

*Universidad de las Ciencias Informáticas*



*“Implementación y prueba del Componente Pago por resultados del subsistema Capital humano del Sistema Integral de Gestión CedruX”*

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

*Autor: Yaily Leiva Valdivia*

*Tutor: Ing. Arnolis Salgueiro Arzuaga*

*Co-tutor: Ing. Aylín Morales Ortiz*

*Ciudad de La Habana*

*Junio, 2013*

*Declaración de auditoría*

Declaro ser autora 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 firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

-----

**Yaily Leiva Valdivia**

**Autor**

-----

**Ing. Arnolis Salgueiro Arzuaga**

**Tutor**

-----

**Ing. Aylin Morales Ortiz**

**Co-tutor**

***Datos del contacto***

**Ing. Arnolis Salgueiro Arzuaga:** Ingeniero en Ciencias Informáticas. Graduado en la Universidad de las Ciencias Informáticas en el 2008. Actualmente se desempeña como jefe del proyecto Capital humano del Departamento de Aduana, CEIGE, Facultad 3.

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

**Ing. Aylin Morales Ortiz:** Ingeniera en Ciencias Informáticas. Graduada de la Universidad de las Ciencias Informáticas en el 2011. Actualmente se desempeña como Analista en la Línea de Capital humano del centro CEIGE en la UCI.

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



*Tener no es poseer. Puede tenerse aquello que no se desea. Posesión*

*José Saramago*

## *Agradecimientos*

*Quiero agradecer principalmente a mi mamá Ramona y papá Miguel por hacer de mí lo que hoy soy, por apoyarme en los momentos más difíciles y compartir conmigo mis alegrías y tristezas, por el sacrificio que han hecho para que yo estuviera en esta universidad.*

*A mi hermano que aunque lo regaño mucho y le digo las cosas como son, siendo yo la más pequeña, lo quiero y lo respeto más que a nada en el mundo.*

*A mi prima Lisandra, que al decirle prima me quedo corta, ella es la hermana que no tuve, a la que respeto y admiro, porque siendo menor que yo me ha mostrado que la vida se tiene que coger a la ligera y darle importancia a las personas que de verdad te admiran y aprecian.*

*A mi tío Teodoro y a mi tía Ana María.*

*A mis amigas Yuliet, Lisvey, Yanira y Dayana, que el orden no las afecte, las quiero a todas con el corazón, ustedes han demostrado ser más que amigas, las tengo como mis hermanitas a las que le debo mucho en esta vida, además agradecerle a sus padres que también se preocuparon como si fuera una hija más.*

*A mi amigo Liandry Mosteslier, que aunque digan que una mujer no tiene amigos, yo rompo la excepción, él ha sido amigo, mi confidente, mi hermano, él me ha corregido, me ha enseñado y me ha demostrado que se debe respetar a todos aunque dicho respeto no sea reciproco, y agradecerle a su mamá Nancy que se preocupó por mí y estar pendiente de mis recaídas.*

## *Agradecimientos*

*A mi Tía Marilyn (la vieja) como le digo de cariño, que ha sido como una madre para mí en todo este tiempo, me ha apoyado en todo y ha estado pendiente de todo lo que me ha pasado, y también a sus hijos Liset y Ahmed.*

*A Julio German, que ha sido la personas que más me marcó en esta universidad, con él aprendí que amar: es alegría, tristeza y debilidad, él aguantó mis ataques y pesadeces, y a pesar que las cosas no funcionaron entro nosotros, me apoyó mucho y estuvo conmigo hasta el final, también agradecerle a su mamá Juana, que a pesar de que compartimos muy poco, la aprecio y la respeto por darme un lugarcito en su corazón y apoyarme cuando más lo necesitaba.*

*A mis compañeros de estudio, Lianet, Claudia, Rafael, Yanier, Jorge, Wilfredo, Adrián (Pelú), Yeni (la gordita), Luis Ángel (lafis) y especialmente a Rogelio por estar conmigo hasta el final apoyándome en todo lo que me hizo falta.*

*A Marvileivis (Presley) y a René Leandro, que me ayudaron mucho en el momento en que los necesité, cuando estuve enredada programando.*

*A Lucrecia y Eiler que les debo gran parte de este logro.*

*A mis tutores por su apoyó en las últimas circunstancias.*

*A mis compañeros de Capital humano, proyecto donde trabajé.*

*A mi oponente que estuvo a su disposición en ayudarme en lo que fuera.*

*A todos que de alguna forma formaron parte de mi vida universitaria.*

*Gracias a todos los que tuvieron FE en mí.*

***Dedicatoria***

*Me dedico este trabajo, ya que yo he sido la principal partícipe.*

*A mi sobrina Melany, que hace poco ha llegado a la familia y a la cual espero servir como guía para la vida y enseñarle ser mujer y amiga en el futuro.*

*A mis padres y a mi hermano por ser parte de mi vida, a ellos le dedico este título, pues se sacrificaron junto conmigo.*

## **Resumen**

En la Universidad de las Ciencias Informáticas (UCI) se desarrolla un sistema integral de gestión llamado Cedrux. Contiene un conjunto de líneas productivas dentro de las que se encuentra la encargada de desarrollar el subsistema Capital humano. Este sostiene un módulo para la Estimulación Moral y Material que incluye los procesos para el cálculo de la nómina de salario de los trabajadores.

El presente trabajo propone un componente para el Pago por resultados que esté contenido dentro del módulo de Estimulación Moral y Material del subsistema Capital humano de Cedrux y que se integre al componente Nómina.

La solución se basa en tecnologías, lenguajes y herramientas establecidos en el marco de trabajo Sauxe y guiado por el modelo de desarrollo propuesto por el Centro de Informatización de la Gestión de Entidades (CEIGE) para el desarrollo de Cedrux.

Este documento recoge las trazas de la investigación realizada y describe el desarrollo del componente Pago por resultados, para lo cual se realiza un estudio de sistemas de gestión empresarial que soportan la Estimulación Moral y Material.

Palabras claves: Capital humano, componente, Estimulación Moral y Material, Pago por resultado, sistema integral de gestión.

**Índice**

**INTRODUCCIÓN..... 1**

**CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA..... 4**

1.2 Principales conceptos..... 4

1.3 Punto de partida para la implementación..... 5

1.4 Sistemas que incluyen el proceso de Pago por resultados ..... 6

1.4.1 ASSETS NS..... 6

1.4.2 SAGE MAS 500 ..... 6

1.4.3 SAP ..... 6

1.4.4 Versat-Sarasola ..... 6

1.4.5 Rodas XXI ..... 7

Resultados del estudio de los sistemas informáticos ..... 7

1.5 Modelo de desarrollo del software ..... 8

1.5.1 Descripción general de la disciplina implementación ..... 9

1.5.2 Descripción general de la disciplina Pruebas internas ..... 9

Conclusiones parciales ..... 9

**CAPÍTULO 2: PROPUESTA DE SOLUCIÓN..... 10**

2.1 Introducción..... 10

2.2 Funcionalidades que debe cumplir el componente ..... 10

2.3 Patrones de diseño ..... 10

2.4 Diseño del sistema ..... 11

Conclusiones parciales ..... 22

**CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA..... 23**

3.1 Introducción..... 23

3.2 Implementación ..... 23

3.3 Pruebas de software ..... 34

Conclusiones parciales ..... 41

**CONCLUSIONES..... 42**

**RECOMENDACIONES..... 43**

**REFERENCIAS BIBLIOGRÁFICAS ..... 44**

**ANEXOS ..... 49**

## **Índice de figuras**

<b>Figura 1:</b> Ciclo de vida de proyectos CEIGE. ....	8
<b>Figura 2:</b> Gestionar sistema de Pago por resultados.....	12
<b>Figura 3:</b> Patrón Fachada. ....	13
<b>Figura 4:</b> Patrón Experto. ....	13
<b>Figura 5:</b> Patrón Controlador.....	14
<b>Figura 6:</b> Patrón Creador. ....	14
<b>Figura 7:</b> Modelo de datos. ....	18
<b>Figura 8:</b> Diagrama de componentes. ....	19
<b>Figura 9:</b> Diagrama de despliegue de escenario para PC cliente sin disco. ....	21
<b>Figura 10:</b> Diagrama de despliegue de escenario para PC cliente con disco. ....	21
<b>Figura 11:</b> Contenido de las carpetas apps y web para Capital humano. ....	23
<b>Figura 12:</b> Contenido del fichero loC.xml interno. ....	25
<b>Figura 13:</b> Contenido del fichero loC.xml externo. ....	25
<b>Figura 14:</b> Complejidad algorítmica de la funcionalidad calcular el pago por resultados. ....	26
<b>Figura 15:</b> Funcionalidad contabilizar Nómina.....	27
<b>Figura 16:</b> Validar áreas y cargos que no estén repetidos en un mismo indicador. ....	28
<b>Figura 17:</b> Asociar Trabajador para ser penalizado.....	28
<b>Figura 18:</b> Insertar evaluación de los indicadores. ....	29
<b>Figura 19:</b> Escenario Gestionar indicadores. ....	30
<b>Figura 20:</b> Escenario Gestionar penalización.....	31
<b>Figura 21:</b> Escenario Evaluación de Indicadores. ....	32
<b>Figura 22:</b> Escenario total de evaluaciones.....	32
<b>Figura 23:</b> Adicionar nómina pago por resultados. ....	33
<b>Figura 24:</b> Revisar Nóminas.....	34
<b>Figura 25:</b> Representación de técnica de pruebas de caja blanca. ....	35
<b>Figura 26:</b> Notación de grafos de flujo para las instrucciones: Secuenciales, Si, Mientras.....	35
<b>Figura 27:</b> Funcionalidad Calcular Pago por resultados. ....	36
<b>Figura 28:</b> Grafo de flujo asociado al método.....	37
<b>Figura 29:</b> Representación de técnica de prueba de Caja negra.....	39

**Índice de tablas**

**Tabla 1:** Resultados del estudio de los sistemas informáticos. .... 7

**Tabla 2:** Clase controladora GestionarpagoresultadoControlle.php..... 14

**Tabla 3:** Clase modelo DatPagoresultadoModel.php..... 15

**Tabla 4:** Clase del dominio DatPagoresultado.php. .... 16

**Tabla 5:** Servicios que utiliza el componente Pago por resultados. .... 19

**Tabla 6:** Servicios que brinda el componente Pago por resultados. .... 20

**Tabla 7:** Caso de prueba para el camino básico 1..... 38

**Tabla 8:** Caso de prueba para el camino básico 2..... 38

**Tabla 9:** Iteraciones realizada por el proyecto Capital humano. .... 40

**Tabla 10:** Iteraciones realizadas por el grupo de calidad del centro. .... 41

### **INTRODUCCIÓN**

El Pago por resultados es un Sistema de pago<sup>1</sup> para estimular el trabajo y la calidad del mismo. Este responde a la Ley de distribución socialista: “de cada cual según su capacidad, a cada cual según su trabajo” y en nuestro país está regida por la Resolución 9 del Ministerio de Trabajo y Seguridad Social (MTSS), puesta en vigor en el 2008.

El Pago por resultados es una herramienta para lograr mayores incrementos de la productividad, la eficiencia y la calidad para que quienes más y mejor laboren reciban más, mientras el empeño en el trabajo se exprese en una mayor productividad y eficiencia económicas (1).

Actualmente los sistemas de pagos están contenidos en sistemas informáticos dedicados a la gestión empresarial o sistemas ERP<sup>2</sup>. Cuba, ampliando su visión tecnológica se lanza en el empleo de dichos sistemas informáticos para así mejorar la calidad de sus procesos y entre ellos el de pago por resultado, haciendo posible que el trabajador obtenga de su trabajo lo que ha sido capaz de hacer (2).

En la UCI 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 y promover el aporte laboral de los trabajadores en la consecución de los objetivos estratégicos y la elevación de la cultura de la organización, así como la satisfacción individual y colectiva de los trabajadores (3). Mientras que la estimulación material es un sistema de acciones que interactúan y se integran con la estimulación moral socialista para motivar a los trabajadores en el logro de la eficiencia y eficacia y en la consecución de los objetivos estratégicos de la organización (3). 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 del Pago por resultados, responsable de mostrar los salarios, retenciones, tiempo de trabajo y demás datos de cada trabajador en un período de pago.

---

<sup>1</sup> Los sistemas de pagos: son las modalidades de la forma de pago que se adoptan en correspondencia con las

<sup>2</sup> Planificación de recursos de empresariales (en inglés **ERP**, enterprise resource planning) Son sistemas de gestión de información que integran y automatizan muchas de las prácticas de negocio asociadas con los aspectos operativos o productivos de una empresa.

Hoy en día, en la mayoría de las empresas cubanas, el proceso Pago por resultados se realiza de forma manual y aunque algunas lo han automatizado con el uso de aplicaciones informáticas, no existe una herramienta estándar para ello, lo que trae como consecuencia la falta de uniformidad en la obtención de la información a nivel nacional. Además la mayoría de estas aplicaciones son privativas lo que trae como resultado un gasto elevado para el país por concepto de licencias de software, además las mismas trabajan de forma independiente y no existe la vinculación con otros sistemas. Al no estar automatizado el proceso genera gran cantidad de documentos que provoca que la tarea no se pueda realizar con la calidad que la misma requiere, lo trae consigo que se genere una gran cantidad de documentos en formato duro, lo que provoca que este se convierta en una tarea tediosa y lenta. Además que atenta contra la velocidad de respuesta y la fiabilidad de la información que se gestiona en la empresa siendo estas características de gran importancia en este proceso.

Se han realizado algunos estudios acerca de la gestión del Pago por resultados para las entidades cubanas especialmente para el subsistema Capital humano del Sistema Integral de Gestión Cedrux, los cuales han dado como resultado el aumento y disminución de dicho proceso pero aún no se tiene una solución completa.

A partir de la situación problemática anteriormente planteada se define como **problema a resolver**: La forma en que se realiza actualmente el Pago por resultados en las entidades cubanas no garantiza el control de la información que se gestiona y retrasa el desarrollo del proceso. Definiéndose como **objeto de estudio**: La gestión del Pago por resultados y delimitando como **campo de acción**: El proceso de Pago por resultados en las entidades cubanas.

Para llevar a cabo esta investigación se plantea como **objetivo general**: Desarrollar el componente Pago por resultados como parte del proceso Estimulación Moral y Material del subsistema Capital humano del Sistema Integral de Gestión Cedrux que garantice el control de la información que se gestiona en las entidades cubanas y agilice el desarrollo del proceso.

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 del Pago por resultados.
- Implementar el componente Pago por resultados partiendo del diseño previamente realizado.
- Validar el componente obtenido mediante la realización de pruebas.

Concretándose como **idea a defender**: Con el desarrollo del componente Pago por resultados como parte del proceso Estimulación Moral y Material del subsistema Capital humano del Sistema Integral de Gestión Cedrux se garantiza el control de la información en las entidades cubanas y agiliza el desarrollo del proceso.

**Tareas a cumplir:**

- Estudio de trabajo científico relacionado con el Pago por resultados para Cedrux.
- Análisis de los sistemas existentes para la gestión del Pago por resultados identificando sus características y deficiencias fundamentales.
- Caracterización de las tecnologías, lenguajes y herramientas propuestas para el desarrollo del componente.
- Definición de servicios a brindar y consumir por el componente.
- Realización del diagrama de despliegue teniendo en cuenta los aspectos técnicos de la solución.
- Realización la descripción por funcionalidades.
- Implementación del componente Pago por resultados partiendo del diseño realizado.
- Validación del sistema a través de pruebas de liberación internas.

**Posibles resultados:** Obtención del componente Pago por resultados integrado al subsistema Capital humano del Sistema Integral de Gestión Cedrux que soporte los requisitos identificados en el estudio del proceso.

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

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

**CAPÍTULO 2: Propuesta de solución:** Se describe la implementación de la solución realizando las integraciones entre el componente desarrollado y los existentes del sistema integral de gestión Cedrux.

**CAPÍTULO 3: Implementación y prueba:** Se abordan las pruebas realizadas al componente, en específico las pruebas de caja blanca y caja negra, además se hace una valoración de las mismas según los resultados obtenidos.

### **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

#### **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 Pago por resultados, 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 del proceso Pago por resultados y se mencionan las tecnologías utilizadas para el desarrollo de la solución propuesta en la tesis de análisis y diseño del componente Pago por resultados.

#### **1.2 Principales conceptos**

##### **Administración del Capital humano**

En este proceso se realiza el registro y actualización de la información de los trabajadores. Se realizan los movimientos de la fuerza de trabajo dando como resultado los Movimientos de nómina. Se registran las incidencias de los trabajadores dando como resultado la pre-nómina donde se refleja el tiempo trabajado. También se llevan a cabo las actividades relacionadas con las vacaciones, la seguridad social, disciplina laboral, tratamiento a las interrupciones laborales y disponibilidad de trabajadores (4).

##### **Estimulación material**

Sistema de acciones que interactúan y se integran con la estimulación moral, para motivar a los trabajadores en el logro de la eficiencia y eficacia y en la consecución de los objetivos estratégicos de la organización. El pago con arreglo al trabajo, por cantidad y calidad, es el elemento principal de la estimulación material (3).

##### **Estimulación moral**

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 consecución de los objetivos estratégicos y la elevación de la cultura de la organización, así como la satisfacción individual y colectiva de los trabajadores (3).

##### **Pago por resultados**

El proceso Pago por resultados son los pagos que se realizan, teniendo en cuenta los niveles de cumplimiento de los indicadores establecidos en los sistemas de pago en cualquiera de sus modalidades. En correspondencia con los resultados alcanzados se obtiene un salario que puede ser igual, superior o inferior al salario escala, más los incrementos debidamente aprobados y que procedan, según tiempo real trabajado (4).

## *Capítulo 1: Fundamentación teórica*

### **1.3 Punto de partida para la implementación**

El siguiente trabajo de diploma es la continuidad de una tesis que abordó el análisis y diseño del proceso Pago por resultados, donde se estudiaron los sistemas informáticos existentes en Cuba que realizan el proceso Pago por resultado, estos fueron Versat Sarasola y Rodass XXI, cuyos resultados están descritos el trabajo de diploma Análisis y diseño del componente Pago por resultados: (5) .

Para el desarrollo de la solución se propone el uso del marco de trabajo Sauxe, el mismo está compuesto por un grupo de herramientas y tecnologías entre las que se encuentran **Zend Framework 1.7** para el desarrollo de aplicaciones web, **Doctrine 1.2.1** para el acceso a la base de datos y **Extjs 2.2** para la construcción de interfaces. Como lenguaje de programación se utiliza del lado del cliente **JavaScript** y del lado del servidor **PHP 5.2.6**, como herramienta para el modelado **Visual Paradigm 6.4**, para el desarrollo **Netbeans 6.9**, como gestor de base de datos **PostgreSQL 8.3**, como servidor de aplicaciones web **Apache 2.2.9** y **Mozilla Firefox 2.17** o superior como navegador web.

La investigación anterior propone la utilización del modelo de desarrollo propuesto por el centro CEIGE, transitándose por las primeras 3 disciplinas y obteniéndose los resultados y artefactos que se listan a continuación: (5).

- En la disciplina del negocio se realizaron los mapas de relaciones internas y externas y los diagramas de proceso del negocio. Una vez identificados los procesos de negocio se realizó el modelo conceptual con el objetivo de identificar las entidades del negocio. Se identificaron y se describieron los requisitos funcionales y no funcionales que debe presentar la solución a construir (5).
- Durante la disciplina de requisitos se generaron 9 agrupaciones de requisitos que generan un total de 73 requisitos.
- Durante el diseño se hace un estudio de los patrones GRASP (**G**eneral **R**esponsibility **A**ssignment **S**oftware **P**atterns, Patrones Generales de Software para Asignación de Responsabilidades), dentro de los cuales se proponen para la implementación de la solución el uso de: Experto, Creador, Alta cohesión, Bajo acoplamiento y Controlador. También se realizó la validación del diseño a través de las métricas Tamaño operacional de clases (TOC) y Relación entre clases (RC), el resultado de las métricas RC demuestran que el diseño para el proceso de Pago por resultados es satisfactorio pues se comprobó el bajo acoplamiento, la baja complejidad y la baja cantidad de pruebas que se necesitan realizar en el diseño, confirmando la elevada reutilización del mismo. Para la arquitectura se propone el uso del patrón arquitectónico Modelo Vista Controlador (MVC) (5).

## *Capítulo 1: Fundamentación teórica*

En la tesis desarrollada anteriormente no se realizó un estudio de los patrones GoF los cuales en el presente trabajo se estudian y se aplican, también se conformaron los diagramas del diseño que faltaban como parte de la solución de este trabajo.

### **1.4 Sistemas que incluyen el proceso de Pago por resultados**

Se continuó con el estudio de los sistemas estudiados en la tesis de análisis y diseño, y además se estudiaron otros sistemas, con el objetivo de recopilar información que sea útil para la implementación del componente cumpliendo con las siguientes características:

- Software libre que permita obtener el acceso a las funcionalidades.
- Integración con el subsistema Capital humano.
- Acatamiento de las normas cubanas encargadas de regular el sistema de gestión del Pago por resultados.
- Acceso a la documentación del sistema.

#### **1.4.1 ASSETS NS**

Es un Sistema de Gestión Integral creado por Assets S.A.: una compañía de desarrollo de aplicaciones que posee representación en Cuba. Está bastante adaptado a las necesidades del país y está siendo utilizado en las Universidades. Assets NS es una aplicación cliente-servidor programada en Visual Basic 6.0 y Microsoft SQL Server 2000, este sistema no permite acceso a la documentación, y para poder acceder a ella se tiene que pagar por el concepto de licencia de software (6).

#### **1.4.2 SAGE MAS 500**

Es un sistema que ofrece soluciones integradas que automatizan todas las áreas de administración de empresas (7). Sage MAS 500, está programada en Visual Basic 6, es software propietario por lo cual no se tiene acceso al código ni a la documentación del mismo y no cumple con las normas cubanas (8).

#### **1.4.3 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 (9). El sistema SAP no es un software libre, por lo que no se puede acceder a su documentación al menos que se pague por la licencia del mismo y tampoco cumple con las normas cubanas, pero el mismo se puede pagar para que le hagan modificaciones y configuraciones para adaptarlo a las condiciones del país.

#### **1.4.4 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

## Capítulo 1: Fundamentación teórica

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). El sistema esta implementado en Delphi que es una tecnología privativa, es comercializado por Desoft<sup>3</sup>, no es multiplataforma, no se puede acceder a la documentación ni al manual de usuario, al no ser que se compre la licencia del software y sí cumple con las norma cubanas (10).

### 1.4.5 Rodas XXI

El Sistema Integral Económico Administrativo RODAS XXI es un sistema multiempresa desarrollado por la entidad CITMATEL, cuenta actualmente con seis módulos: 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 (11).

El sistema contable Rodas XXI tiene funcionalidades muy útiles pero está desarrollado para plataforma de software propietario, además es privativo por lo que no se tiene acceso al código, es comercial y no se tiene acceso a la documentación ni al manual de usuario, al no ser que se compre la licencia del software y sí cumple con las normas cubanas.

### Resultados del estudio de los sistemas informáticos

Después del estudio de los sistemas anteriores se refleja en la siguiente tabla los indicadores que estos deben cumplir para ser utilizados, los mismos aparecen por las columnas y por las filas los sistemas analizados, ver Tabla 1.

**Tabla 1:** Resultados del estudio de los sistemas informáticos.

<b>Sistemas Informáticos</b> <b>Características</b>	Sage Mas 500	Rodas XXI	Assets	SAP	Versat Sarasola
Software libre que permita obtener el acceso a las funcionalidades.	No	No	No	No	No
Acatamiento de las normas cubanas	No	Si	Si	No	Si

<sup>3</sup> Desoft Empresa encargada de ofrecer Soluciones Integrales en Tecnologías de la Información para la informatización de la sociedad cubana.

## Capítulo 1: Fundamentación teórica

encargadas de regular el sistema de gestión del Pago por resultados.					
Integración con el subsistema Capital humano.	No	No	No	No	No
Acceso a la documentación del sistema.	No	No	No	No	No

Luego de haber estudiado y analizado los indicadores propuestos para los sistemas seleccionados se puede concluir que ninguno cumple con los mismos, por lo que es más factible crear un sistema que responda a la norma cubana, que se tenga acceso a la documentación, que sea un software libre y que permita la integración con el subsistema Capital humano.

### 1.5 Modelo de desarrollo del software

Este trabajo transita fundamentalmente por las disciplinas de implementación y pruebas internas del modelo de desarrollo propuesto por el centro ver Figura 1 (12).



Figura 1: Ciclo de vida de proyectos CEIGE.

## *Capítulo 1: Fundamentación teórica*

### **1.5.1 Descripción general de la disciplina implementación**

La disciplina de implementación, a partir de lo establecido en el modelo de desarrollo dicta que “A partir de los resultados del análisis y diseño se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ejecutables y similares. Al reutilizar componentes de software ya implementados se lleva a cabo el desarrollo necesario para ajustar a los requisitos actuales y posteriormente realizar la integración de los componentes” (12).

### **1.5.2 Descripción general de la disciplina Pruebas internas**

La disciplina de prueba, a partir de lo establecido en el modelo de desarrollo dicta que “Durante esta disciplina se desarrollan las pruebas del grupo de calidad del centro verificando el resultado de la implementación. Permite identificar posibles errores en la documentación y el software, es decir requisitos que el producto debería cumplir y que aún no los cumple” (12).

- No conformidades.
- Acta de liberación.

### **Conclusiones parciales**

- Se definen las herramientas y tecnologías que se utilizan para desarrollar el componente Pago por resultados las cuales están propuestas por el marco de trabajo Sauxe.
- Se especificaron las disciplinas por las que va a transitar dicho componente.
- La solución a implementar responderá a la norma cubana para el Pago por resultados.
- Para validar la solución se van realizar las pruebas de caja blanca y caja negra.

### CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

#### 2.1 Introducción

El presente capítulo parte de la valoración de la tesis de Análisis y diseño del componente Pago por resultado y sobre la base de las deficiencias se agregarán o modificarán los artefactos necesarios para comenzar la implementación. Esto permite precisar un vocabulario común entre diseñadores y hacer el trabajo más entendible para los desarrolladores.

#### 2.2 Funcionalidades que debe cumplir el componente

*Las funcionalidades que se van a implementar están descritas en el capítulo 2 de la tesis de Análisis y diseño del proceso Pago por resultados (5) , ver Anexo 1.*

Los requisitos no funcionales que propone el centro CEIGE se encuentran definidos en el documento Descripción de requerimientos suplementarios. Proyecto Capital Humano v1.0 (13), estos son:

- Funcionalidad
- Usabilidad
- Eficiencia
- Mantenibilidad
- Contrastabilidad

#### 2.3 Patrones de diseño

Además de los patrones especificados en la tesis de Análisis y diseño del proceso Pago por resultados, se incluye en esta investigación los patrones *Gang Of Four*(GoF), 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. Los patrones utilizados son los siguientes:

**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. Ejemplo donde se aplica este patrón es en los Servicios, ya que la relación existente entre estos y las clases *Controller*(Controladoras) posibilitan utilizar métodos de otros componentes que se encuentran tanto dentro como fuera del subsistema Capital Humano (14).

**Cadena de responsabilidad:** Permite establecer la línea que deben llevar los mensajes para que los objetos realicen la tarea indicada. La cadena de responsabilidad 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 (15). Este patrón se refleja en las clases controladoras y las del modelo donde se establece la una cadena a seguir para las peticiones de la clase controladora al modelo y del modelo a la entidad. Ejemplos: *GestionarpenalizacionController* y *DatPenalizacionModel*.

### 2.4 Diseño del sistema

En la tesis de análisis y diseño no se realizaron todos los diagramas de clase del diseño, como parte de la solución para este trabajo se realizaron los restantes diagramas, ver **Anexo 2**.

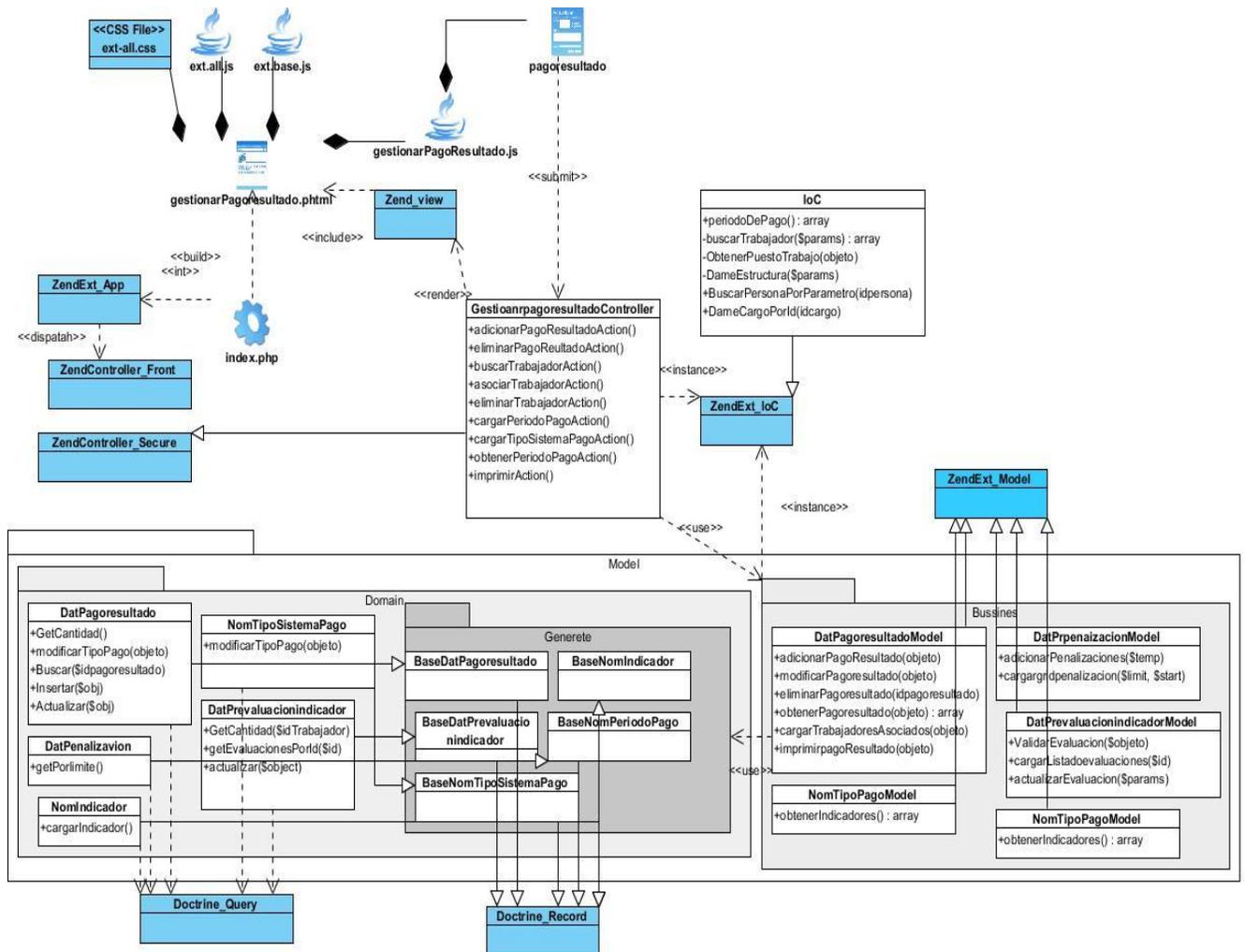
#### 2.4.1 Diagrama de Clases

En esta solución se generaron 5 diagramas de clase que responden a los escenarios

- Gestionar Indicadores.
- Gestionar tipos de sistema de pago.
- Procesar evaluación.
- Gestionar penalización.
- Administrar nóminas.

En la Figura 2 se muestra el diagrama de clases del diseño del escenario Gestionar Pago por resultados, en el cual se evidencian las principales clases, operaciones y relaciones necesarias en la ejecución de los requisitos funcionales. Los diagramas de clase del diseño aplican el patrón arquitectónico MVC, el cual separa en tres capas, la interfaz, la lógica del negocio y los datos.

Las clases *gestionarpagoresultado.js* y *gestionarpagoresultado.phtml* representan la capa arquitectónica de presentación. La clase *GestionarpagoresultadoControlle.php* se encarga del manejo de la comunicación entre la vista y el modelo, la clase *DatPagoresultadoModel.php* es la encargada de implementar las funcionalidades que garantizan el cumplimiento de los requisitos identificados; encargadas del acceso a los datos se encuentran *DatPagoresultado.php*, así como las clases Base de las cuales extienden las anteriormente mencionadas.

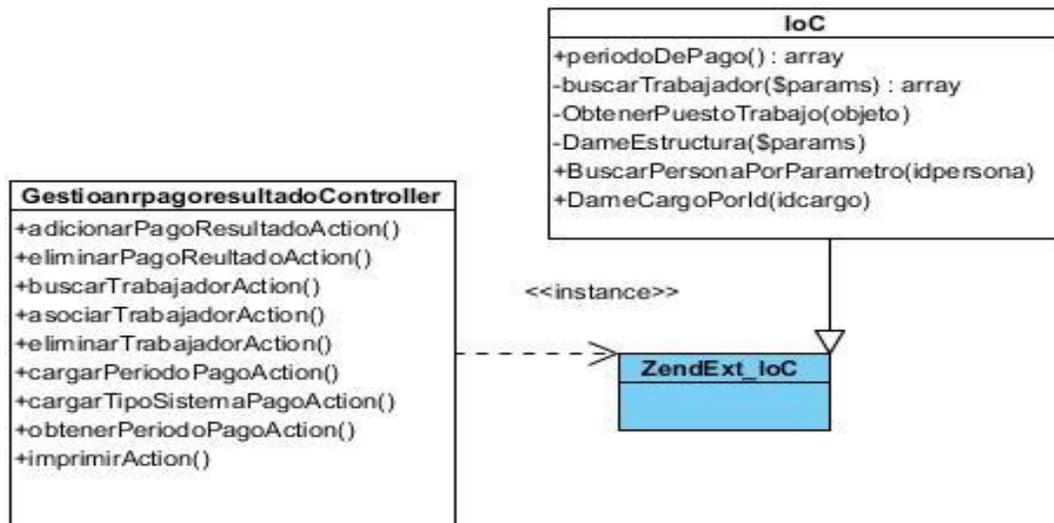


**Figura 2:** Gestionar sistema de Pago por resultados.

## 2.4.2 Patrones utilizados

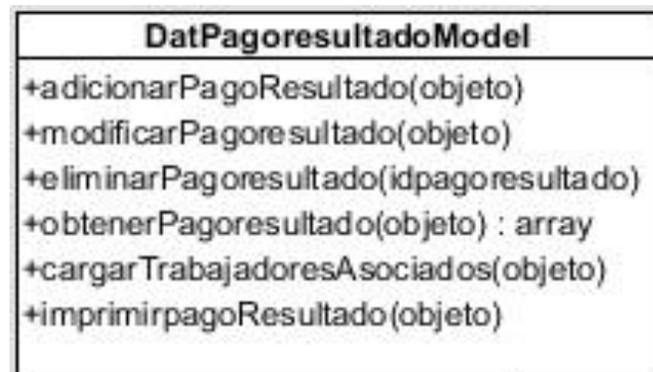
Para contribuir a que el sistema sea más flexible y robusto se utilizaron los patrones de diseño siguientes:

El patrón fachada se evidencia en la creación y aplicación del control de inversión (IoC), debido a las necesidades de integrar los distintos componentes que conforman Cedrux se hizo necesario la creación de una clase fachada donde se publicaran todos los servicios con el objetivo de mantener la comunicación entre los distintos componentes, este patrón se evidencia en la Figura 3.



**Figura 3:** Patrón Fachada.

El patrón experto es los encargados de efectuar operaciones específicas de acuerdo a la información que manejan, por ejemplo en la clase `DatPagoresultadoModel.php` es la encargada de efectuar todas las operaciones que tienen que ver con la gestión del pago por resultados, este patrón se evidencia en la Figura 4.



**Figura 4:** Patrón Experto.

El patrón controlador es el que efectúa las operaciones necesarias para dar cumplimiento a dichos requisitos, ejemplo de esto es la clase controladora `Gestionarpagoporrresultadocontroller.php` se encarga de darle cumplimiento a los requisitos relacionados con la gestión del pago por resultados, este patrón se evidencia en la Figura 5.

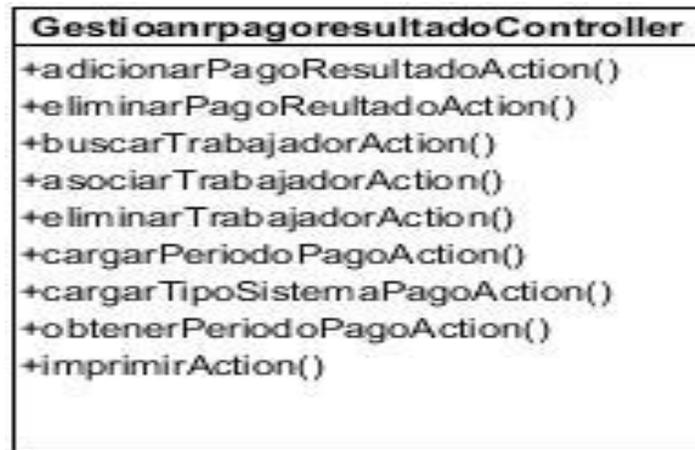


Figura 5: Patrón Controlador.

Patrón Creador se aplica en las clases pertenecientes al dominio, las cuales son encargadas de crear objetos de tipo doctrine query para poder tener acceso a la información almacenada a nivel de datos, este patrón se evidencia en la Figura 6.

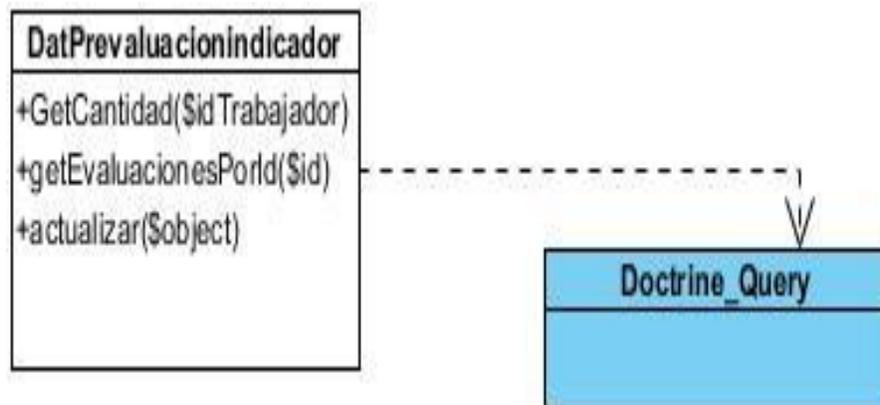


Figura 6: Patrón Creador.

### 2.4.2 Descripción de las funcionalidades implementadas en el componente Pago por resultado

Las restantes funcionalidades se encuentran en el **Anexo 3**.

Tabla 2: Clase controladora *GestionarpagoresultadoControlle.php*.

<b>Nombre: GestionarpagoresultadoControlller</b>	
<b>Tipo de clases: Controladora</b>	
<b>Para cada responsabilidad</b>	
<b>Atributos</b>	<b>Tipo</b>

## Capítulo 2: Propuesta de solución

adicionarPagoResultadoAction()	Este método recibe los datos de la interfaz y hace un llamado al método Guardar(\$obj) de la clase <i>DatPagoresultadoModel.php</i> .
modificarPagoResultadoAction()	Este método recibe los datos de la interfaz y hace un llamado al método Modificar(\$obj) de la clase <i>DatPagoresultadoModel.php</i> .
eliminarPagoReultadoAction()	Este método recibe por parámetros el id del pago por resultados que se desea eliminar y hace un llamado al método Eliminar(\$obj) de la clase <i>DatPagoresultadoModel.php</i> del dominio.
obtenerIndicadoresAction	Se encarga de dar todos los indicadores que hay en la clase Gestionar indicadores.
cargarcomboindicadoresAction()	Este método recibe por parámetros el id de los indicadores que se desean cargar y se hace un llamado al método cargargridindicador() de la clase <i>NomIndicadoresModel.php</i> .
cargarcomboTipoPagoAction()	Este método recibe por parámetros el id de los tipos de pago que se desean cargar y se hace un llamado al método cargarsistemaPago() de la clase <i>NomTipopagoresultadoModel.php</i> .

**Tabla 3:** Clase modelo *DatPagoresultadoModel.php*.

<b>Nombre: DatPagoresultadoModel</b>	
<b>Tipo de clase:</b>	
<b>Atributos</b>	<b>Tipo</b>
<b>Para cada responsabilidad</b>	
Guardar(\$DatPagoresultado)	Este método recibe por parámetros un objeto y hace un llamado al método adicionar(obj) de la clase
Modificar(\$obj)	Retorna el valor que devuelve el método actualizar (obj) de la clase del dominio <i>DatPagoresultado.php</i> se le pasan los datos de la pago por resultados que se está modificando.

## Capítulo 2: Propuesta de solución

Eliminar(\$idpagoresultado)	Este método recibe por parámetros el id del pago por resultados que se desea eliminar y hace un llamado al método Buscar(idpagoresultado) de la clase <i>DatPagoresultado.php</i> del dominio y elimina el pago por resultados una vez encontrada en la
cargarPagoResultado(\$limit, \$start)	Este método a través del método GetPorLimite(limit, start, tipo) y GetTodos() de la clase <i>DatPagoresultado.php</i> del dominio, obtiene todos los pagos por resultados que se encuentran en la base de datos según el tipo deseado.

**Tabla 4:** Clase del dominio *DatPagoresultado.php*.

<b>Nombre:</b> DatPagoresultado	
<b>Tipo de clase:</b> Modelo (Domain)	
<b>Para cada responsabilidad</b>	
<b>Nombre del método</b>	<b>Descripción</b>
Buscar(idindicador)	Este método busca en la base de datos el pago por resultados que coincida con el identificador que se le pasa por parámetro.
GetTodos()	Este método devuelve todos los pagos por resultados que se encuentran en la base de datos.
GetPorLimite(limite = 20, inicio = 0)	Devuelve todos los pagos por resultados de la base de datos de acuerdo a un rango.
adicionar(obj)	Este método recibe como parámetro un objeto insertándolo en la base de datos.
actualizar(obj)	Recibe como parámetro un objeto actualizándolo en la base de datos.

### **2.4.3 Modelo de datos**

El modelo de datos propuesto en la tesis de análisis y diseño, fue modificado para facilitar el diseño de la base de datos y mostrar con más exactitud los datos que contendrán dichas tablas, el modelo de datos antes realizado no cumplía con todas las relaciones de clases que se necesitan para la implementación.

El modelo de datos propuesto en la solución cuenta con un total de 16 tablas, de ellas 6 son propias del proceso y hay 4 que no pertenecen directamente al componente, pero son imprescindibles para llevar a cabo el proceso Pago por resultados, dichas tablas son nom\_periodo\_pago donde se almacenan los períodos de pago de la empresa, dat\_trabajador donde se registran los datos de los trabajadores, dat\_estructuraop donde se registran las áreas a las que pertenece un trabajador y dat\_cargo que registra los cargos que tiene cada trabajador. El modelo contiene 6 nomencladores encargados de gestionar conceptos específicos del negocio ya predefinido, ejemplo de esto es la tabla "nom\_indicadores" que gestiona los distintos indicadores del componente Pago por resultados que están definidos en la entidad. Las restantes tablas se encargan de gestionar los datos que son necesarios tener registrados, por ejemplo, las tabla "dat\_pago\_resultado", es la encargada de almacenar todos los datos de los sistemas de pago que puedan existir en una entidad como el tipo de sistema de pago y los trabajadores abarcados. Esta tabla es una de las fundamentales del proceso pues contiene los datos y las relaciones necesarias para evaluar el trabajador y realizar la nómina de Pago por resultados. A continuación se muestra el diagrama del modelo de datos para el proceso de Pago por resultados ver Figura 7. Las tablas de color gris son la que no pertenecen al componente Pago por resultado.

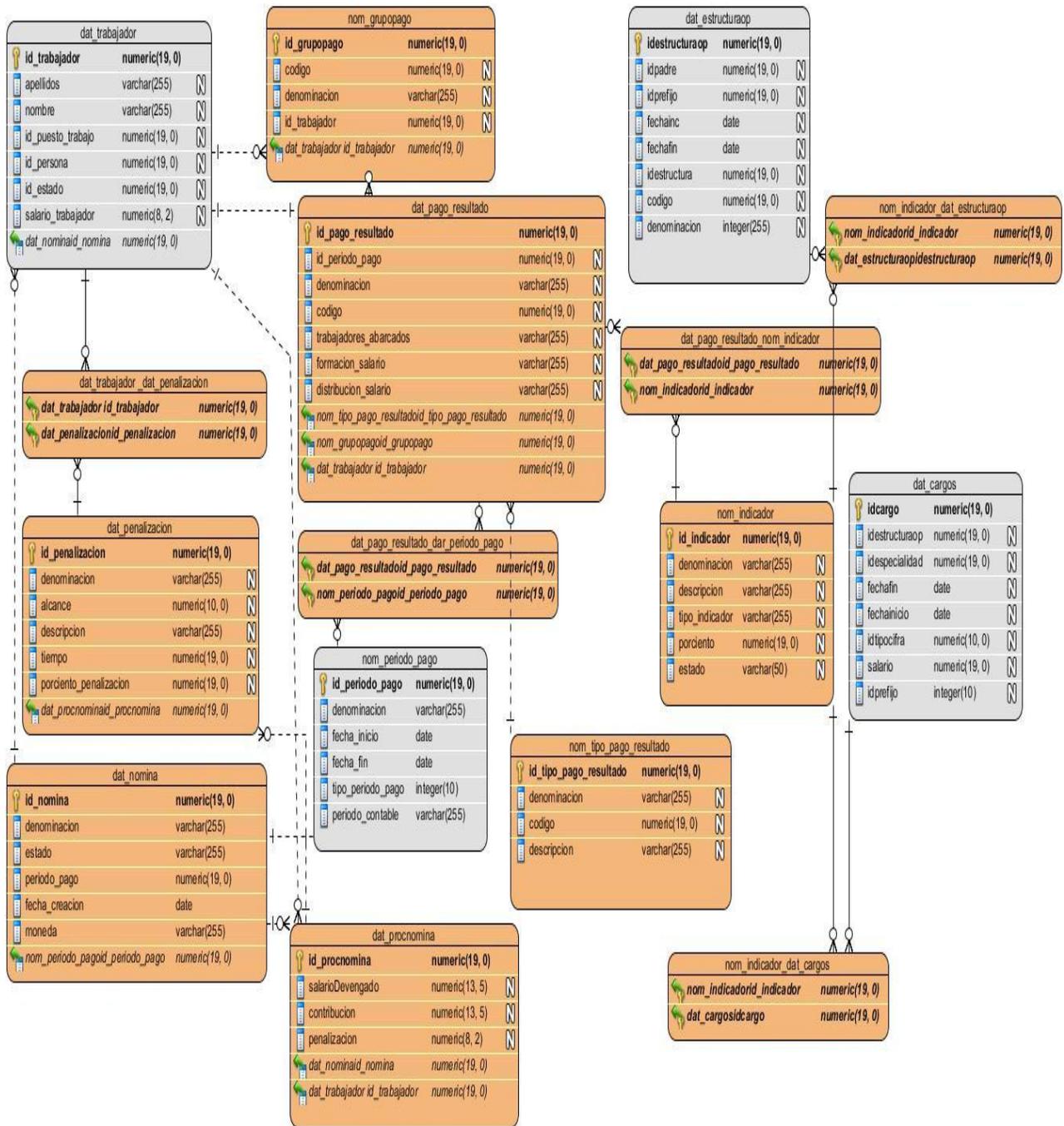
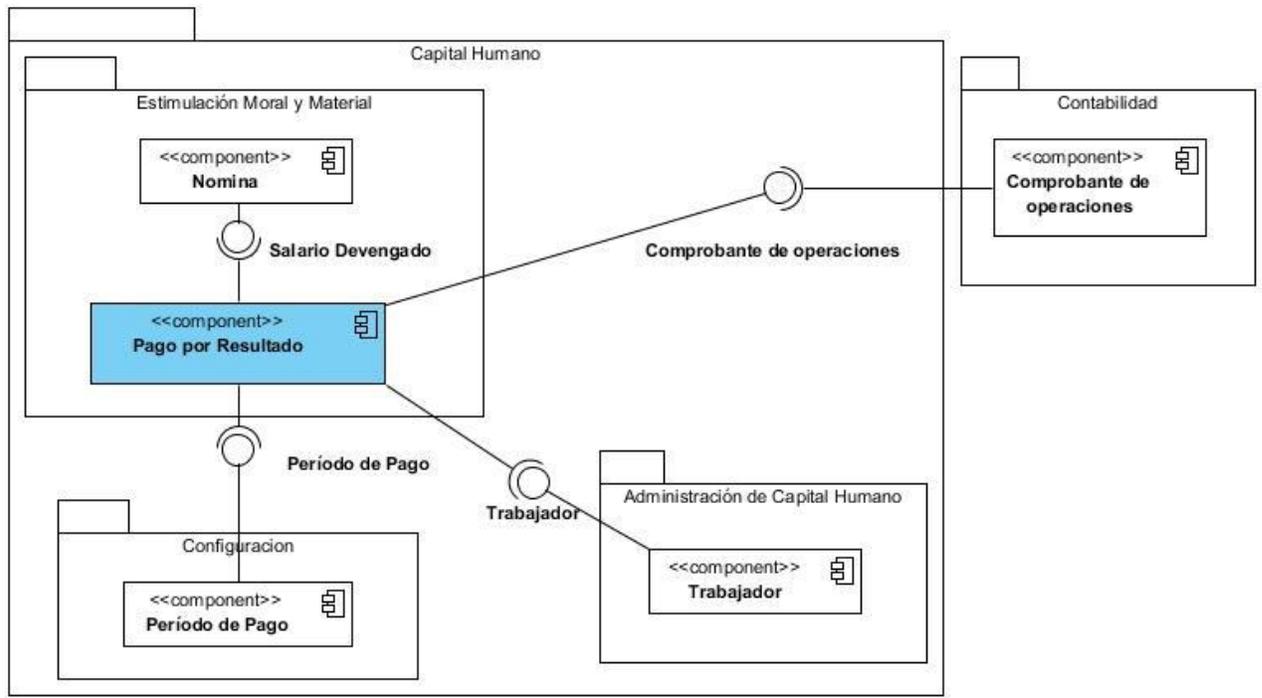


Figura 7: Modelo de datos.

## 2.4.4 Diagrama de componente

El diagrama de componente realizado en la tesis de análisis y diseño fue modificado, en el mismo se reflejaba un componente que no fue utilizado en la implementación del proceso Pago por resultados. A continuación se representa el diagrama de componentes relacionado con el proceso de Pago por resultados Figura 8.



**Figura 8:** Diagrama de componentes.

El componente Pago por resultados utiliza varios servicios dados por otros componentes que se encuentran dentro del subsistema Capital Humano. A continuación se muestra una descripción de todos los servicios que necesita el componente Pago por resultados.

**Tabla 5:** Servicios que utiliza el componente Pago por resultados.

Servicios que utiliza el componente Pago por resultados				
Subsistema	Módulo	Componente	Servicio	Descripción
Capital Humano	Configuración	Periodo de pago	Periodo de pago	Servicio que brinda las fechas en que se debe de realizar las evaluaciones parciales de cada trabajador.
Capital Humano	Administración del Capital humano	Trabajador	Trabajador	Servicio que muestra todos los datos del trabajador, así como todas las áreas de trabajo que existen en la empresa.
Capital Humano	Organización de trabajo	Puesto de trabajo	Puesto de trabajo	Servicio que muestra los puestos de trabajo de un

## Capítulo 2: Propuesta de solución

				trabajador.
--	--	--	--	-------------

El componente Pago por resultados brinda varios servicios a otros componentes que se encuentran dentro del subsistema Capital Humano. A continuación se muestra una descripción de todos los servicios que el mismo brinda.

**Tabla 6:** Servicios que brinda el componente Pago por resultados.

Servicios que brinda el componente Pago por resultados				
Subsistema	Módulo	Componente	Servicio	Descripción
Capital Humano	Estimulación Moral y Material	Nómina	Registro de evaluaciones	Servicio que brinda todas las evaluaciones realizadas a los trabajadores en un periodo contable dado.
Capital Humano	Estimulación Moral y Material	Nómina	Registro de penalizaciones	Servicio que brinda todas las penalizaciones realizadas a los trabajadores en un periodo contable dado.

### 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 componente pago por resultados se encuentra en el macro proceso Estimulación Moral y Material corresponde al subsistema Capital humano el cual pertenece a Cedrux, para desplegar dicho componente se tiene que desplegar el subsistema Capital humano. A continuación se muestran los requerimientos de software necesarios para el despliegue de la aplicación y los posibles escenarios.

#### Servidores

##### Servidor de Aplicaciones Web

- Sistema Operativo: Ubuntu Server
- Servidor Web: Apache v2.2.9
- Librerías Adicionales: PHP v5.2.6

##### Servidor de Base de Datos

- Sistema Operativo: Ubuntu Server
- Sistema Gestor de Base de Datos: PostgreSQL v8.3

### Servidor de Clientes Ligeros

- Sistema Operativo: Nova Server
- Navegador Web: Mozilla Firefox v2.17 o superior
- Herramientas Ofimáticas
- Visualizador de ficheros pdf.
- Herramienta de administración de clientes ligeros

### Clientes

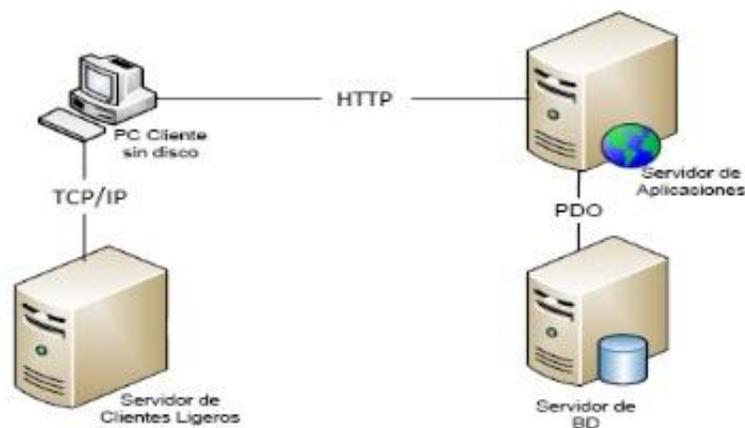
#### PC Cliente con disco duro

- Sistema Operativo: Linux o Windows
- Navegador Web: Mozilla Firefox v2.17 o superior
- Herramientas Ofimáticas
- Visualizador de ficheros pdf.

#### PC Cliente sin disco duro

- Todo se instala en el servidor de clientes ligeros

La Figura 9 y la Figura 10 muestran los diagramas de despliegue tanto para los clientes con disco o sin disco:



**Figura 9:** Diagrama de despliegue de escenario para PC cliente sin disco.



**Figura 10:** Diagrama de despliegue de escenario para PC cliente con disco.

### **Conclusiones parciales**

- Se adicionaron y modificaron algunos artefactos que no estaban especificados en la tesis de Análisis y diseño del componente Pago por Resultado.
- Quedó plasmada la utilización de los patrones del diseño en la implementación del componente.
- Se mostraron los posibles escenarios en los cuales puede ser desplegado el subsistema Capital humano a través del diagrama de despliegue, así como los requerimientos de software necesarios para un correcto funcionamiento de este.

### CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

#### 3.1 Introducción

En el presente capítulo se describe la implementación del sistema, se especifica la estructura del componente, se describen los servicios que brinda y consume. Se especifican las pruebas de software que son usadas para la validación y comprobación de la calidad del sistema desarrollado.

#### 3.2 Implementación

##### 3.2.1 Estructura del componente Pago por resultados

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, Pago por resultados se encuentra contenida en la carpeta Capital humano/Estimulación moral y material como se observa en la Figura 11.

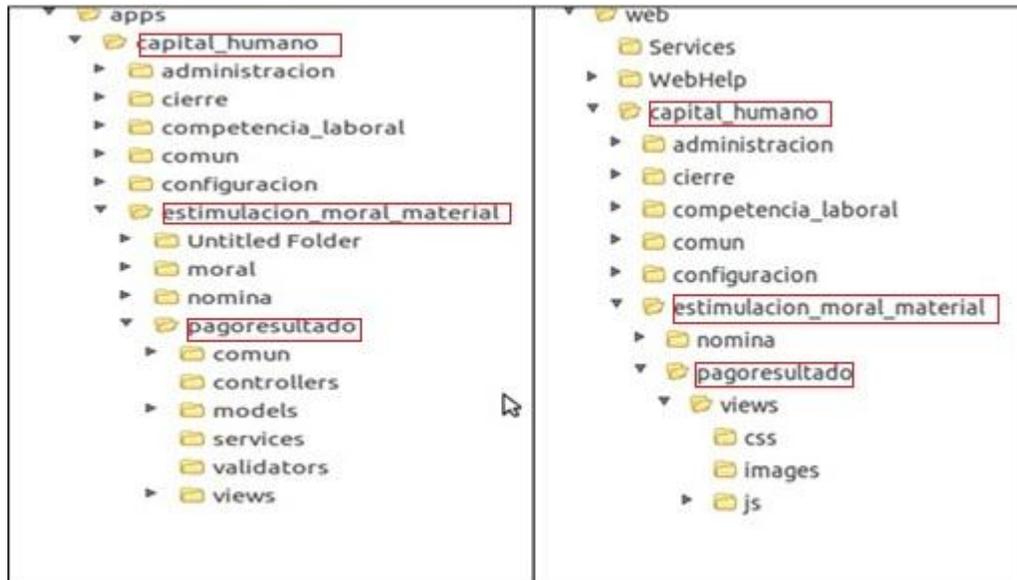


Figura 11: Contenido de las carpetas apps y web para Capital humano.

##### Descripción de la carpeta apps del componente Pago por resultados

**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.

## Capítulo 3: Implementación y prueba

**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 una determinada funcionalidad sea ejecutada.

**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.

### Descripción de la carpeta web del componente Pago por resultados

**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).

#### 3.2.2 Servicios utilizados

Todo el código dentro de un mismo componente utiliza llamadas a funcionalidades o eventos de forma directa, sin embargo, para lograr comunicación entre diferentes componentes, se utiliza el patrón inversión de control (IoC). 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.

#### Servicios que consume el componente Pago por resultados

- **BuscarTrabajador:** Se obtiene el trabajador a partir del parámetro enviado.
- **BuscarPersonaPorParam:** Se obtienen los datos de la persona a partir del parámetro enviado.
- **periodosDePago:** Se obtienen todos los periodos de pago actual.
- **DameEstructurasInternas:** Se obtiene todas las áreas de una estructura.

#### Servicios que brinda el componente Pago por resultados

- **obtenerEvaluacion:** Muestra las evaluaciones realizadas al trabajador.
- **obtenerPenalizacion:** Muestra las penalizaciones hechas a un trabajador.

En la Figura 12 se muestra un fragmento correspondiente al servicio interno *obtenerPenalizacion* que brinda el componente Pago por resultados.

```
<obtenerPenalizacion reference="">
  <inyector clase="pagoresultado" metodo="obtenerPenalizacion" />
  <prototipo>
    <parametro nombre="params" tipo="stdClass"/>
    <resultado tipo="array" />
  </prototipo>
</obtenerPenalizacion>
```

**Figura 12:** Contenido del fichero IoC.xml interno.

La Figura 13 muestra un fragmento de un servicio externo *DameEstructurasInternas* del subsistema Capital humano.

```
<DameEstructurasInternas>
  <inyector clase="EstructuraService" metodo="DameEstructurasInternas"/>
  <prototipo>
    <parametro nombre="idEstructura" tipo="enterospos"/>
    <parametro nombre="soloRaices" tipo="bool"/>
    <resultado tipo="array"/>
  </prototipo>
</DameEstructurasInternas>
```

**Figura 13:** Contenido del fichero IoC.xml externo.

### 3.2.3 Funcionalidades relevantes

Se reutilizaron y modificaron funcionalidades del componente nómina para realizar la implementación del proceso Pago por resultados, entre estas:

Procesar Nómina: Esta hace una llamada a la funcionalidad *calcularPagoResultado*, también llama a las funcionalidades *salarioescala*, *contribucion* y *retenciones*, y hace un uso de servicios *obtenerEvaluacion*, *obtenerPenalizacion*, calculando la nómina de Pago por resultados con complejidad algorítmica  $O(n^2)$ , ver Figura 14.

```

public function calcularPagoResultado($trabajador, $idnomina,$idprenomina,$idperiodopago){
    $params->idnomina = $idnomina;    O(1)
    $params->idtrabajador = $trabajador; O(1)
    $procnom = new DatNprocnomina(); O(1)
    $procesamiento = $procnom ->buscarProcNomina($params); O(n)
    $params->idprocnomina=$procesamiento->idprocnomina; O(1)
    $salarioBruto=$procesamiento->salarioescala-$procesamiento->contribucion; O(1)
    $penalizacion= $this->pIntegrator->pago_resultado->obtenerPenalizacion();O(n)^2
    $cumplimiento= $this->pIntegrator->pago_resultado->obtenerEvaluacion(); O(1)
    $sumaCumplimiento=$this->sumaCumplimiento($cumplimiento,$trabajador); O(n)
    $sumaPenalizacion=$this->sumaPenalizacion($penalizacion,$trabajador); O(n)
    $valorrealpenalizacion=$salarioBruto*($sumaPenalizacion/100); O(1)
    $valorrealcumplimiento=($salarioBruto-$valorrealpenalizacion)*($sumaCumplimiento/100); O(1)
    $sn=$salarioBruto-$valorrealpenalizacion+$valorrealcumplimiento; O(1)
    $salcobrar=$sn-$procesamiento->retencion;O(1)
    if($salcobrar<0){ O(1)
        $salcobrar=0; O(1)
    }
    $params->pagoresultado=$salcobrar; O(1)
    $params->salarioNeto=$sn; O(1)
    $params->salarioBruto=$salarioBruto; O(1)
    $params->salarioscobrar=$salcobrar; O(1)
    $result=$this->ActualizarCampos($params); O(n)
    return $result; O(1)
}

```

**Figura 14:** Complejidad algorítmica de la funcionalidad calcular el pago por resultados.

El pago por resultados se le calcula al salario escala en base a la evaluación de los indicadores definidos en la entidad y las penalizaciones aplicadas a cada trabajador, teniendo en cuenta además la contribución especial y las retenciones. A continuación el desglose de la ecuación utilizada para el procesamiento del pago por resultados.

$$SB^4 = SE^5 - CE^6$$

$$PS^7 = SB * \sum P^8 / 100$$

$$CS^9 = (SB - PS) * \sum C^{10} / 100$$

<sup>4</sup> SB: Salario Bruto

<sup>5</sup> SE: Salario escala

<sup>6</sup> CE: Contribución especial

<sup>7</sup> PS: Penalización del salario

<sup>8</sup> P: Penalización

<sup>9</sup> CS: Cumplimiento del salario

$$SN^{11} = SB - PS + CS$$

$$SC^{12} = SN - R^{13}$$

Contabilizar Nómina: Se generan los comprobantes de operaciones, se realizan los asientos contables, se envían las nóminas, ver Figura 15.

```
function contabilizarNominaAction() {
    $params->idperiodo=$this->_request->getPost('idperiodo');
    $params->cant=$this->_request->getPost('cant');
    $params->estado=$this->_request->getPost('estado');
    $params->progreso=$this->_request->getPost('progreso');
    $params->cantidad=$this->_request->getPost('cantidad');

    $params->idestructura = $this->idestructura = $this->global->Estructura->idestructura;
    $params->idsubsisistema = $this->idsubsisistema = $this->global->Subsisistema->idsubsisistema;
    $params->fechacontable = $this->fechadelsubsisistema = $this->global->FechaContable->fecha;

    echo json_encode($this->nomina->contabilizarNominas($params));
}
```

**Figura 15:** Funcionalidad contabilizar Nómina.

El método *contabilizarNominaAction* hace una llamada al *contabilizarNominas* de la clase modelo *DatNnominaModel.php*, para poder contabilizar la nómina se definen una serie de pasos:

Chequear que estén configurados los conceptos de pago definidos en la entidad.

Conformar comprobante: Aquí se conforman los comprobantes (1 comprobante por cada cuenta afectada en el proceso de contabilización de la nómina).

Contabilizar la nómina, se invoca un servicio definido en contabilidad para contabilizar las nóminas de Capital humano, se le pasa como parámetro el id del usuario, los comprobantes conformados, el período contable, el id del subsistema y el id estructura.

Además se reutilizaron otras funcionalidades básicas del componente Nómina entre las que se encuentran: adicionar, modificar y eliminar nómina y tipo de nómina respectivamente.

Entre las funcionalidades desarrolladas en el componente Pago por resultados se encuentran:

En el escenario Gestionar indicador se cuenta con las funcionalidades asociar área y cargo, donde se obtiene el cargo de un trabajador a través del área en la que trabaja, además se valida que el área y el cargo no se le asocie al mismo indicador dos veces Figura 16, cuenta también con las

---

<sup>10</sup> C: Cumplimiento

<sup>11</sup> SN: Salario Net

<sup>12</sup> SC: Salario a cobrar

<sup>13</sup> R : Retenciones

## Capítulo 3: Implementación y prueba

funcionalidades, modificar y eliminar indicador, esta última no se puede realizar si tiene al menos un área o un cargo asociado.

```
function ValidarAsociacion($params) {
    $existencia = 0;
    if($params->tipo=='area'){
        $IA= new DatPrindicadorarea();
        $result = $IA->GetTodos();
        for ($i = 0; $i < count($result); $i++) {
            if ($result[$i][idindicador] === $params->idindicador
                &&($result[$i][idarea]=== $params->idarea)) {
                $existencia = 2;
            }
        }
    } else{
        $IC= new DatPrindicadorcargo();
        $result = $IC->GetTodos();

        for ($j = 0; $j < count($result); $j++) {
            if ($result[$j][idindicador] === $params->idindicador
                &&($result[$j][idcargo]=== $params->idcargo)) {
                $existencia = 2;
            }
        }
    }
    return $existencia;
}
```

**Figura 16:** Validar áreas y cargos que no estén repetidos en un mismo indicador.

En el escenario Gestionar penalización se cuenta con las funcionalidades asociar y desasociar trabajador, donde se obtienen los trabajadores que se van a evaluar según la penalización obtenida y en dependencia del indicador que incumplieron, cuenta también con las funcionalidades, modificar y eliminar penalización, esta última funcionalidad no se puede realizar si tiene al menos un trabajador asociado, ver Figura 17.

```
function asociarTrabajadorAction(){
    $params->idpenalizacion = $this->_request->getPost ('idpenalizacion');
    //print_r($params->idgrupopago ); die;
    $params->idtrabajador = $this->_request->getPost ('idtrabajador');
    $Asociacion= new DatTrabajadorpenalizacionModel();

    return $Asociacion->asociarTrabajadores($params);
}
```

**Figura 17:** Asociar Trabajador para ser penalizado.

En el escenario Procesar evaluación las funcionalidades más significativas son *CargarevaluacionesindicadoresAction* en la cual se evalúa el trabajador por el cumplimiento obtenido

según el indicador, e *InsertarevaluacionesindicadoresAction* la cual permite insertar las evaluaciones de los trabajadores al sistema, ver Figura 18, estas pueden ser modificadas o eliminadas.

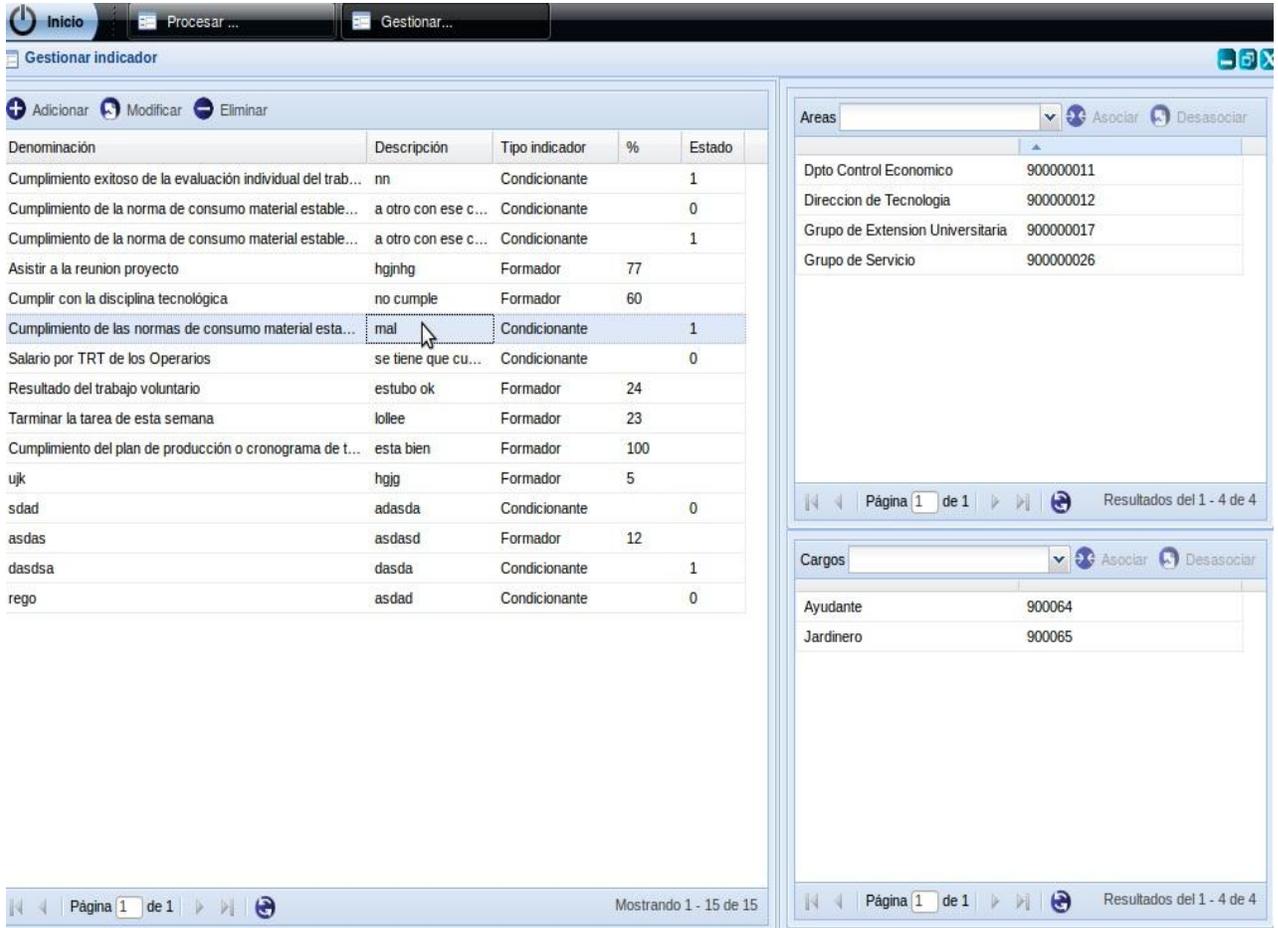
```
function InsertarevaluacionesindicadoresAction(){
    $evaModel=new DatPrevaluacionindicadorModel();
    $datos = json_decode(stripslashes($this->_request->getPost('dato')));
    $vacio=false;
    for ($i= 0; $i < count($datos); $i++){
        if($datos[$i]->cumplimiento==null || $datos[$i]->recomendaciones==null){
            $vacio=true ;
            break;
        }
    }
    if($vacio){ echo 3;}
    else{
        $objeto->idtrabajador = $this->_request->getPost('idtrabajador');
        $objeto->fechaevaluacion= $this->_request->getPost('fecha');
        $cantidadtotal= $this->_request->getPost('cantidadtotal');
        $evaluados= count($datos);
        if($evaluados!=$cantidadtotal){
            $result=4;
        } else{
            $result=$evaModel->Insertarevaluacionindicador($objeto, $datos);
        }
    }
    echo $result;
}
```

**Figura 18:** Insertar evaluación de los indicadores.

### 3.2.4 Flujo de escenarios para realizar el Pago por resultados

Para realizar el pago por resultados en una entidad se deben configurar una serie de nomencladores, y seguir una serie de pasos que se describen a continuación:

**1: Gestionar indicadores** En el escenario Gestiona indicador, se adicionan los indicadores, una vez adicionados los mismos se le asocian las áreas y los cargos, también permite modificar y eliminar los indicadores y desasociar las áreas y los cargos que se le han asignado a los mismos, ver Figura 19.



**Figura 19:** Escenario Gestionar indicadores.

**2: Gestionar penalización:** En el escenario Gestionar penalización se adicionan las penalizaciones que se le realizan a los trabajadores por incumplimiento, una vez adicionadas estas penalizaciones, se le asocian los trabajadores que se van a penalizar por el tipo de indicador, ver Figura 20.

## Capítulo 3: Implementación y prueba

Inicio Procesar ... Gestionar... Gestionar...

Gestionar penalización

+ Adicionar Modificar - Eliminar

Denominación	Alcance	Tiempo	Penalización	Por ciento	Descripción
Reuniones	10	2	Condicionante	2	no asistio
Ultimo	9	9	Formador	44	cdvver
Reuniones	9	2	Perdida	34	Se borro las funciones del marco de trabajo del subsistema capital ...
CedruX	3	1	Formador	50	Debe trabajar mas en el sistema
caso	2	2	Formador	2	2
Ultim	1	22	Formador	23	se esfuerza pero le falta mucho
lk	98	9	Formador	9	klkpkp

Página 1 de 1 Mostrando 1 - 7 de 7

Asociar Trabajador Desasociar Trabajador

Nombre	Primer Apellido	Segundo Apellido	Puesto de Trabajo
ESPERANZA	CASTRO	GARCIA	Cajero
DAYAN	CANOVA	RAMIREZ	Jefe
Arnolis	Salgueiro	Arzuaga	Informatico
DAYAN	CANOVA	RAMIREZ	Jefe

Página 1 de 1 Mostrando 1 - 4 de 4

**Figura 20:** Escenario Gestionar penalización.

**3: Procesar evaluación:** En el escenario Procesar evaluaciones, se evalúan los trabajadores según el cumplimiento de sus indicadores Figura 21. Luego de realizar las evaluaciones se pueden modificar y eliminar ver Figura 22.

## Capítulo 3: Implementación y prueba

Fecha: 03/06/2013 Evaluaciones

Cumplimiento %	Denominación	Tipo indicador	Sugerencias
32	Asistir a la reunion proyecto	Formador	Trabajar mas
2	Cumplimiento de la norma de consu...	Condicionante	Esforzarse
22	Cumplimiento de la norma de consu...	Condicionante	Trabajar mas
2	Cumplir con la disciplina tecnológica	Formador	Esforzarse
	Cumplimiento exitoso de la evaluació...	Condicionante	Esforzarse

Página 1 de 1 Resultados del 1 - 5 de 5

Cancelar Aplicar Aceptar

Figura 21: Escenario Evaluación de Indicadores.

Total de evaluaciones

Cumplimiento %	Fecha	Sugerencias
32	2013-06-03	Trabajar mas
2	2013-06-03	Esforzarse
22	2013-06-03	Trabajar mas
2	2013-06-03	Esforzarse
8	2013-06-03	Esforzarse
52	2013-04-30	Esforzarse
12	2013-04-30	Participar mas con el centro

Página 1 de 1 Resultados del 1 - 7 de 7

Cancelar Aplicar Aceptar

Figura 22: Escenario total de evaluaciones.

## Capítulo 3: Implementación y prueba

**4: Administrar nómina:** Este escenario es el encargado de adicionar la nómina Pago por resultados una vez adicionada se procesa y se obtiene el pago por resultado del trabajador, también permite modificar y eliminar la nómina Pago por resultados antes de ser procesada, ver Figura 23.

The screenshot displays a web-based payroll management application. At the top, there is a navigation bar with buttons for 'Inicio', 'Procesar ...', 'Gestionar...', 'Gestionar...', and 'Administr...'. Below this is a header for 'Administrar nóminas' with sub-actions: '+ Adicionar', 'Modificar', 'Eliminar', 'Procesar', and 'Imprimir'. A 'Periodo' dropdown menu is also present.

The main area contains a table of payroll entries:

Denominación	Tipo de nómina	Período de pago	Estado	Fecha
Pago resultado	Pago resultado final	Enero	Procesada	2013-06-08
Pago resultado	Pago resultado final	Enero	En Edición	2013-06-08
fgdf	nominadesempeno2	Enero	Confirmada	2013-05-30
Salario	Nómina de Salario	Enero	Confirmada	2012-06-15
Prueba	Pago resultado final	Enero	Confirmada	2013-06-08
Prueba del pago por resultado	Pago resultado final	Enero	Confirmada	2013-06-08
fdsf	Nómina de Salario	Enero	Procesada	2013-06-08
Lolito	Prueba	Enero	Procesada	2013-05-27
fgffgdgfg	Salario enero	Enero	Procesada	2013-05-29
Pago	Pago resultado final	Enero	Confirmada	2013-06-08
Locura	Prueba Interna	Enero	Procesada	2013-06-11

Below the table, there is a pagination control showing 'Página 12 de 12' and 'Resultados del 221 - 237 de 237'. A second table provides a detailed view of a payroll entry:

Exp. Interno	Nombre	1er Apellido	2do Apellido	Salario esc...	Horas trabajadas	Pagos adiciona...	Salario bruto	Contribución especial	Retenciones	Salario a cobrar	Di...	Imp...
15249	Arnolis	Salgueiro	Arzuaga	235	0	0	235	0	0	9	0	0
15244	ARITA	ABELLAs	PEREZ	235	191	0	235	0	0	235	0	0

The 'Salario a cobrar' column in the second table has a red border around the values 9 and 235. At the bottom of the interface, there is another pagination control showing 'Página 1 de 1' and 'Mostrando 1 - 2 de 2'.

**Figura 23:** Adicionar nómina pago por resultados.

**5: Revisar nómina:** Una vez procesada la nómina pago por resultados se procede a revisar dicha nómina, esta se confirma y luego se contabiliza la cual crea y envía un comprobante de operaciones a contabilidad, ver Figura 24.

## Capítulo 3: Implementación y prueba

The screenshot shows a web application window titled "Revisar nóminas". At the top, there are buttons for "Confirmar", "Cancelar", and "Contabilizar". The current period is "Enero". Below this is a table with the following data:

Denominación	Tipo de nómina	Periodo de pago	Estado	Fecha
Prueba del pago por resultado	Pago resultado final	Enero	Confirmada	2013-06-08
fdsf	Nómina de Salario	Enero	Procesada	2013-06-08
Lolito	Prueba	Enero	Procesada	2013-05-27
fgfdgdfg	Salario enero	Enero	Procesada	2013-05-29
Pago	Pago resultado final	Enero	Confirmada	2013-06-08
Locura	Prueba Interna	Enero	Procesada	2013-06-11
Yaily	Pago resultado final	Enero	Confirmada	2013-06-14

Below this table is a navigation bar showing "Página 9 de 9" and "Mostrando 161 - 167 de 167". There are search fields for "Exp. Interno", "Nombre", "1er Apellido", and "2do Apellido". Below these is another table with the following data:

Exp. Interno	Nombre	1er Apellido	2do Apellido	Salario es...	Horas trabaj...	Pagos adicio...	Salario b...	Contribución esp...	Retencio...	Di...	Sal...	Imp...
15249	Arnolis	Salgueiro	Arzuaga	235	0	0	235	0	0	0	9	0
15244	ARITA	ABELLAS	PEREZ	235	191	0	235	0	0	0	235	0

At the bottom, there is another navigation bar showing "Página 1 de 1" and "Mostrando 1 - 2 de 2".

Figura 24: Revisar Nóminas.

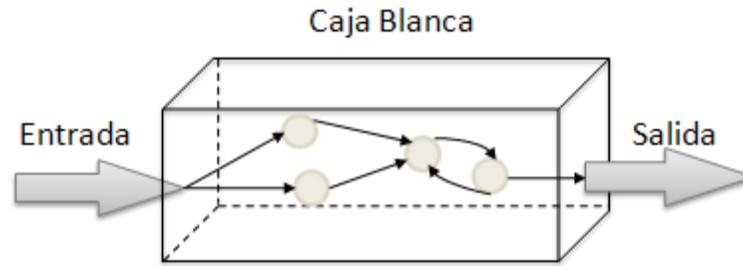
### 3.3 Pruebas de software

Al desarrollar un software es muy importante tener una vía para validar que lo que se hizo realmente es correcto o cumple con las expectativas del cliente o usuario final. Para ello es necesario conocer las funcionalidades específicas y determinar si realmente cumplen el objetivo para lo cual fueron diseñadas, demostrando que las funciones son operativas. De igual manera se pueden realizar pruebas internas en el código que se genera producto de la implementación, y así asegurar que todos los elementos existentes cumplan una función y den lugar al producto final. Todo esto se resume en las pruebas de caja negra y las pruebas de caja blanca.

#### 3.3.1 Pruebas de Caja blanca o Estructurales

Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que:

1. Garanticen que se ejercita al 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 (16).



**Figura 25:** Representación de técnica de pruebas de caja blanca.

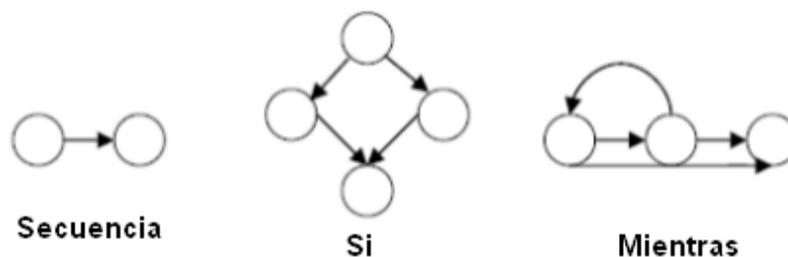
A continuación se citan algunas de las técnicas de prueba de Caja blanca:

- 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 que se muestran en la Figura 26.



**Figura 26:** 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:

## Capítulo 3: Implementación y prueba

- **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 de la funcionalidad encargada de realizar el Pago por resultados en la clase Administrar Nómina, en esta funcionalidad se realiza el pago del trabajador obtenido una vez realizada todas las funcionalidades implementada, se muestra a continuación en la Figura 27.

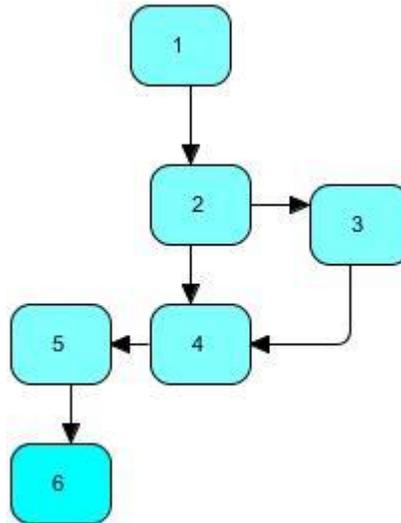
```
public function calcularPagoResultado($trabajador, $idnomina,$idprenomina,$idperiodopago){

    $params->idnomina = $idnomina; 1
    $params->idtrabajador = $trabajador;1
    $procnom = new DatNprocnomina(); 1
    $procesamiento = $procnom ->buscarProcNomina($params); 1
    $params->idprocnomina=$procesamiento->idprocnomina; 1
    $salarioBruto=$procesamiento->salarioescala-$procesamiento->contribucion; 1
    $penalizacion= $this->pIntegrator->pago_resultado->obtenerPenalizacion(); 1
    $cumplimiento= $this->pIntegrator->pago_resultado->obtenerEvaluacion(); 1
    $sumaCumplimiento=$this->sumaCumplimiento($cumplimiento,$trabajador); 1
    $sumaPenalizacion=$this->sumaPenalizacion($penalizacion,$trabajador); 1
    $valorrealpenalizacion=$salarioBruto*($sumaPenalizacion/100); 1
    $valorrealcumplimiento=(($salarioBruto-$valorrealpenalizacion)*($sumaCumplimiento/100)); 1
    $sn=$salarioBruto-$valorrealpenalizacion+$valorrealcumplimiento; 11
    $salcobrar=$sn-$procesamiento->retencion; 1
    if($salcobrar<0){ 2
        $salcobrar=0;3
    } 4
    $params->pagoresultado=$salcobrar; 5
    $params->salarioNeto=$sn; 5
    $params->salarioBruto=$salarioBruto; 5
    $params->salarioscobrar=$salcobrar; 5
    $result=$this->ActualizarCampos($params); 5
    return $result; 5
} 6
```

Figura 27: Funcionalidad Calcular Pago por resultados.

## Capítulo 3: Implementación y prueba

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 28.



**Figura 28:** 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.

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

$$V(G) = 2$$

**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.

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

$$V(G) = 2$$

**Fórmula 3.**  $V(G) = R$

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

## Capítulo 3: Implementación y prueba

Resultado.

$$V(G) = 2$$

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, 3, 4, 5, 4, 6

**Camino básico # 2:** 1, 4, 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 7:** Caso de prueba para el camino básico 1.

<b>Camino básico # 1:</b> 1, 2, 3, 4, 5, 4, 6	
Descripción	Se calcula el Pago por resultados de los trabajadores.
Condición de ejecución	Deben estar realizadas todas las funcionalidades.
Resultados esperados	Obtener el pago de los trabajadores al procesar la nómina Pago por resultados en la clase Administrar nómina.
Salida	Salario a cobrar del trabajador.

**Tabla 8:** Caso de prueba para el camino básico 2.

<b>Camino básico # 2:</b> 1, 2,4,5,6	
Descripción	Se calcula el Pago por resultados de los trabajadores.

Condición de ejecución	Deben estar realizadas todas las funcionalidades.
Resultados esperados	Obtener el pago de los trabajadores al procesar la nómina Pago por resultados en la clase Administrar nómina.
Salida	Devuelve 0 porque en algún momento el pago fue menos que el salario escala que le corresponde al trabajador.

### Resultado de las pruebas de Caja blanca

Para la realización de las pruebas de caja blanca al sistema, fueron analizadas todas las funcionalidades, con las que se comprobaron que el flujo de trabajo de las funciones es el correcto y que cumple con las condiciones necesarias que se habían planteado.

### 3.2.2 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 (16):

- 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 29:** Representación de técnica de prueba de Caja negra.

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

- Métodos de prueba basados en grafos.
- Análisis de valores límites.

## Capítulo 3: Implementación y prueba

- 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.
- 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.

### Resultado de las pruebas de Caja negra

Al componente Pago por resultado se le realizaron pruebas de caja negra por parte de los desarrolladores del proyecto Capital humano. Se realizaron 4 iteraciones encontrando no conformidades en las tres primeras y dejando el componente libre en la última iteración.

A continuación se ofrece detalles de los resultados obtenidos durante dichas pruebas, representándose la cantidad de no conformidades detectadas para cada iteración. Se clasifican en significativas y no significativas según su complejidad y grado de impacto en la solución como muestra la Tabla 9.

**Tabla 9:** Iteraciones realizada por el proyecto Capital humano.

Iteraciones	Cantidad de NC	Tipo		Significativas	No significativas
		Documentación	Aplicación		
1ra	45	25	20	15	30
2da	20	13	7	8	12
3ra	9	5	4	3	6

## *Capítulo 3: Implementación y prueba*

<b>4ta</b>	0	0	0	0	0
------------	---	---	---	---	---

También se realizaron las pruebas de caja negra a la solución desarrollada por el Grupo de Aseguramiento de la Calidad del Centro CEIGE, por problemas de personal solo se le realizó una iteración al componente donde se encontraron no conformidades como se muestra en la Tabla 10. Partiendo de que no se realizaron todas las iteraciones a tiempo se recomienda que el componente realice todas las iteraciones necesarias para obtener el alta de calidad.

**Tabla 10:** Iteraciones realizadas por el grupo de calidad del centro.

Iteraciones	Cantidad de NC	Tipo		Significativas	No significativas
		Documentación	Aplicación		
<b>1ra</b>	30	18	12	14	16

### **Conclusiones parciales**

- Se implementaron todas las funcionalidades del componente Pago por resultados cumpliendo con las normas cubanas.
- Se describieron y aplicaron las pruebas de caja blanca y caja negra, para evaluar el sistema desarrollado, las cuales mostraron que el sistema contaba con un adecuado funcionamiento, demostrando la calidad requerida en el mismo.

## **CONCLUSIONES**

- Se desarrolló el componente Pago por resultados del subsistema Capital humano a partir del diseño propuesto, el cual permite la gestión de este proceso como establecen las normas vigentes en el país para las entidades empresariales presupuestadas cubanas.
- Se realizaron las pruebas de caja blanca y caja negra con el objetivo de minimizar la ocurrencia de errores del sistema antes de su implantación en las entidades nacionales, las cuales mostraron que el sistema contaba con un adecuado funcionamiento, demostrando así la calidad requerida en el mismo.

### **RECOMENDACIONES**

El objetivo general de este trabajo fue alcanzado, pero durante su desarrollo, han surgido ideas que sería recomendable tener en cuenta para su futuro perfeccionamiento:

- Realizarle las pruebas de caja negra por el grupo de calidad del centro.
- Liberar el componente Pago por resultados por Calisoft<sup>14</sup>.
- Realizar el despliegue del componente propuesto como parte del subsistema de Capital humano del sistema CedruX.

---

<sup>14</sup> Calisoft Centro de calidad para soluciones informáticas.

## **REFERENCIAS BIBLIOGRÁFICAS**

1. [En línea] [Citado el: 10 de 02 de 2013.] [http://www.ecured.cu/index.php/Sistema\\_de\\_Pago\\_por\\_Resultados](http://www.ecured.cu/index.php/Sistema_de_Pago_por_Resultados).
2. Cuesta Santos, Armando. Tecnología de gestión de recursos humanos. La Habana : s.n., 2005.
3. 3000), Oficina Nacional de Normalización (NC. SISTEMA DE GESTIÓN INTEGRADA DE CAPITAL HUMANO—. La Habana : s.n., 2007.
4. GEIGE. Proceso Capital Humano. 2010.
5. Martínez Sarrión , Ana Margarita. Análisis y diseño del proceso Pago por resultado para el subsistema Capital Humano del Sistema Integral de Gestión Cedrux. La Habana : s.n., 2012. s.n..
6. [En línea] [Citado el: 13 de 05 de 2013.] <http://www.assets.co.cu/asset.asp>.
7. Saga. Sage ERP Mas. [En línea] [Citado el: 4 de 12 de 2012.] <http://www.sagemas.com/Products/Sage-ERP-MAS-500/Human-Resources-Management/Abra-Payroll..>
8. [En línea] [Citado el: 21 de 05 de 2013.] <http://na.sage.com/sage-500-erp>.
9. SAP. [En línea] [Citado el: 10 de 1 de 2013.] <http://www.sap.com..>
10. www.ecured.cu. [En línea] 07 de 12 de 2007. [Citado el: 15 de 1 de 2013.] [http://www.ecured.cu/index.php/Versat\\_Sarasola..](http://www.ecured.cu/index.php/Versat_Sarasola..)
11. CIMETEL, RodasXXI. CIMETEL. [En línea] 2002. [Citado el: 15 de 01 de 2013.] <http://www.rodasxxi.cu/..>
12. CEIGE. CEIGE-Modelo de Desarrollo de Software. La Habana : s.n., 2012. s.n..
13. Álvarez, Larisa González. Descripción de requerimientos suplementarios. Proyecto Capital Humano v1.0.
14. www.ecured.cu. [En línea] [Citado el: 10 de 04 de 2013.] [http://www.ecured.cu/index.php/Patrones\\_Gof](http://www.ecured.cu/index.php/Patrones_Gof).
15. JUAN, FRANCISCO JAVIER MARTÍNEZ. GUÍA DE CONSTRUCCIÓN DE SOFTWARE EN JAVA CON PATRONES DEL DISEÑO.
16. Pressman, Roger. Ingeniería de software.Un enfoque práctico. s.l. : MC Graw Gil, 1998. s.1..
17. Martínez, Larissa G. GestioPolis. Administración de recursos humanos. [En línea] enero de 2004. [Citado el: 30 de 11 de 2012.] <http://www.gestipolis.com/recursos/documentos/fulldocs/rrhh1/admonrrhhlari.htm..>

## *Referencia Bibliográfica*

18. Isasi, Alina de la Concepción. Estudio sobre la información que se gestiona de los recursos humanos, en los ERP. DESOFT. Habana : s.n., 2010.
19. Assets. Assets. [En línea] [Citado el: 4 de 12 de 2012.] [http://www.assets.co.cu/..](http://www.assets.co.cu/)
20. Saga Software, Inc. SagaMas. [En línea] 12 de 2009. [Citado el: 7 de 12 de 2012.] [http://www.dsdinc.com/dsd/pdf/MAS\\_500\\_StarShip\\_Parcel\\_spec.pdf.56..](http://www.dsdinc.com/dsd/pdf/MAS_500_StarShip_Parcel_spec.pdf.56..)
21. Partner, Sage North America Top ERP. Blytheco. [En línea] [Citado el: 10 de 12 de 2012.] <http://www.blyheco.com/mas500/price-list.asp>.
22. SAP Professionals 2007. Módulo de Activos Fijos. [En línea] 2007. [Citado el: 12 de 1 de 2013.] [http://www.sap.com/argentina/ecosystem/sap\\_professionals/modules/index.epx..](http://www.sap.com/argentina/ecosystem/sap_professionals/modules/index.epx..)
23. EICMA, Dirección . EICMA.Versat Sarasola,Un efectivo sistema cubano de contabilidad. [En línea] 23 de 09 de 2010. [Citado el: 15 de 1 de 2013.] <http://www.eimagr.cu/index.php/component/content/article/60-portada/101-versat-sarasola-un-sistema-para-trabajar>.
24. Gómez Baryolo, Oinier. SOLUCIÓN INFORMÁTICA DE AUTORIZACIÓN EN ENTORNOS MULTIENTIDAD Y MULTISISTEMA. La Habana : s.n., 2010. s.n..
25. Carvalho de Paula Souza, Ana Carolina. [En línea] 4 de 04 de 2011. [Citado el: 16 de 1 de 2013.] <http://www.espweb.uem.br/wp/wp-content/uploads/2011/09/anacarolina.pdf..>
26. Oiner Gómez Baryolo, Yoandry Morejón Borbón y Darien García. ARQUITECTURA TECNOLÓGICA PARA EL DESARROLLO DE SOFTWARE. [En línea] 22 de 05 de 2011. [Citado el: 17 de 1 de 2013.] <http://www.buenastareas.com/ensayos/Arquitectura-Tecnol%C3%B3gica-Pa>.
27. Pérez, Marío Raúl. Google Sites. [En línea] 6 de 06 de 2011. [Citado el: 17 de 01 de 2013.] <https://sites.google.com/site/zendframeworkextjsdoctrine/guias/doctrine>.
28. Sons, John Wiley &. Javascript® bible,sixth edition. New York : Inc.New York, 2007.
29. Flanagan, David. Java Script. The definitive guide. Estados Unidos de America : O'Reilly, 2006.
30. Nidia. Buenas tareas. Lenguaje De Programación. [En línea] 09 de 09 de 2011. [Citado el: 20 de 01 de 2013.] <http://www.buenastareas.com/ensayos/Lenguaje-De-Programacion/2770237.html..>
31. Lerdorf, Kevin Tatroe & Peter MacIntyre Rasmus. Programming PHP. Estados Unidos de América : O'Reilly, 2006.
32. free download manager. [En línea] [Citado el: 20 de 01 de 2013.] [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML).

## *Referencia Bibliográfica*

33. Netbeans. [En línea] [Citado el: 18 de enero de 2013.] [http://netbeans.org/community/releases/69/index\\_es.html](http://netbeans.org/community/releases/69/index_es.html).
34. The Apache Software Foundation. Apache Software Foundation. IndexProject Listing. [En línea] [Citado el: 23 de 01 de 2013.] <http://projects.apache.org/indexes/quick.html>.
35. López, Alejandro Cadaval. Mozilla Firefox, el navegador web del momento. [En línea] [Citado el: 23 de 01 de 2013.] <http://www.maestrosdelweb.com/editorial/firefox/>.
36. Martínez, Alejandro y Martínez, Raúl. Guía a Rational Unified Process. Universidad de Castilla-La Mancha. [En línea] [Citado el: 1 de 02 de 2013.] <http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-Guia%20RUP.pdf>.
37. Chiavenato, Idalberto. Administración de recursos Humanos. México : Editorial Mc Graw – Hill, 1988.
38. Santos, Armando Cuesta. Tecnología de la Gestión de Recursos Humanos. Habana : Editorial Academia, 1999.
39. Dr.Noel Varela Izquierdo, Msc Guillermo A. López, Msc Eduardo Concepción. Procedimiento para la Evaluación del Desempeño en el Sector Hotelero Cubano. Cienfuegos : s.n.
40. Cubana, Norma. SISTEMA DE GESTIÓN INTEGRADA DE CAPITAL HUMANO—. 2007.
41. Morales, Edualdo Gutierrez. GRH, evolución, conceptos y diferentes perspectivas vistas en la realidad cubana. Sancti Spiritus : s.n., 2009.
42. Sierra, Pavel Pérez. “Evaldes” Aplicación informática de apoyo al Sistema de Pago Adicional Centrado en la Evaluación por Desempeño. La habana : s.n., 2010.
43. CEIGE. CEIGE-Modelo de Desarrollo de Software. La Habana : s.n., 2012.
44. Baryolo, Oiner Gómez. SOLUCIÓN INFORMÁTICA DE AUTORIZACIÓN EN ENTORNOS MULTIENTIDAD Y MULTISISTEMA. La Habana : s.n., 2010.
45. —. Plantilla Registro de la Propiedad intelectual(Sauxe). La Habana : s.n., 2010.
46. Flanagan, David. Java Script. The definitive guide. Estados Unidos de America : O’Reilly, 2006. 0-596-10199-2.
47. Sons, John Wiley &. Javascript® bible, sixth edition. New York : Inc. New York, 2007. 9780470069165.
48. Lerdorf, Kevin Tatroe & Peter MacIntyre Rasmus. Programming PHP. Estados Unidos de América : O’Reilly, 2006. 978-0-596-00681-5.
49. Cortés, María Escat. GestioPolis. GestioPolis. [En línea] GestioPolis., 2008. <http://www.gestiopolis.com/canales/derrhh/articulos/37/grrhhest.htm>.

## *Referencia Bibliográfica*

50. S., Larissa G. Martínez. GestioPolis. Administración de recursos humanos. [En línea] enero de 2004. [Citado el: 10 de 12 de 2012.] <http://www.gestiopolis.com/recursos/documentos/fulldocs/rrhh1/admonrrhhlari.htm>.
51. Wilsoft. [En línea] <http://www.wilsoft-la.com/qtraining.htm>.
52. Universia. Universia. [En línea] 16 de noviembre de 2008. [http://www.universia.cl/portada/actualidad/noticia\\_actualidad.jsp?noticia=139652](http://www.universia.cl/portada/actualidad/noticia_actualidad.jsp?noticia=139652).
53. Oiner Gómez Baryolo, Yoandry Morejón Borbón y Darien García. ARQUITECTURA TECNOLÓGICA PARA EL DESARROLLO DE SOFTWARE. [En línea] 22 de mayo de 2011. [Citado el: 10 de diciembre de 2012.] <http://www.buenastareas.com/ensayos/Arquitectura-Tecnol%C3%B3gica-Para-El-Desarrollo-De/2210642.html>.
54. Pérez, Mario Raúl. Google Sites. [En línea] 6 de julio de 2011. [Citado el: 10 de diciembre de 2012.] <https://sites.google.com/site/zendframeworkextjsdoctrine/guias/doctrine>.
55. SOUZA, ANA CAROLINA CARVALHO DE PAULA. EspWeb. [En línea] 4 de abril de 2011. [Citado el: 10 de diciembre de 2012.] <http://www.espweb.uem.br/wp/wp-content/uploads/2011/09/anacarolina.pdf>.
56. Nidia. Buenas tareas. Lenguaje De Programación. [En línea] 9 de septiembre de 2011. [Citado el: 10 de diciembre de 2012.] <http://www.buenastareas.com/ensayos/Lenguaje-De-Programacion/2770237.html>.
57. UBUNTU, GUÍA DOCUMENTADA PARA. Subversion. [En línea] 9 de enero de 2009. [Citado el: 9 de diciembre de 2012.] <http://www.guia-ubuntu.org/index.php?title=Subversion>.
58. The Apache Software Foundation. Apache Software Foundation Index: Project Listing. [En línea] [Citado el: 8 de diciembre de 2012.] <http://projects.apache.org/indexes/quick.html>.
59. López, Alejandro Cadavid. Mozilla Firefox, el navegador web del momento. [En línea] [Citado el: 10 de diciembre de 2012.] <http://www.maestrosdelweb.com/editorial/firefox/>.
60. Salinas, Oscar Javier. Gestipolis. Gestipolis. [En línea] noviembre de 2000. <http://www.gestiopolis.com/canales/derrhh/articulos/No%202/evaluaciondeldesempeo.htm>.
61. Netbeans. [En línea] [Citado el: 18 de enero de 2013.] [http://netbeans.org/community/releases/69/index\\_es.html](http://netbeans.org/community/releases/69/index_es.html).
62. Arias, Fernando. Administración de recursos humanos. 1989.
63. otros, Consuelo Mora y. Administración de Recursos Humanos. Mexico : s.n., 1996.
64. Pressman, Roger. Ingeniería de software. Un enfoque práctico. s.l. : Interamericana de España, 2002.
65. Pressman, Rger. Ingeniería de software. 2005.
66. Ruiz, Francisco Alarcos. [En línea]

## *Referencia Bibliográfica*

67. Álvarez, Larisa González. Descripción de requerimientos suplementarios. Proyecto Capital Humano v1.0.
68. Martínez, Raúl Martínez y Alejandro. Guía a Rational Unified Process. Universidad de Castilla-La Mancha. [En línea] [Citado el: 18 de febrero de 2013.] <http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-Guia%20RUP.pdf>.
69. Pressman, Roger. Ingeniería de software. Un enfoque práctico. s.l. : MC Graw Gil, 1998.
70. 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>.
71. Álvarez, Larisa González. Descripción de requerimientos suplementarios. Proyecto Capital Humano . v1.0..
72. [En línea] [Citado el: 20 de 4 de 2013.] [http://www.ecured.cu/index.php/Pruebas\\_de\\_caja\\_blanca](http://www.ecured.cu/index.php/Pruebas_de_caja_blanca).
73. Álvarez, Larisa González. Descripción de requerimientos suplementarios. Proyecto Capital Humano v1.0. [En línea]
74. [En línea] [Citado el: 10 de 02 de 2013.] <http://www.monografias.com/trabajos89/generalidades-pago-resultados-area-servicios-internos2>.
75. [En línea] [Citado el: 10 de 02 de 2013.] <http://www.monografias.com/trabajos82/sistema-accion-perfeccionar-sistema-estimacion-moral/sistema-accion-perfeccionar-sistema-estimacion-moral2.shtml>.

## ANEXOS

### Anexo 1: Requisitos funcionales

<b>RF-1</b>	<p><b>Gestionar sistemas de pago:</b></p> <p>En la agrupación de requisito se gestionan los sistemas de pago de una empresa.</p> <p>1.1 Adicionar sistema de pago.            1.2 Modificar sistema de pago.            1.3 Eliminar sistemas de pago.            1.4 Listar indicadores.            1.5 Asociar indicadores al sistema de pago.            1.6 Desasociar indicadores al sistema de pago.            1.7 Listar sistemas de pago.</p>
<b>RF-2</b>	<p><b>Gestionar grupo de pago</b></p> <p>En la agrupación de requisito se gestionan los grupos de pago que pueden existir en una empresa.</p> <p>2.1 Adicionar grupo de pago.            2.2 Modificar grupo de pago.            2.3 Eliminar grupo de pago.            2.4 Asociar trabajadores al grupo de pago.            2.5 Desasociar trabajadores al grupo de pago.            2.6 Listar trabajadores.            2.7 Listar grupos de pago.</p>
<b>RF-3</b>	<p><b>Gestionar tipos de sistemas de pago</b></p> <p>En la agrupación de requisito se gestionan los tipos de sistemas de pago que pueden existir en una empresa.</p> <p>3.1 Adicionar tipo de sistema de pago.            3.2 Modificar tipo de sistemas de pago.            3.3 Eliminar tipo de sistemas de pago.            3.4 Listar tipo de sistema de pago.</p>
<b>RF-4</b>	<p><b>Gestionar Indicadores</b></p> <p>En la agrupación de requisito se gestionan los indicadores que se pueden asociar a los sistemas de pago.</p>

	<p>4.1 Adicionar indicadores.</p> <p>4.2 Modificar indicadores.</p> <p>4.3 Eliminar indicadores.</p> <p>4.4 Listar indicadores.</p> <p>4.5 Asociar indicador al área.</p> <p>4.6 Desasociar indicador al área.</p> <p>4.7 Asociar indicador al cargo.</p> <p>4.8 Desasociar indicador al cargo.</p> <p>4.9 Listar cargos asociados a un indicador.</p> <p>4.10 Listar áreas asociadas a un indicador.</p>
<b>RF-5</b>	<p><b>Gestionar penalización</b></p> <p>En la agrupación de requisito se gestionan las penalizaciones de los indicadores asociados a los trabajadores y los indicadores asociados a la empresa en general.</p> <p>5.1 Adicionar penalización.</p> <p>5.2 Modificar penalización.</p> <p>5.3 Eliminar penalizaciones.</p> <p>5.4 Asociar trabajadores a las penalizaciones.</p> <p>5.5 Desasociar trabajadores a las penalizaciones.</p> <p>5.6 Listar penalizaciones</p> <p>5.7 Listar trabajadores</p>
<b>RF-6</b>	<p><b>Evaluación de los indicadores</b></p> <p>En la agrupación de requisito se permite evaluar a los indicadores.</p> <p>6.1 Listar trabajadores</p> <p>6.2 Evaluar indicadores.</p> <p>6.3 Modificar evaluación.</p> <p>6.4 Eliminar evaluación</p> <p>6.5 Listar evaluaciones de indicador.</p> <p>6.6 Listar todas las evaluaciones</p>
<b>RF-7</b>	<p><b>Administrar nómina de Pago por resultados</b></p> <p>En la agrupación de requisito se permite administrar la nómina de Pago por resultados que no es más que agrupar las funcionalidades de adicionar, modificar y eliminar así como también adicionar trabajadores y procesar la nómina.</p> <p>7.1 Adicionar nómina Pago por resultados.</p>

	<p>7.2 Adicionar trabajadores a la nómina.</p> <p>7.3 Adicionar fácilmente trabajadores a la nómina.</p> <p>7.4 Modificar nómina Pago por resultados.</p> <p>7.5 Eliminar nómina Pago por resultados.</p> <p>7.6 Procesar nómina Pago por resultados.</p> <p>7.7 Listar nómina Pago por resultados.</p> <p>7.8 Listar trabajadores.</p> <p>7.9 Imprimir nómina Pago por resultados.</p>
<b>RF-8</b>	<p><b>Revisar nómina Pago por resultados</b></p> <p>En la agrupación de requisito se realiza la revisión de la nómina para comprobar que los cálculos estén correctamente procesados</p>
	<p>8.1 Confirmar nómina Pago por resultados.</p> <p>8.2 Rechazar nómina Pago por resultados.</p> <p>8.3 Contabilizar nómina Pago por resultados.</p> <p>8.4 Listar nómina Pago por resultados.</p> <p>8.5 Buscar nómina Pago por resultados.</p> <p>8.6 Imprimir nómina Pago por resultados.</p>
	<p><b>Funciones de la nómina</b></p> <p>En la agrupación de requisito se realizan todas las funciones o cálculos de la nómina.</p>
<b>RF-9</b>	<p>9.1 Calcular Pago por resultados</p> <p>9.2 Calcular salario por escala</p> <p>9.3 Contribución de seguridad social al trabajador.</p> <p>9.4 Retenciones</p> <p>9.5 Realizar redondeo.</p>

**Anexo 2:** Diagrama de clases.

Diagrama de clases gestionar tipo de pago.

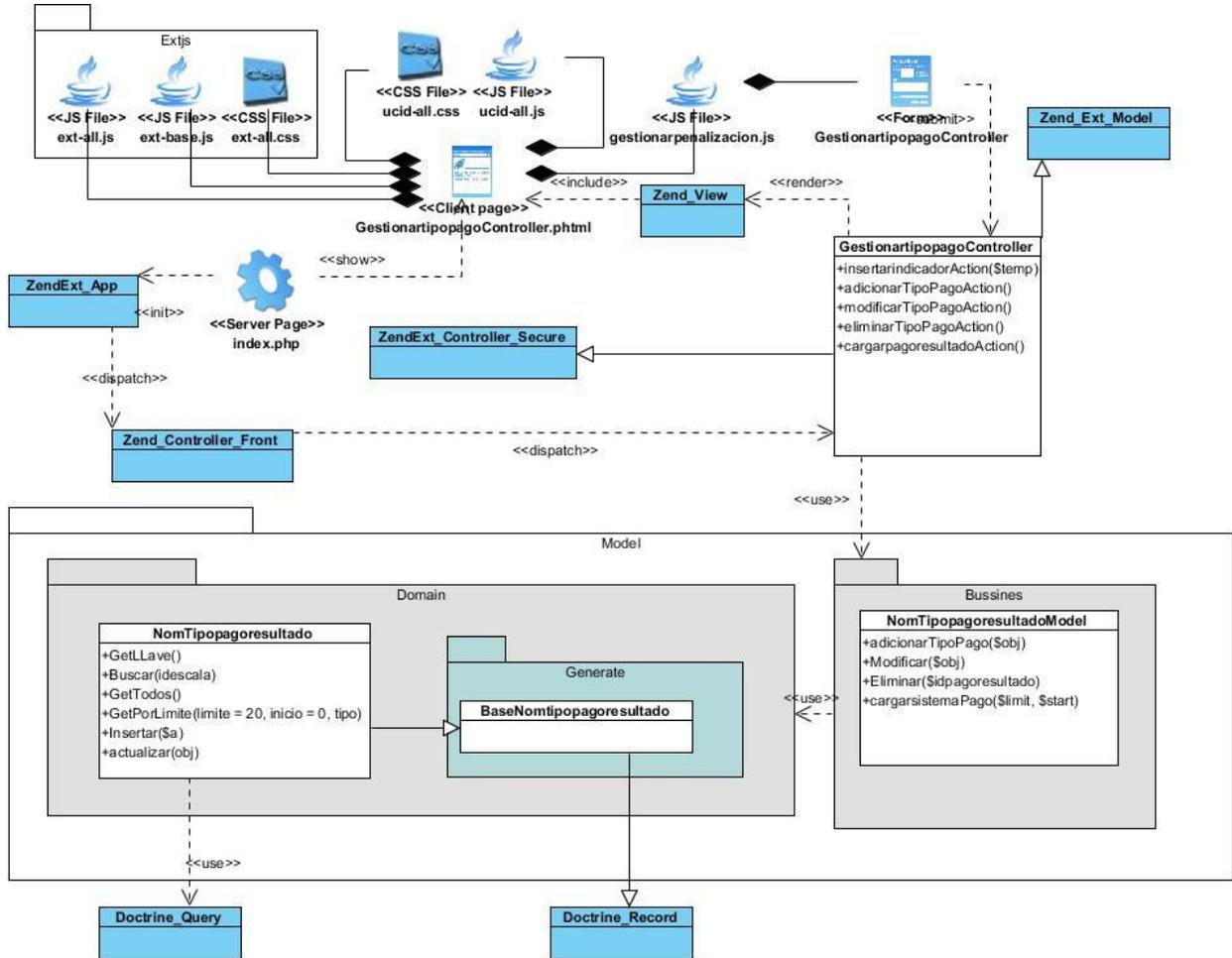




Diagrama de clases Gestionar penalización.

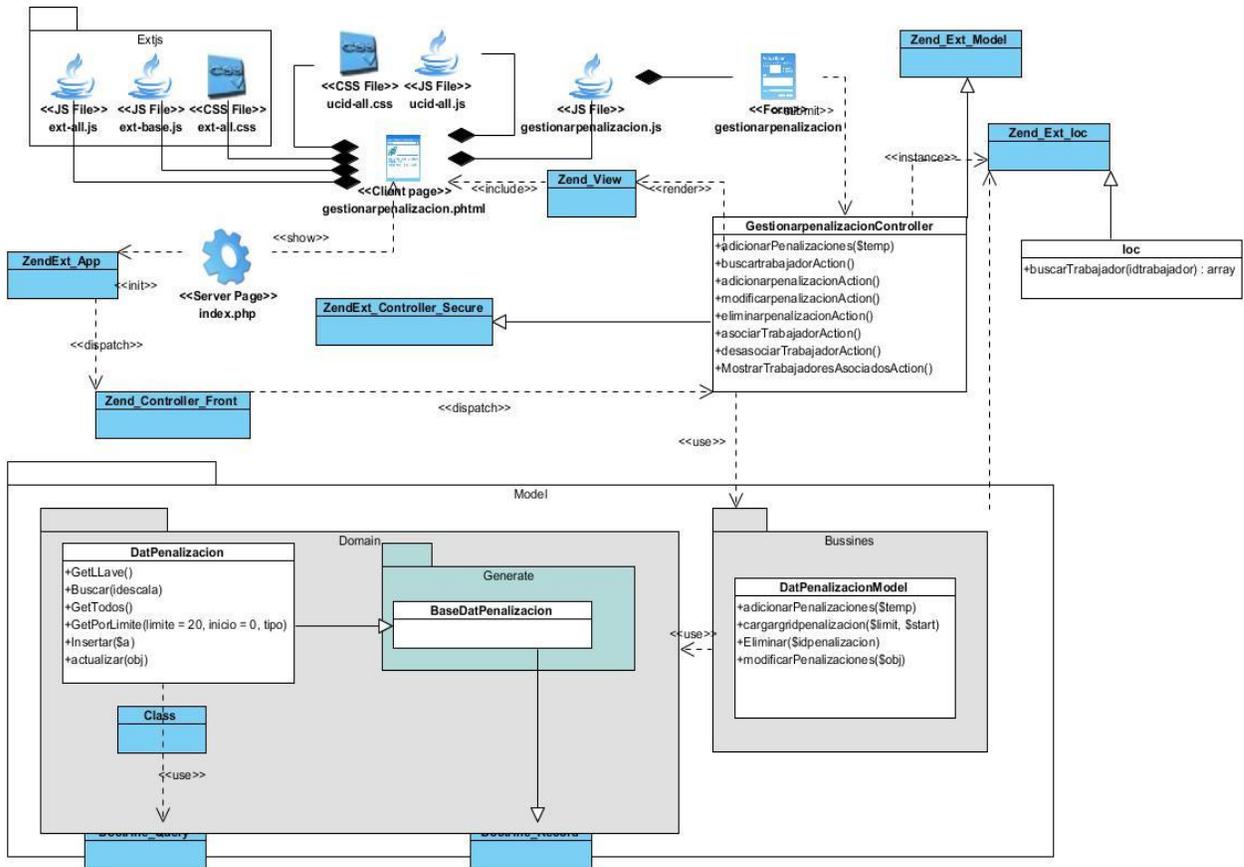
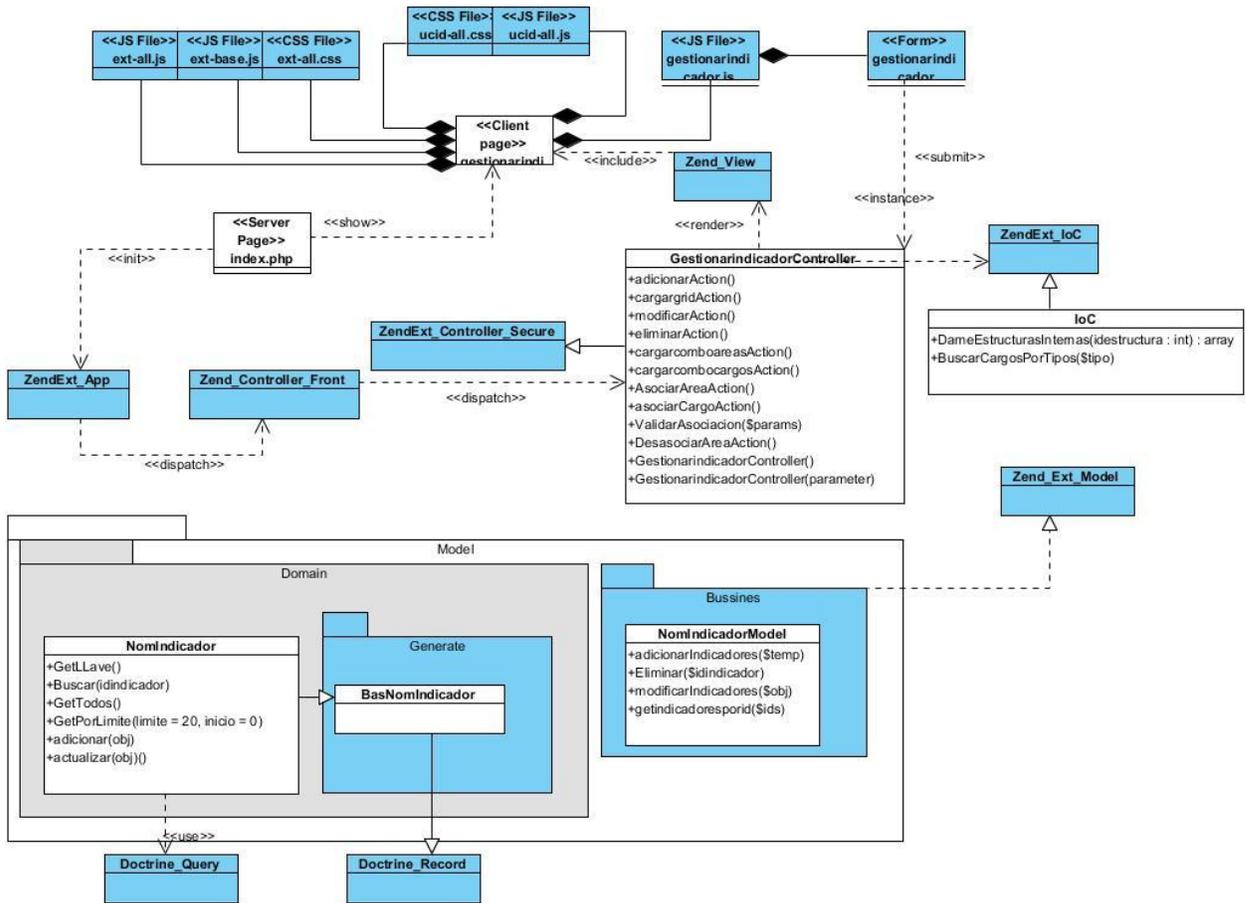


Diagrama de clases Gestionar Indicadores.





## Anexo 3:

Tabla de la clase controladora GestionarpenalizacionController.

Nombre: GestionarpenalizacionController	
Tipo de clases: Controladora	
Atributos	Tipo
adicionarpenalizacionAction()	Este método recibe los datos de la interfaz y hace un llamado al método adicionarPenalizaciones(\$obj) de la clase
modificarpenalizacionAction()	Este método recibe los datos de la interfaz y hace un llamado al método modificarPenalizaciones(\$obj) de la clase
eliminarpenalizacionAction ()	Este método recibe por parámetros el id del pag por resultados que se desea eliminar y hace un llamado al método Eliminar(\$obj) de la clase DatPenalizacionModel del dominio.
buscartrabajadorAction	Hace uso del servicio BuscarTrabajador que devuelve los trabajadores que hay en el sistema.
desasociarTrabajadorAction()	Este método una vez escogido el trabajador se asocia al sistema haciendo un llamado al método desasociarTrabajador(\$Des)de la clase <i>DatTrabajadorpenalizacionModel.php</i> .
asociarTrabajadorAction()	Este método una vez escogido el trabajador se asocia al sistema haciendo un llamado al método asociarTrabajadores(\$params) de la clase <i>DatTrabajadorpenalizacionModel.php</i> pago que se desea. cargar y se hace un llamado al método cargarsistemaPago() de la clase <i>NomTipopagoresultadoModel.php</i> .
MostrarTrabajadoresAsociadosAction()	Este método muestra los trabajadores asociados a una penalización llamando al método GetCantidadAsociados de la clase <i>DatTrabajadorpenalizacionModel.php</i>

adicionarpenalizacionAction()	Este método recibe los datos de la interfaz y hace un llamado al método adicionarPenalizaciones(\$obj) de la clase <i>DatPenalizacionModel.php</i> .
-------------------------------	--

Tabla de la clase Modelo DatPenalizacionModel.

<b>Nombre: DatPenalizacionModel</b>	
<b>Tipo de clases: DatPenalizacionModel</b>	
<b>Atributos</b>	<b>Tipo</b>
adicionarPenalizaciones(\$temp)	Este método recibe los datos de la interfaz y hace un llamado al método adicionarPenalizaciones(\$obj) de la clase DatPenalizacionModel.
modificarPenalizaciones(\$obj)	Este método recibe los datos de la interfaz y hace un llamado al método actualizar(\$obj) de la clase DatPenalizacion.
Eliminar(\$idpenalizacion)	Este método recibe por parámetros el id del pago por resultados que se desea eliminar y hace un llamado al método Buscar(\$idpenalizacion) de la clase DatPenalizacion.
buscartrabajadorAction	Hace uso del servicio BuscarTrabajador que devuelve los trabajadores que hay en el sistema.
cargargridpenalizacion(\$limit, \$start)	Este método hace un llamado al método GetPorLimite(\$limit, \$start) de la clase DatPenalizacion() que se desea.

Tabla de la clase del dominio DatPenalizacion.

<b>Nombre: DatPenalizacion</b>	
<b>Tipo de clase:</b> Modelo (Domain)	
<b>Para cada responsabilidad</b>	
<b>Nombre del método</b>	<b>Descripción</b>
Buscar(idindicador)	Este método busca en la base de datos el pago por resultados que coincida con el identificador que se le pasa por parámetro.
GetTodos()	Este método devuelve todos los pagos por resultados que se encuentran en la base de datos.
GetPorLimite(limite = 20, inicio = 0)	Devuelve todos los pagos por resultados de la base de datos de acuerdo a un rango.
Insertara(\$a)	Este método recibe como parámetro un objeto insertándolo en la base de datos.
actualizar(obj)	Recibe como parámetro un objeto actualizándolo en la base de datos.