el diseño del componente.

**js:** Es la carpeta donde se incluyen las clases *javascript*, ficheros con la extensión *.js* donde se encuentra el código correspondiente a la capa de presentación (interfaces de usuario).



### **2.4.2. Descripción del componente**

El componente Nómina es el encargado en el subsistema Capital humano de la gestión (incluye crear, eliminar, modificar, revisar y procesar) de las diferentes nóminas, para efectuar el pago a los trabajadores dentro del período definido en cada empresa, realizar el descuento de retenciones contraídas por los trabajadores, el cálculo de la provisión de vacaciones y la emisión del comprobante de operaciones contables realizadas.

### **2.4.3. Diagrama de componentes**

Un diagrama de componentes muestra la organización y las dependencias entre un conjunto de componentes. Puede ser un tipo especial de diagrama de clases que se centra en los elementos físicos del sistema y sus relaciones (38). Proveen una vista arquitectónica de alto nivel del sistema y ayuda a los desarrolladores a visualizar el camino y tomar decisiones respecto a las tareas de implementación (39). En la figura 11 se muestra el diagrama de componentes elaborado desde la perspectiva del componente Nómina para el subsistema Capital Humano. Dentro del subsistema recibe u ofrece información a otros componentes como son:

- Vacaciones
- Trabajador
- Subsidios
- Retenciones
- Puesto de trabajo
- Período de pago
- Pagos Adicionales
- Incidencias
- Impuestos y contribuciones
- Concepto de pago
- Cierre

En la figura 12 se muestra el diagrama de componentes externos con los cuales interactúa el componente Nómina como son:

- Comprobante tipo.
- Configuración.
- Metadatos.
- Multimoneda.
- Nomenclador de Cuentas.
- Nomencladores y Configuración.
- Seguridad.

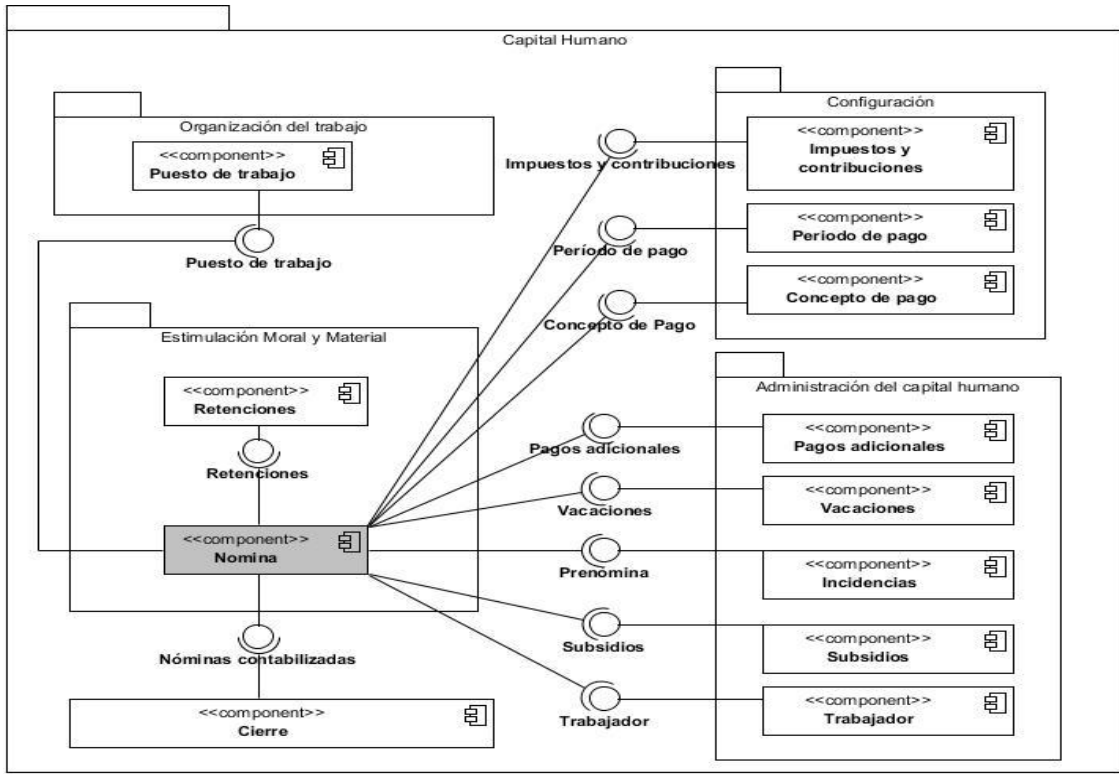


Figura 11. Diagrama de componentes internos.

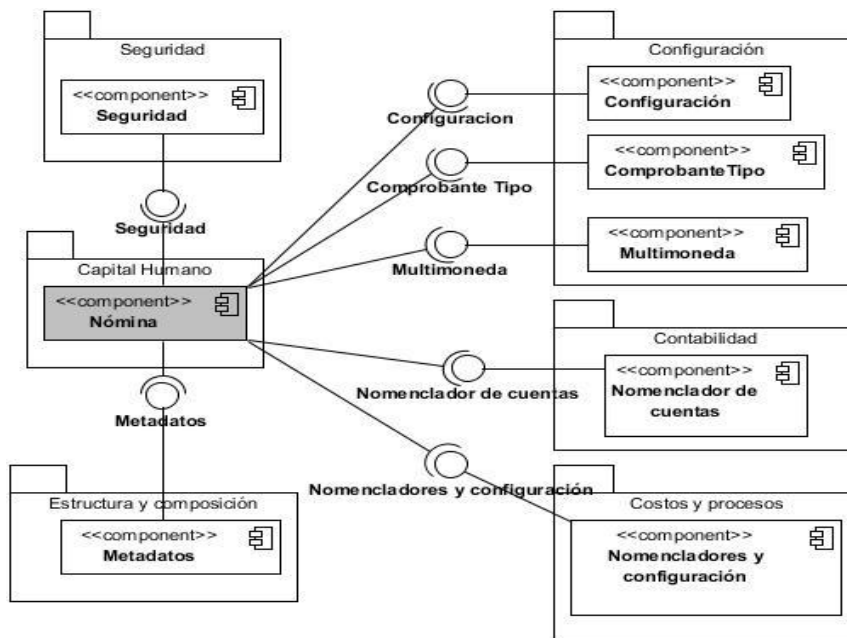


Figura 12. Diagrama de componentes externos.

#### 2.4.4. Servicios utilizados

Todo el código dentro de un mismo componente utiliza llamadas a métodos o eventos de forma directa, sin embargo, para lograr comunicación entre diferentes módulos, se utiliza el patrón inversión de control (IoC)<sup>6</sup>. La utilización de este patrón se evidencia en el fichero .xml con igual nombre (ioc.xml) encargado de registrar las funcionalidades que ofrecen los componentes del sistema para su utilización desde otros componentes. Existe un fichero ioc.xml interno contenido en la carpeta *apps* de Capital humano para la integración entre componentes de dicho subsistema, en la figura 13 se muestra un fragmento de este correspondiente al servicio *NominasTodasContabilizadas* del componente Nómina. Para la integración entre subsistemas se utiliza el ioc.xml externo, la figura 14 muestra un fragmento de este correspondiente al servicio *NominasContabilizadas* del subsistema Capital Humano.

```
<?xml version="1.0" encoding="utf-8"?>
<ioc>
  <nomina src="capital_humano/estimulacion_moral_material/nomina">
    <NominasTodasContabilizadas>
      <inyector clase="AdministrarNominaService" metodo="NominasTodasContabilizadas"/>
      <prototipo>
        <parametro nombre="params" tipo="stdClass"/>
        <resultado tipo="enterospos"/>
      </prototipo>
    </NominasTodasContabilizadas>
  </nomina>
</ioc>
```

Figura 13. Contenido del fichero ioc.xml interno.

<sup>6</sup> IoC: es un concepto junto a unas técnicas de programación en las que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales, en los que la interacción se expresa de forma imperativa haciendo llamadas a procedimientos o funciones. Tradicionalmente el programador especifica la secuencia de decisiones y procedimientos que pueden darse durante el ciclo de vida de un programa mediante llamadas a funciones. En su lugar, en la inversión de control se especifican respuestas deseadas a sucesos o solicitudes de datos concretas, dejando que algún tipo de entidad o arquitectura externa lleve a cabo las acciones de control que se requieran en el orden necesario y para el conjunto de sucesos que tengan que ocurrir.

```
<?xml version="1.0" encoding="utf-8"?>
<ioc>
  <capitalhumano src="capital_humano">
    <NominasContabilizadas reference = "remuneracion_nomina/nomina">
      <inyector clase = "NominaService" metodo = "NominasContabilizadas" />
      <prototipo>
        <resultado tipo="bool"/>
      </prototipo>
    </NominasContabilizadas>
  </capitalhumano>
</ioc>
```

Figura 14. Contenido del fichero ioc.xml externo.

#### 2.4.4.1. Servicios que consume nómina

- subsidiosDeTraborPeriodo: Se obtienen las notificaciones de subsidio confirmadas en el período de pago especificado.
- notificacionesPeriodoConfirmadas: Se obtienen las notificaciones de vacaciones confirmadas en el período de pago correspondiente a la nómina.
- buscarTrabajador: Se obtiene el trabajador a partir del parámetro enviado.
- buscarPersonaPorParam: Se obtienen los datos de la persona a partir del parámetro enviado.
- obtenerPuestoTrabajo: Se obtiene el puesto de trabajo ocupado por el trabajador.
- periodosDePago: Se obtienen todos los periodos de pago abiertos.
- periodoActual: Se obtiene el período de pago actual en que se encuentra el subsistema.
- obtenerprenominasParams: Se obtienen las pre-nóminas del período de pago seleccionado y que están confirmadas.
- obtenerPrenominasNoEliminadas: Se obtienen todas las pre-nóminas que no han sido eliminadas.
- buscarTrabPrenomina: Se obtienen los trabajadores asociados a una pre-nómina.
- obtenerSalarioEscalaActual: Se obtiene el salario escala del trabajador seleccionado en dependencia del cargo que ocupa.
- buscarTrabPrenomina: Retorna los trabajadores de la pre nómina.
- obtenerImpuestos: Retorna los impuestos definidos en la entidad.
- conceptoshoras: Devuelve los datos de las horas de las incidencias por las que está registrado el trabajador en la pre-nómina.

- obtenerPeriodoporids: Obtiene los datos del período de pago a partir del identificador especificado.
- calcularPagosAdicionalesArray: Se obtiene el valor de los pagos adicionales del trabajador.
- subsidiosDeTrabporPeriodo: Se obtienen los subsidios del trabajador en el período seleccionado.
- rentencionesDelTrabajador: Retorna todas las retenciones activas asociadas al trabajador.
- cambiarEstadoNotificaciones: Se cambian las notificaciones de los subsidios que se encuentran confirmadas a pagadas.
- periodoActualSubsistema: Se obtiene el período actual en el cual se encuentra el subsistema.
- cuentaPorOperacion: Se obtiene la cuenta asociada a la operación en reglas contables.
- buscarIdMonedaEntidad: Se obtiene el identificador de la moneda para una entidad.
- buscarTasaActual: Se obtiene la tasa de cambio para una determinada moneda.
- buscarMonedaContable: Se obtiene la moneda contable definida en el subsistema.
- dameEstructurasInternas: Se obtiene todas las áreas de una estructura.
- dameAreasPorId: Se obtienen datos específicos de un área.
- existeAsociacionCuentaCentroElemento: Se verifica que exista la asociación elemento-centro-cuenta.
- descripcionElemento: Se obtienen datos que describen e identifican el elemento de gasto.
- obtenerCuentaPorId: Se obtienen datos que describen e identifican la cuenta contable.
- getUsersProfile: Se obtienen todos los datos del usuario de la sesión.
- contabilizarCH: Recibe el comprobante de operaciones de Capital humano para asentarlo en Contabilidad.
- actualizarSubmayor: Encargado de actualizar los datos al submayor de retenciones.
- insertarOperaciones: Encargado de actualizar los datos al submayor de vacaciones.
- obtenerConcepto: Se obtienen los conceptos de pago asociados a la nómina.

#### **2.4.4.2. Servicios que brinda nómina**

- nominasTodasContabilizadas: Retorna un número que sirve para determinar si están todas las nóminas del período especificado contabilizadas.

### **2.4.5. Diagrama de despliegue**

El diagrama de despliegue es utilizado para modelar el hardware empleado en las implementaciones de sistemas y las relaciones entre sus componentes. Constituye un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. El despliegue del componente Nómina se rige por la especificación dada para el subsistema Capital humano. A continuación se muestran los posibles escenarios con los requerimientos de software.

#### **Servidores**

##### **Servidor de Aplicaciones Web**

- Sistema Operativo: Ubuntu Server
- Servidor Web: Apache 2.0
- Librerías Adicionales: PHP 5

##### **Servidor de Base de Datos**

- Sistema Operativo: Ubuntu Server
- Sistema Gestor de Base de Datos: PostgreSQL 8.3.8

##### **Servidor de Clientes Ligeros**

- Sistema Operativo: Nova Server
- Navegador Web: Mozilla Firefox 2.17 ó superior
- Herramientas Ofimáticas
- Visualizador de ficheros pdf
- Herramienta de administración de clientes ligeros

#### **Clientes**

##### **PC Cliente con disco duro**

- Sistema Operativo: Linux o Windows
- Navegador Web: Mozilla Firefox v2.2 ó superior
- Herramientas Ofimáticas
- Visualizador de ficheros pdf

##### **PC Cliente sin disco duro**

- Todo se instala en el servidor de clientes ligeros

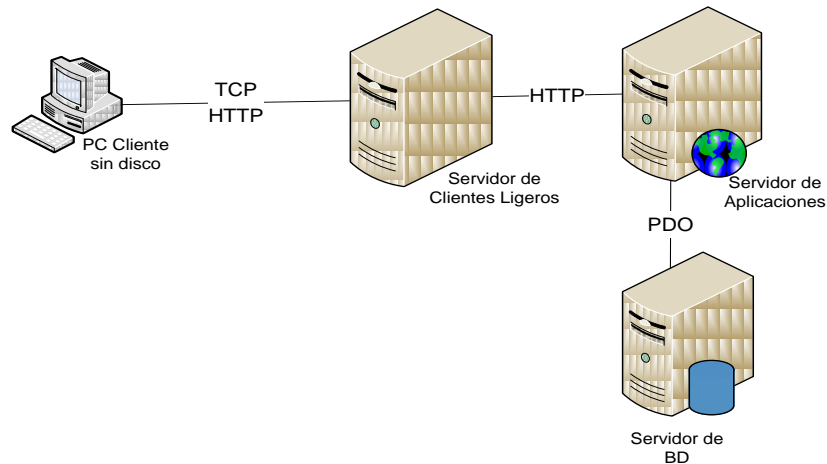


Figura 15. Diagrama de despliegue de escenario para PC cliente sin disco.

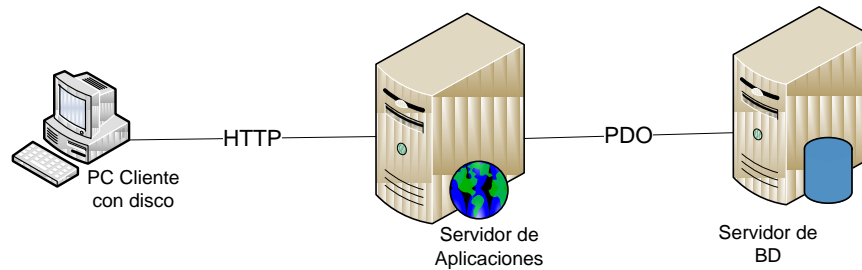


Figura 16. Diagrama de despliegue de escenario para PC cliente con disco.

#### 2.4.6. Funcionalidades implementadas

En la figura 17 se muestra la interfaz administrar nómina que ofrece las opciones de Adicionar, Modificar, Duplicar, Procesar, Eliminar e Imprimir una nómina, además se pueden filtrar por período de pago. También permite adicionar trabajadores a la nómina y eliminarlos, todo esto luego de que la nómina ha sido seleccionada y se encuentra en estado “En edición”.

Existen tres tipos de nóminas genéricas: Salario, Vacaciones y Subsidio, en dependencia del tipo seleccionado se muestran u ocultan las columnas que aparecen en la parte inferior de la interfaz.

Para todas las nóminas estarán siempre visibles los campos: Expediente interno, Nombre, Primer apellido, Segundo apellido, Contribución especial, Retenciones y Salario a cobrar, la visibilidad del resto de los campos se muestran en la figura 18, representando con 1 su presencia y con 0 la ausencia.

Figura 17. Interfaz administrar nómina.

Campo/ Tipo de nómina	Salario Escala	Horas Trabajadas	Pagos Adicionales	Salario Bruto	Días	Importe	Días Acumulados	Importe Acumulado
Salario	1	1	1	1	0	0	1	1
Vacaciones	0	0	0	0	1	1	0	0
Subsidio	0	0	0	0	1	1	0	1

Figura 18. Visibilidad de los campos según el tipo de nómina.



En la figura 19 se muestra la interfaz revisar nómina la cual ofrece las opciones de Confirmar, Contabilizar y Cancelar las nóminas, además muestra el período actual con el que se está trabajando y permite filtrar por período de pago. En la parte inferior muestra los trabajadores asociados a la nómina seleccionada y facilita la búsqueda de estos mediante los campos expediente interno, nombre, primer apellido y segundo apellido. Mediante esta interfaz se verifica que las nóminas procesadas estén elaboradas correctamente, se confirman y se procede a contabilizarlas, en caso contrario se cancelan cambiando su estado a “En Edición”, lo que le permite al escenario administrar nómina procesarla nuevamente.

Inicio Revisar n...

Revisar nóminas

Confirmar Cancelar Contabilizar Período actual : Primero Período de pago Primero

Denominación	Tipo de nómina	Período de pago	Estado	Fecha
Reintegros Abril 2012	Salario	Primero	Procesada	2012-05-11
Abril 2012	Salario	Primero	Procesada	2012-05-11
Tamara	Salario	Primero	Confirmada	2012-03-06
Tami	Tamara	Primero	Confirmada	2012-03-08

Página 1 de 1 Mostrando 1 - 4 de 4

Exp. Interno Nombre 1er Apellido 2do Apellido

Exp. Interno	Nombre	1er Apellido	2do Apellido	Salario escala	Horas trabajadas	Pagos adicionales	Salario bruto	Contribución especial	Retenciones	Días	Importe
27	Tamara	Rodríguez	Sánchez	475.00000	180.0000	100.00	575.00000	28.75000	100.00	16...	52.2675

Página 1 de 1 Mostrando 1 - 1 de 1

Figura 19. Interfaz revisar nómina.

En la figura 20 se muestra la interfaz gestionar tipo de nómina, que brinda las opciones de Adicionar, Modificar, Eliminar e Imprimir un tipo de nómina. Cada tipo de nómina tiene asociadas un grupo de funciones que siguen un determinado orden para su aplicación durante el procesamiento. Esta asociación y orden específico está dado por el tipo de nómina genérica seleccionado (Salario, Subsidio, Vacaciones) al momento de ser creado. Los tipos de nómina tienen asociados además conceptos de pago, que le permiten a la funcionalidad contabilizar tener los datos de las cuentas que se afectan y que se detallan en el comprobante de operaciones.

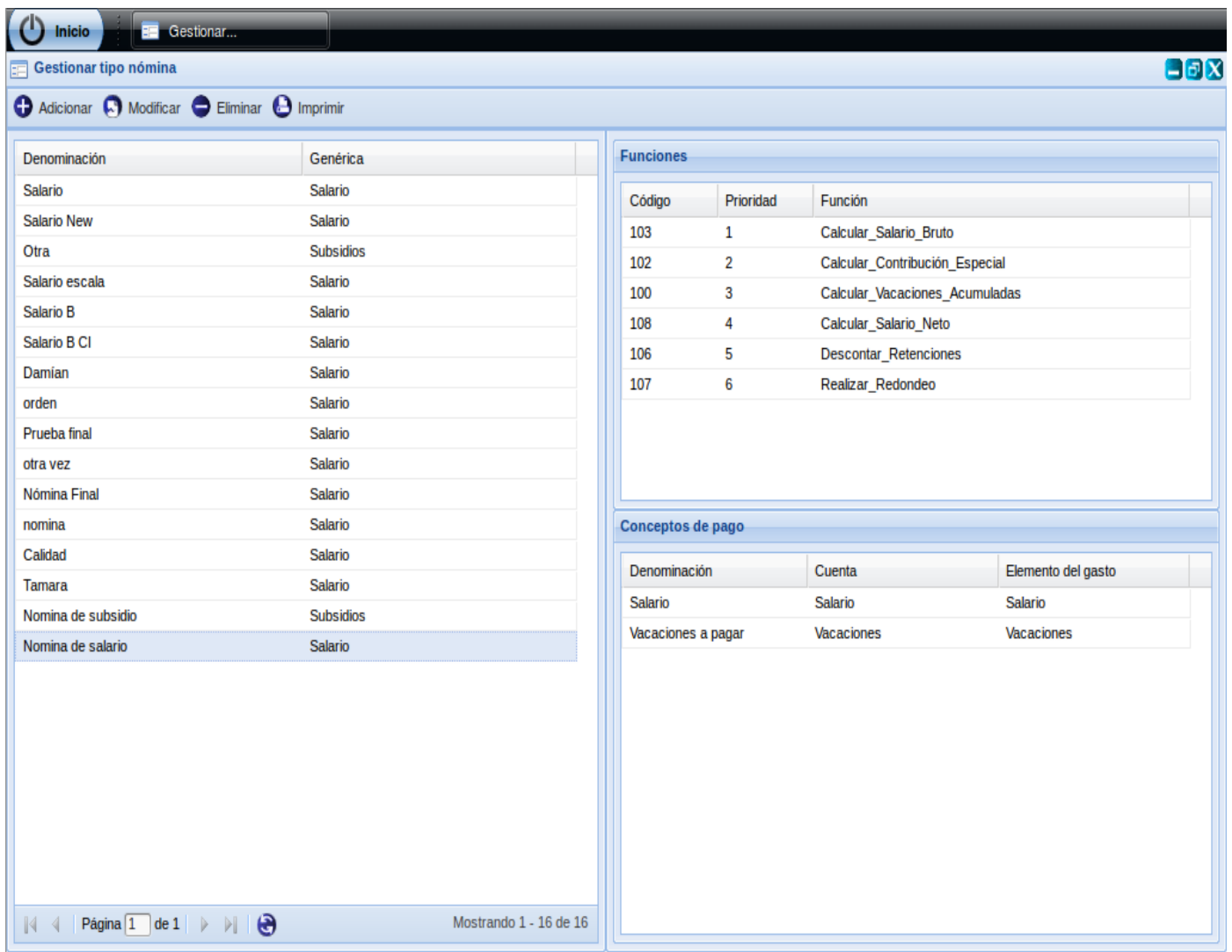


Figura 20. Interfaz gestionar tipo de nómina.

## **2.5. Conclusiones parciales**

En el capítulo se realizó la modelación del diseño, el cual permitió identificar las principales clases de la solución, las relaciones existentes entre ellas y sus métodos asociados. Además se elaboró el diagrama de componentes, que muestra la relación del componente Nómina con el resto del subsistema y los servicios que este utiliza. La persistencia de los datos en la base de datos pudo observarse a través del modelo de datos físico, encargado de representar las tablas, sus propiedades y las relaciones entre ellas. Se mostraron los posibles escenarios en los cuales puede ser desplegado el sistema a través del diagrama de despliegue, así como los requerimientos de software necesarios para un correcto funcionamiento de este. Se realizó la implementación del componente Nómina que permitió contar con la solución, la cual se encuentra lista para ser validada.

## CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN

### 3.1 Introducción

En el presente capítulo se realiza la validación del diseño a través de métricas, en aras de verificar el cumplimiento de los atributos de calidad definidos por las propias métricas seleccionadas. Se describen las pruebas efectuadas al software que tienen como objetivo detectar y corregir el máximo de errores en el sistema, antes de su entrega al cliente.

### 3.2 Validación del modelo de diseño propuesto

Con el objetivo de comprobar cuán bien están definidas las clases, se emplean las métricas Tamaño operacional de la clase y Relaciones entre clases, diseñadas para evaluar los siguientes atributos de calidad:

- **Responsabilidad:** Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto de la problemática propuesta.
- **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 costes 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.

#### 3.2.1 Métrica Tamaño Operacional de Clase

**Tamaño operacional de clase (TOC):** Está dado por el número de métodos asignados a una clase.

A continuación se muestra una serie de tablas encaminadas a un mejor entendimiento de la utilización de esta métrica.

**Tabla 2. Métrica Tamaño Operacional de Clase.**

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.

**Tabla 3. Rango de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica TOC.**

Atributo	Categoría	Criterio
Responsabilidad	Baja	$\leq$ Promedio
	Media	Entre Promedio y $2^*$ Promedio
	Alta	$> 2^*$ Promedio
Complejidad de Implementación	Baja	$\leq$ Promedio
	Media	Entre Promedio y $2^*$ Promedio
	Alta	$> 2^*$ Promedio
Reutilización	Baja	$> 2^*$ Promedio
	Media	Entre Promedio y $2^*$ Promedio
	Alta	$\leq$ Promedio

La tabla que se muestra a continuación ofrece las clases del sistema a las que se le aplicó la métrica y los resultados obtenidos para cada atributo evaluado. Para determinar el valor de los atributos, se calcula el promedio de la columna cantidad de procedimientos y este promedio (en este caso el resultado obtenido fue 12) es el que se emplea en la tabla 3 en la columna criterio.

**Tabla 4. Resultados de la aplicación de la métrica TOC para cada clase del sistema.**

No	Clase	Cantidad de Procedimientos	Responsabilidad	Complejidad	Reutilización
1	AdministrarNominasController	40	Alta	Alta	Baja
2	GestionarTiponominasController	11	Baja	Baja	Alta
3	RevisarNominasController	14	Media	Media	Media
4	DatNprocRetencionesModel	7	Baja	Baja	Alta
5	DatNprocNominalImpuestoModel	4	Baja	Baja	Alta
6	DatNprocNominaModel	12	Baja	Baja	Alta
7	DatNprocConceptoPagoModel	18	Media	Media	Media
8	DatNnominaModel	32	Alta	Alta	Baja
9	NomTipoNominaModel	10	Baja	Baja	Alta
10	NomTipoNominaConceptoModel	5	Baja	Baja	Alta
11	DatTipoNominaFuncionModel	5	Baja	Baja	Alta
12	DatNprocRetenciones	9	Baja	Baja	Alta
13	DatNprocNominalImpuesto	7	Baja	Baja	Alta
14	DatNprocConceptoPago	10	Baja	Baja	Alta
15	DatNnomina	23	Media	Media	Media
16	DatNprocNomina	18	Media	Media	Media
17	NomTipoNominaConcepto	9	Baja	Baja	Alta

18	NomTipoNomina	10	Baja	Baja	Alta
19	DatTipoNominaFuncion	7	Baja	Baja	Alta

Las gráficas que se muestran a continuación muestran los resultados obtenidos para cada uno de los atributos medidos.

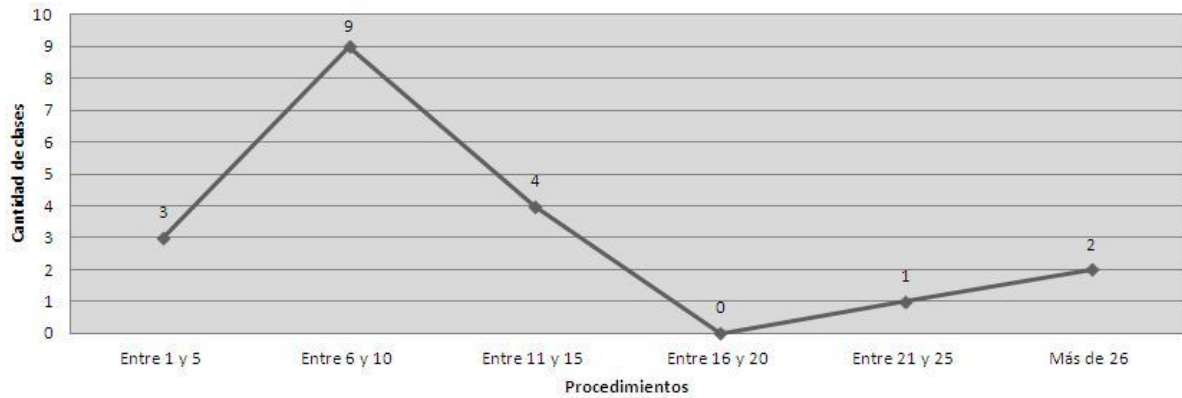


Figura 21. Resultados obtenidos de la aplicación de la métrica TOC en el instrumento agrupados en los intervalos definidos.

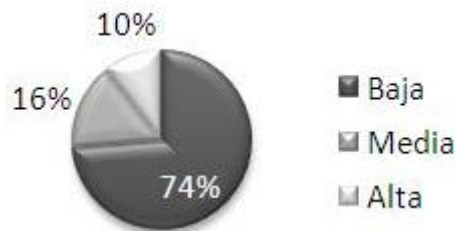
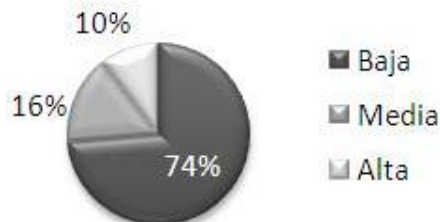
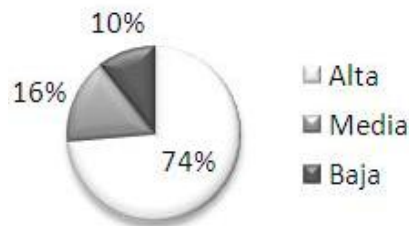


Figura 22. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.



**Figura 23. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de implementación.**



**Figura 24. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.**

Haciendo un análisis de los resultados obtenidos para los atributos de la métrica TOC en la evaluación del instrumento, se puede observar que la mayoría de las clases que conforman el sistema para los atributos responsabilidad y complejidad están dentro de la categoría Media y Baja para un 90% del total, mientras que el atributo Reutilización cuenta con igual por ciento en las categorías Alta y Media mostrando así que el componente cuenta con una elevada reutilización, baja complejidad y responsabilidad en el diseño propuesto. Por lo que se concluye que los resultados obtenidos según esta métrica son positivos.

### 3.2.2 Métrica Relaciones entre Clases

**Relaciones entre clases (RC):** Está dado por el número de relaciones de uso de una clase con otras.

**Tabla 5. Relaciones entre clases.**

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.



**Tabla 6. Rango de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica RC.**

Atributo	Categoría	Criterio
Acoplamiento	Ninguna	0
	Baja	1
	Media	2
	Alta	> 2
Complejidad del mantenimiento	Baja	$\leq$ Promedio
	Media	Entre Promedio y 2* Promedio
	Alta	> 2* Promedio
Reutilización	Baja	> 2* Promedio
	Media	Entre Promedio y 2* Promedio
	Alta	$\leq$ Promedio
Cantidad de pruebas	Baja	$\leq$ Promedio
	Media	Entre Promedio y 2* Promedio
	Alta	> 2* Promedio

La tabla 7 ofrece las clases del sistema a las que se le aplicó la métrica y los resultados obtenidos para cada atributo evaluado. El promedio utilizado para evaluar el criterio (tabla 6), es el resultado del cálculo del promedio de la columna Cantidad de relaciones de uso (tabla 7).

**Tabla 7. Resultados de la aplicación de la métrica RC para cada clase del sistema.**

No	Clase	Cantidad de Relaciones de Uso	Acoplamiento	Complejidad Mantenimiento	Reutilización	Cantidad de Pruebas
1	AdministrarNominasController	5	Alto	Alta	Baja	Alta
2	GestionarTiponom	3	Alto	Alta	Baja	Alta

	inaController					
3	RevisarNominasController	2	Medio	Media	Media	Media
4	DatNprocRetencionesModel	1	Bajo	Baja	Alta	Baja
5	DatNprocNominalmpuestoModel	1	Bajo	Baja	Alta	Baja
6	DatNprocNominaModel	2	Medio	Media	Media	Media
7	DatNprocConceptoPagoModel	2	Medio	Media	Media	Media
8	DatNnominaModel	2	Medio	Media	Media	Media
9	NomTipoNominaModel	1	Bajo	Baja	Alta	Baja
10	NomTipoNominaConceptoModel	1	Bajo	Baja	Alta	Baja
11	DatTipoNominaFuncionModel	1	Bajo	Baja	Alta	Baja
12	DatNprocRetenciones	0	Ninguno	Baja	Alta	Baja
13	DatNprocNominalmpuesto	0	Ninguno	Baja	Alta	Baja
14	DatNprocConceptoPago	0	Ninguno	Baja	Alta	Baja
15	DatNnomina	0	Ninguno	Baja	Alta	Baja
16	DatNprocNomina	0	Ninguno	Baja	Alta	Baja
17	NomTipoNominaConcepto	0	Ninguno	Baja	Alta	Baja
18	NomTipoNomina	0	Ninguno	Baja	Alta	Baja
19	DatTipoNominaFuncion	0	Ninguno	Baja	Alta	Baja

Las gráficas que se muestran a continuación muestran los resultados obtenidos para cada atributo medido.

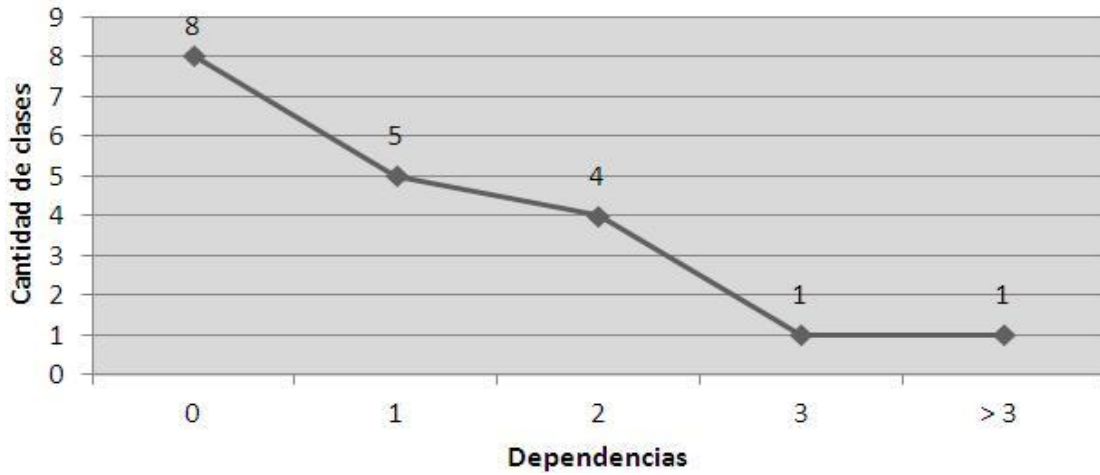


Figura 25. Resultados obtenidos de la aplicación de la métrica RC en el instrumento agrupados en los intervalos definidos.

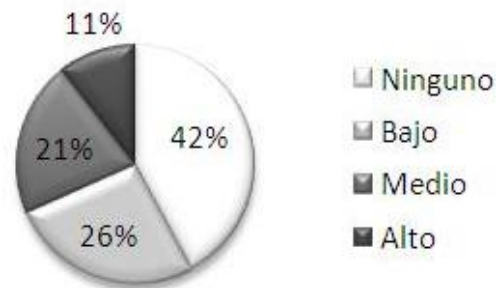


Figura 26. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento.

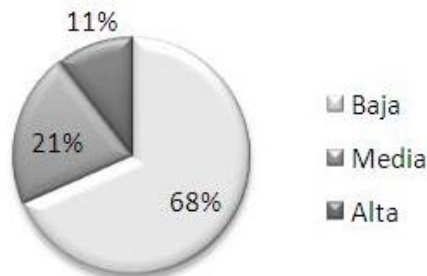
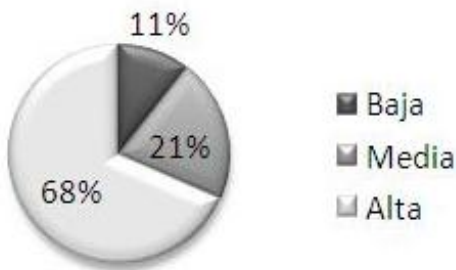
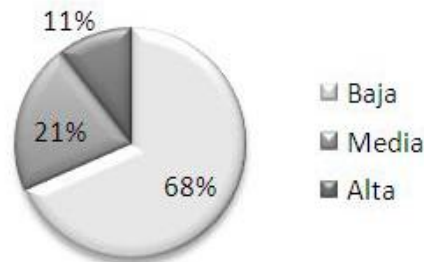


Figura 27. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Complejidad de mantenimiento.



**Figura 28. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Reutilización.**



**Figura 29. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Cantidad de pruebas.**

Los resultados obtenidos durante la evaluación del instrumento de medición de la métrica RC demuestran que las clases del diseño posee un bajo acoplamiento, ya que para este atributo las categorías ninguno y bajo sumaron un 68% del total, mostrando igual por ciento en la categoría alta del atributo reutilización. Los atributos complejidad de mantenimiento y cantidad de pruebas, sumaron un 89 % en las categorías baja y media, lo que demuestra que no es necesario un elevado esfuerzo en el momento de realizar cambios, rectificaciones y pruebas al software.

### 3.3 Pruebas de software

Las pruebas de software son una actividad en la cual el sistema es ejecutado bajo condiciones específicas, para demostrar que tiene o no, la madurez necesaria para ser implantado. Permiten determinar la calidad del producto, detectar todo posible mal funcionamiento y comprobar el grado de cumplimiento de las especificaciones iniciales del sistema. Se considera una prueba exitosa, si se demuestran deficiencias en el software. Las fallas pueden ser en el código o en el modelado, en

dependencia del tipo de pruebas que se le apliquen al software. La prueba no puede asegurar la ausencia de defectos; solo puede demostrar que existen defectos en el software.

### 3.3.1 Pruebas de Caja blanca o Estructurales

La prueba de caja blanca, denominada a veces prueba de caja de cristal, es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que:

1. Garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo.
2. Ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
3. Ejecuten todos los bucles en y con sus límites operacionales.
4. Ejerciten las estructuras internas de datos para asegurar su validez. (40)

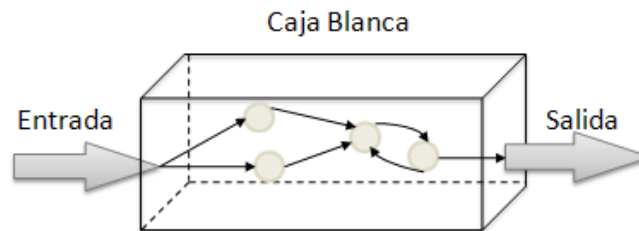


Figura 30. Representación de técnica de pruebas de caja blanca.

A continuación se citan algunas de las técnicas de prueba de Caja blanca:

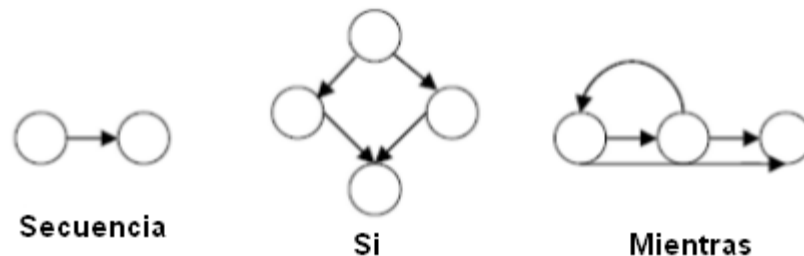
- Condición.
- Flujo de Datos.
- Bucles.
- Camino Básico.

Dentro de la prueba de caja blanca, la técnica que se utilizó fue Camino básico. Para aplicar esta técnica 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 31. Notación de grafos de flujo para las instrucciones: Secuenciales, Si, Mientras.**

Un grafo de flujo está formado por tres 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, el cual contiene 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 prueba del Camino básico es preciso calcular la complejidad ciclomática del algoritmo o fragmento de código a analizar. A continuación se muestra el código del método `adicionarNominaTrabAction()` encargado de adicionar trabajadores a la nómina.

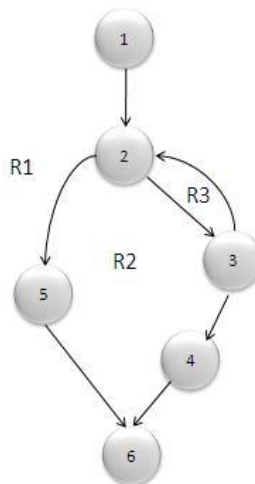
```

function adicionarNominaTrabAction() {
    $procnom = new DatNprocnominaModel(); 1
    $obj->idnomina = $this->_request->getPost('idnomina'); 1
    $data = $procnom->Obtener($obj); 1
    $obj->idtrabajador = $this->_request->getPost('idtrabajador'); 1
    for ($i = 0; $i < count($data); $i++) { 2
        if ($data[$i]['idtrabajador'] == $obj->idtrabajador) { 3
            echo -1; 4
            return;
        }
    }
    $params->idtrabajador = $this->_request->getPost('idtrabajador'); 5
    $trab = $this->pIntegrator->trabajador->BuscarTrabajador($params); 5
    $idpuesto->idpujestotrabajo = $trab[0]->idpujestotrabajo; 5
    $puesto = $this->pIntegrator->puesto_trabajo->ObtenerPuestoTrabajo($idpuesto); 5
    $salarioescala = $this->pIntegrator->movimientonomina->ObtenerSalarioEscalaActual($obj); 5
    $obj->salarioescala = $salarioescala[0]['salarioescala']; 5
    $obj->idcentrocosto = $puesto[0]->centrocosto; 5
    echo $procnom->adicionarNominaTrab($obj); 5
} 6

```

**Figura 32. Método que adiciona trabajadores a la nómina.**

Para el cálculo de la complejidad ciclomática es necesario representar el grafo de flujo asociado al código antes presentado a través de nodos, aristas y regiones, quedando como se muestra en la figura 33:



**Figura 33. Grafo de flujo asociado al método.**

Una vez construido el grafo de flujo asociado al procedimiento anterior se determina la complejidad ciclomática, el cálculo es necesario efectuarlo mediante tres vías o fórmulas, se debe utilizar el mismo grafo en cada caso:

**Fórmula 1** .  $V(G) = (A - N) + 2$

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

Resultado.

$$V(G) = (7-6) + 2$$

$$V(G) = 3$$

**Fórmula 2** .  $V(G) = P + 1$

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

Resultado.

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

$$V(G) = 3$$

**Fórmula 3** .  $V(G) = R$

- Siendo "R" la cantidad total de regiones, se incluye el área exterior del grafo, contando como una región más.

Resultado.

$$V(G) = 3$$

Seguidamente es necesario especificar los caminos básicos que puede tomar el algoritmo durante su ejecución. En estas representaciones se subrayan los elementos de cada camino que los hacen independientes a los demás.

**Camino básico # 1:** 1, 2, 5, 6.

**Camino básico # 2:** 1, 2, 3, 4, 6.

**Camino básico # 3:** 1, 2, 3, 2, 5, 6.

Es necesario aclarar que existen otros caminos en el grafo que no son considerados independientes ya que son combinaciones de los anteriormente especificados.

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 el resultado que se espera que devuelva el procedimiento.
- Resultados: Se muestra el resultado obtenido.
- Salida: Se presenta el valor final.

**Tabla 8. Caso de prueba para el camino básico 1**

<b>Camino básico # 1: 1, 2, 5, 6.</b>	
Descripción	A partir del identificador de la nómina, se obtienen todos los trabajadores asociados a esta. Después se verifica con el identificador del trabajador que este no pertenece a la nómina para posteriormente adicionarlo.
Condición de ejecución	Se debe tener el identificador del trabajador ('idtrabajador') Se debe tener el identificador de la nómina ('idnomina')
Entrada	<code>\$obj-&gt;idnomina = \$this-&gt;_request-&gt;getPost('idnomina');</code> <code>\$obj-&gt;idtrabajador = \$this-&gt;_request-&gt;getPost('idtrabajador');</code>
Resultados esperados	La nómina no tiene trabajadores por lo que se inserta el trabajador.
Resultados	Se adiciona el trabajador a la nómina.
Salida	1

**Tabla 9. Caso de prueba para el camino básico 2**

<b>Camino básico # 2: 1, 2, 3, 4, 6.</b>	
Descripción	A partir del identificador de la nómina, se obtienen todos los trabajadores asociados a esta. Después se verifica con el

	identificador del trabajador que este no pertenece a la nómina para posteriormente adicionarlo.
Condición de ejecución	Se debe tener el identificador del trabajador ('idtrabajador') Se debe tener el identificador de la nómina ('idnomina')
Entrada	<code>\$obj-&gt;idnomina = \$this-&gt;_request-&gt;getPost('idnomina');</code> <code>\$obj-&gt;idtrabajador = \$this-&gt;_request-&gt;getPost('idtrabajador');</code>
Resultados esperados	Se verifica que el trabajador forma parte de la nómina por lo que no se inserta.
Resultados	No se adiciona el trabajador a la nómina.
Salida	-1

**Tabla 10. Caso de prueba para el camino básico 3**

<b>Camino básico # 3: 1, 2, 3, 2, 5, 6</b>	
Descripción	A partir del identificador de la nómina, se obtienen todos los trabajadores asociados a esta. Después se verifica con el identificador del trabajador que este no pertenece a la nómina para posteriormente adicionarlo.
Condición de ejecución	Se debe tener el identificador del trabajador ('idtrabajador') Se debe tener el identificador de la nómina ('idnomina')
Entrada	<code>\$obj-&gt;idnomina = \$this-&gt;_request-&gt;getPost('idnomina');</code> <code>\$obj-&gt;idtrabajador = \$this-&gt;_request-&gt;getPost('idtrabajador');</code>
Resultados esperados	Se verifica que el trabajador no forma parte de la nómina por lo que se inserta.
Resultados	Se adiciona el trabajador a la nómina.
Salida	1

### Resultado de las pruebas de Caja blanca

Para la realización de las pruebas de caja blanca al sistema, fueron analizadas las funcionalidades más significativas, lo que permitió detectar errores en la implementación de las mismas. Luego de corregir los

errores encontrados, se pudo comprobar que el flujo de trabajo de las funciones está correcto ya que cumple con las condiciones necesarias que se habían planteado.

### 3.3.2 Pruebas de Caja negra

Cuando se utiliza el término pruebas de caja negra se está haciendo referencia a las pruebas que se llevan a cabo sobre la interfaz del software sin tener en cuenta el código, por lo que 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 una salida correcta, así como que la integridad de la información externa se mantiene.

El diseño de estas pruebas tiene el propósito de detectar (41):

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



**Figura 34. Representación de técnica de prueba de Caja negra.**

Existen diferentes técnicas de prueba de Caja negra descritas por Pressman (41) para validar la funcionalidad del sistema sin entrar a analizar su ejecución interna:

- Métodos de prueba basados en grafos.
- Análisis de valores límites.
- Tabla ortogonal.
- Partición de equivalencia.

De estas técnicas, la seleccionada fue partición de equivalencia la cual permite examinar los valores válidos e inválidos de las entradas existentes en el software.

Para la aplicación de esta técnica se realizan los diseños de casos de prueba los cuales se basan en una evaluación de las clases de equivalencia para una condición de entrada.

- Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada.
- Regularmente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica. (40)
- Las clases de equivalencia se pueden definir de acuerdo con las siguientes directrices:
  1. Si un parámetro de entrada debe estar comprendido en un cierto rango, aparecen 3 clases de equivalencia: por debajo, por encima y en el rango.
  2. Si una entrada requiere un valor concreto, aparecen 3 clases de equivalencia: por debajo, por encima y en el rango.
  3. Si una entrada requiere un valor de entre los de un conjunto, aparecen 2 clases de equivalencia: en el conjunto o fuera de él.
  4. Si una entrada es booleana, hay 2 clases: sí o no.

Los mismos criterios se aplican a las salidas esperadas: hay que intentar generar resultados en todas y cada una de las clases.

Se realizaron 25 diseños de casos de prueba para validar el componente atendiendo a la técnica partición de equivalencia. A continuación se especifica el caso de prueba para el requisito “Adicionar nómina”.

### **Descripción general**

Se debe seleccionar en el menú principal el subsistema Capital humano/Estimulación moral y material/Nómina/Administrar nómina.

### **Condiciones de ejecución**

- Se debe identificar y autenticar ante el sistema y además debe tener los permisos para ejecutar esta acción.
- Debe haber un período de pago abierto.
- Debe haber trabajadores y Tipos de nómina registrados en el sistema.

- Se debe haber al menos una pre Nómina en estado “Confirmada” en el período de pago abierto, en caso que se desee adicionar una nómina de tipo salario.
- Se debe presionar el botón Adicionar de la barra de opciones.

Tabla 11. Posibles escenarios al adicionar una nómina.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
Administrar nómina	EP 1.1: Adicionar una Nómina introduciendo datos válidos presionando el botón <b>Aceptar</b> .	El sistema debe permitir adicionar una Nómina.	<ul style="list-style-type: none"> <li>– Se introducen los datos de la Nómina correctamente.</li> <li>– Se presiona el botón <b>Aceptar</b>.</li> <li>– Se muestra un mensaje de información.</li> <li>– Se presiona el botón <b>Aceptar</b>.</li> </ul>
	EP 1.2: Adicionar una Nómina introduciendo datos válidos presionando el botón <b>Aplicar</b> .	El sistema debe permitir adicionar uno a una varias Nóminas.	<ul style="list-style-type: none"> <li>– Se presiona el botón <b>Adicionar</b>.</li> <li>– Se introducen los datos de la Nómina correctamente.</li> <li>– Se presiona el botón <b>Aplicar</b>.</li> <li>– Se muestra un mensaje de información.</li> <li>– Se presiona el botón <b>Aceptar</b>.</li> </ul>
	EP 1.3: Adicionar nómina dejando campos vacíos.	El sistema debe señalar en rojo los campos vacíos.	<ul style="list-style-type: none"> <li>– Se presiona el botón <b>Adicionar</b>.</li> <li>– Se introducen los datos dejando algún campo en blanco.</li> </ul>

- Se presiona el botón **Aceptar**.
  - Se muestra un mensaje informando del error.
- 
- Se presiona el botón **Adicionar**.
  - Se introducen o no los datos de la nómina.
  - Se presiona el botón **Cancelar**.

EP 1.4: Cancelar

El sistema debe cancelar la acción.

Tabla 12. Descripción de las variables del Adicionar nómina.

No	Nombre de campo	Tipo	Válido	Inválido	Inválido
1	Número de consecutivo (Nro. orden)	Campo de texto (No editable).	Números.	Letras y caracteres especiales.	Vacío.
2	Denominación	Campo de texto.	Letras y números.	Caracteres especiales.	Vacío.
3	Período	Campo de selección (No editable).	Letras y números.	Números y caracteres especiales.	Vacío.
4	Tipo de nómina	Campo de selección (No editable).	Letras y números.	Letras y caracteres especiales excepto el “/”.	Vacío.
5	Prenómina	Campo de selección (No editable).	Letras, números y Vacío si el tipo de nómina es Salario.	Caracteres especiales.	Vacío

Tabla 13. Respuestas del sistema para los posibles escenarios al adicionar una nómina.

ID	Escenario	Nro. consecutivo	Denominación	Período	Tipo de nómina	Prenómina	Respuesta del sistema
EP 1.1	Adicionar una Nómina introduciendo datos válidos presionando el botón <b>Aceptar.</b>	V(72)*	V(Salario)	V(Abril2011)	V(Salario)	V(nombre Prenómina)	El sistema adiciona la nómina y muestra el mensaje de información: "La nómina se ha adicionado satisfactoriamente ..".
		V(72)	V(Salario)	V(Abril2011)	V(Subsidio)	V(Vacío)	El sistema cierra la interfaz. El sistema actualiza el listado de nóminas.
EP 1.2	Adicionar una Nómina introduciendo datos válidos presionando el botón <b>Aplicar</b>	V(72)*	V(Salario)	V(Abril2011)	V(Salario)	V(nombre Prenómina)	El sistema adiciona la Nómina y muestra el mensaje de información: "Se ha adicionado la Nómina satisfactoriamente ..".
		V(72)	V(Salario)	V(Abril2011)	V(Subsidio)	V(Vacío)	El sistema mantiene la interfaz abierta. Se actualiza el número consecutivo y el resto de los campos se actualizan en blanco. El sistema actualiza el listado de nóminas.

EP 1.3	Adicionar nómina dejando campos vacíos.	V(72)*	I(Vacío)	V(Abril2011)	V(Subsidio)	V(Vacío)	El sistema muestra el mensaje: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)". El sistema subraya el campo en rojo mostrando el mensaje: "Este campo es obligatorio." El sistema mantiene la interfaz abierta.
		V(72)	V(Salario)	I(Vacío)	V(Subsidio)	V(Vacío)	
		V(72)	V(Salario)	V(Abril2011)	V(Vacío)	V(Vacío)	
		V(72)	V(Salario)	V(Abril2011)	V(Salario)	I(Vacío)	
EP 1.4	Cancelar	N/A	N/A	N/A	N/A	N/A	El sistema cierra la interfaz sin realizar ninguna operación.

### Resultado de las pruebas de Caja negra

Las pruebas de caja negra a la solución fueron desarrolladas por el Grupo de Aseguramiento de la Calidad del Centro de Informatización para la Gestión de Entidades (CEIGE). Estas se realizaron en cinco iteraciones, donde finalmente se comprobó en la quinta iteración que el componente estaba libre de no conformidades culminando así la fase de pruebas internas. En el anexo 3 se puede observar el acta de liberación emitida. El próximo paso para la liberación se está llevando a cabo por CALISOFT<sup>7</sup>, encontrándose actualmente en la segunda iteración.

\*Se autogenera. No modificable.

<sup>7</sup> CALISOFT: Centro de calidad para soluciones informáticas.



La tabla que se muestra a continuación ofrece detalles de los resultados obtenidos hasta el momento, desglosando las no conformidades (NC) detectadas en significativas (S) que incluyen errores de validación, de función y excepciones, las que no proceden (NP) y las no significativas NS que incluyen los errores en los casos de prueba.

**Tabla 14. Resultado de las iteraciones de las pruebas de caja negra realizadas por CALISOFT.**

Agrupación de requisitos	No Conformidades							
	Iteración 1				Iteración 2			
	NC	NP	S	NS	NC	NP	S	NS
Administrar Nómina	5	0	2	3	1	0	0	1
Gestionar Tipo de Nómina	4	0	4	0	4	0	2	2
Revisar Nómina	0	0	0	0	0	0	0	0
Total	9	0	6	3	5	0	2	3

### 3.4 Conclusiones parciales

En el capítulo se realizó la validación del diseño a través de las métricas seleccionadas, las cuales mostraron que el diseño cumple satisfactoriamente con los atributos medidos. A través de las pruebas de caja negra se validó la funcionalidad del sistema, mostrando cómo respondía a los requisitos funcionales, contribuyendo a una mayor satisfacción de las necesidades de los usuarios y demandas del cliente. De igual forma se aplicaron pruebas de caja blanca para efectuar las revisiones al código, las cuales permitieron que ocurran la menor cantidad de errores en el software.

## **CONCLUSIONES**

Con la realización y culminación del presente trabajo, se logró obtener un componente configurable y totalmente funcional capaz de gestionar de manera integral los principales procesos de la nómina que se llevan a cabo en las entidades cubanas. Este componente contribuye al proceso de informatización del país y posibilita realizar el cálculo para el pago del salario a los trabajadores con el Sistema Integral de Gestión Cedrux, facilitando el manejo de los datos y contribuyendo a la seguridad e integridad de la información en las entidades.

Para lograr dicho resultado se realizó un estudio que permitió conocer detalles del proceso de gestión de las nóminas. El análisis de varios sistemas que incluyen dicho proceso, concluyó que es más factible crear una solución propia que responda a las particularidades de la economía cubana y cumpla con el paradigma de independencia tecnológica por la que aboga el país. Se detallaron las tecnologías y herramientas seleccionadas para la implementación del componente y se efectuó el modelado del sistema aplicando patrones, que contribuyeron a un ahorro considerable de tiempo en la construcción del software y la reutilización del código. Se realizó la validación del diseño aplicando métricas donde los resultados arrojados demuestran la presencia de valores positivos en los indicadores de calidad medidos.

Se desarrolló el componente Nómina a partir del diseño validado haciendo uso del marco de trabajo Sauxe, como solución al problema planteado y finalmente para minimizar la ocurrencia de errores del sistema antes de su implantación en las entidades nacionales se realizaron pruebas al mismo.

## **RECOMENDACIONES**

Se recomienda:

- Agregar funcionalidades al componente nómina que permitan la gestión de nominillas y multimoneda.
- Profundizar en temas referentes a la gestión de las nóminas para detectar posibles debilidades en el componente y agregar mejoras al mismo.
- Realizar el despliegue del componente propuesto como parte del subsistema de Capital Humano del sistema Cedrux.

## REFERENCIAS BIBLIOGRÁFICAS

1. **S., Larissa G. Martínez.** GestioPolis. *Administración de recursos humanos*. [En línea] enero de 2004. [Citado el: 10 de diciembre de 2011.] <http://www.gestiopolis.com/recursos/documentos/fulldocs/rrhh1/admonrrhhlari.htm>.
2. **Genix, Ing. Alina de la Concepción Isasi.** *Estudio sobre la información que se gestiona de los recursos humanos, en los ERP*. Habana : DESOFT, 2010.
3. **Recursos, Sistema de Gestión Integrada de los.** *Norma Cubana*. La Habana : s.n., 2007. 3000.
4. **Española, Real Academia.** *Diccionario de la Lengua Española*. [En línea] [Citado el: 6 de diciembre de 2011.] [http://buscon.rae.es/draeI/SrvltConsulta?TIPO\\_BUS=3&LEMA=nomina](http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=nomina).
5. **Soto, Lauro.** *Nómina*. [En línea] 2010. [Citado el: 9 de diciembre de 2011.] <http://www.mitecnologico.com/Main/Nomina>.
6. **Catillo Ortiz, Isabel Natali, y otros.** *Nómina, su análisis y aplicación*. Mexico, D.F : s.n., 2009.
7. **sage.** Sage ERP Mas. [En línea] <http://www.sagemas.com/Products/Sage-ERP-MAS-500/Human-Resources-Management/Abra-Payroll>.
8. **Sage Software, Inc.** SageMas. [En línea] 12 de 2009. [Citado el: 24 de 2 de 2012.] [http://www.dsdinc.com/dsd/pdf/MAS\\_500\\_StarShip\\_Parcel\\_spec.pdf](http://www.dsdinc.com/dsd/pdf/MAS_500_StarShip_Parcel_spec.pdf). 56.
9. **Partner, Sage North America Top ERP.** Blytheco. [En línea] [Citado el: 5 de 5 de 2012.] <http://www.blytheco.com/mas500/price-list.asp>.
10. **2000, Seven.** seven2k. [En línea] Amplus Sistemas C.A. [Citado el: 15 de 2 de 2012.] <http://www.seven2k.com/index.html>.
11. Catálogo de software. [En línea] DIGITAL WARE S.A., 29 de Diciembre de 2011. <http://www.catalogodesoftware.com/producto-seven-erp-234>.
12. Assets. [En línea] Assets. [Citado el: 20 de 1 de 2012.] <http://www.assets.co.cu/>.
13. SAP. [En línea] [Citado el: 12 de 3 de 2012.] <http://www.sap.com>.

14. **Lic. Antonio Toledo Carnicero, Lic. Pablo Perez Perez.** *GESTIÓN DE SISTEMAS DE INFORMACIÓN.* Universidad de Deusto : Facultad de Ingeniería, 2004.
15. **EICMA, Dirección.** EICMA. *Versat Sarasola, Un efectivo sistema cubano de contabilidad.* [En línea] 23 de septiembre de 2010. [Citado el: 9 de diciembre de 2011.] <http://www.eimagr.cu/index.php/component/content/article/60-portada/101-versat-sarasola-un-sistema-para-trabajar>.
16. **autores, Colectivo.** *Manual del Usuario y de Explotación Versat Sarasola.* Villa Clara, Cuba : s.n.
17. **CITMAEL.** *Sistema Integral Administrativo Rodas XXI y Servicios Postventa.* [En línea] [Citado el: 8 de diciembre de 2011.] [http://www.citmatel.cu/geren1\\_1.php](http://www.citmatel.cu/geren1_1.php).
18. **CITMATEL.** RodasXXI. [En línea] CITMATEL, 2002-2012. [Citado el: 15 de 1 de 2012.] <http://www.rodasxxi.cu/>.
19. **Vega Miniet, Ing. Yanet, y otros.** *Ciclo de vida del proyecto.* 2009.
20. **Baryolo, Oiner Gómez, y otros.** *Plantilla Registro de la Propiedad intelectual(Sauxe).* Habana : s.n., 2008.
21. **Baryolo, Oiner Gómez.** *SOLUCIÓN INFORMÁTICA DE AUTORIZACIÓN EN ENTORNOS MULTIENTIDAD Y MULTISISTEMA.* La Habana : s.n., 2010.
22. **Gómez Baryolo, Oiner, Morejón Borbón, Yoandry y Garcia, Darien.** *ARQUITECTURA TECNOLÓGICA PARA EL DESARROLLO DE SOFTWARE.* [En línea] 22 de mayo de 2011. [Citado el: 10 de diciembre de 2011.] <http://www.buenastareas.com/ensayos/Arquitectura-Tecnol%C3%B3gica-Para-El-Desarrollo-De/2210642.html>.
23. **Pérez, Mario Raúl.** Google Sites. *Doctrine.* [En línea] 6 de julio de 2011. [Citado el: 10 de diciembre de 2011.] <https://sites.google.com/site/zendframeworkextjsdoctrine/guias/doctrine>.
24. **SOUZA, ANA CAROLINA CARVALHO DE PAULA.** EspWeb. [En línea] 4 de Abril de 2011. [Citado el: 21 de Mayo de 2012.] <http://www.espweb.uem.br/wp/wp-content/uploads/2011/09/anacarolina.pdf>.
25. **Nidia.** Buenas tareas. *Lenguaje De Programación.* [En línea] 20 de septiembre de 2011. [Citado el: 11 de diciembre de 2011.] <http://www.buenastareas.com/ensayos/Lenguaje-De-Programacion/2770237.html>.

26. **Sons, John Wiley &.** *Javascript® bible, sixth edition.* New York : Inc. New York, 2007. 9780470069165.
27. **Flanagan, David.** *Java Script. The definitive guide.* Estados Unidos de America : O'Reilly, 2006. 978-0-596-10199-2.
28. **Rasmus Lerdorf, Kevin Tatroe & Peter MacIntyre.** *Programming PHP.* Estados Unidos de América : O'Reilly, 2006. 978-0-596-00681-5.
29. Visual Paradigm. *Visual Paradigm for UML - UML tool for software application development.* [En línea] 13 de marzo de 2012. <http://www.visual-paradigm.com/product/vpuml/>.
30. **UBUNTU, GUÍA DOCUMENTADA PARA.** *Subversion.* [En línea] 9 de enero de 2009. [Citado el: 7 de diciembre de 2011.] <http://www.guia-ubuntu.org/index.php?title=Subversion>.
31. The Apache Software Foundation. *Apache Software Foundation Index: Project Listing.* [En línea] [Citado el: 10 de Marzo de 2012.] <http://projects.apache.org/indexes/quick.html>.
32. **López, Alejandro Cadavid.** Mozilla Firefox, el navegador web del momento. [En línea] [Citado el: 10 de diciembre de 2011.] <http://www.maestrosdelweb.com/editorial/firefox/>.
33. **Tedeschi, Nicolás.** Microsoft MSDN. *¿Qué es un Patrón de Diseño?* [En línea] [Citado el: 28 de febrero de 2012.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.
34. **Alexander, Christopher.** *The Timeless Way of Building.* 1979.
35. **Schmidt, Douglas, y otros.** *PATTERN-ORIENTED SOFTWARE ARCHITECTURE VOLUME 2: Patterns for Concurrent and Networked Objects.* 2000.
36. **Martínez, Alejandro y Martínez, Raúl.** Guía a Rational Unified Process. *Universidad de Castilla-La Mancha.* [En línea] [Citado el: 28 de febrero de 2012.] <http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-Guia%20RUP.pdf>.
37. ebookbrowse.com. [En línea] 29 de Marzo de 2011. [Citado el: 5 de Marzo de 2012.] <http://ebookbrowse.com/tema-1-normalizaci%C3%B3n-bibliograf%C3%ADa-libro-de-bd-de-rosa-maria-pdf-d91217792>.

38. **Facultad de Informática – UPM.** *fermat.usach.cl.* [En línea] [Citado el: 17 de Febrero de 2012.] [fermat.usach.cl/~msanchez/comprimido/OBJETOS.pdf](http://fermat.usach.cl/~msanchez/comprimido/OBJETOS.pdf).
39. *es.scribd.com.* [En línea] 11 de Noviembre de 2008. [Citado el: 24 de Febrero de 2012.] <http://es.scribd.com/doc/7884665/Arquitectura-de-Software-II-Diagrama-de-Componentes-y-Despliegue>.
40. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque práctico.* s.l. : MC Graw Gil, 1998.
41. **Pressman, Roger.** *Ingeniería del Software. Un enfoque práctico.* Interamericana de España. : McGraw-Hil, 2002.

## **GLOSARIO DE TÉRMINOS**

**Capital Humano:** Cuantificación y valoración de los recursos humanos. Valor de las habilidades, capacidades, experiencias y conocimientos de las personas que integran una organización.

**Gestión:** Es el conjunto de diligencias que se realizan para desarrollar un proceso o para lograr un producto determinado. Es también la dirección o administración de una empresa o de un negocio.

**Entidad:** Empresa, unidad presupuestada u otro tipo de organización similar con una gestión económica, financiera, organizativa, técnica, productiva, comercial, laboral y contractual, con autonomía controlada, en cumplimiento de lo establecido por el Gobierno.

**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.

**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.

**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.

**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.

**Trabajador:** Es aquella persona que ocupa un puesto de trabajo en la entidad y desempeña una labor determinada por la cual recibe haberes o salarios.