

**Universidad de las Ciencias Informáticas**  
**Facultad 3**



**“Desarrollo del componente Disciplina Laboral del  
subsistema Capital Humano del Sistema Integral de  
Gestión Cedrux”**

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

**Autor (es):** Yossel Bernal González

**Tutores:** Ing. William González Obregón

Ing. Rodolfo Rodríguez Molinet

Ciudad de la Habana, junio de 2011

## DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco 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 \_\_\_\_\_.

**Yossel Bernal González**

**Rodolfo Rodríguez Molinet**

---

Firma del Autor

---

Firma del Tutor

## DATOS DE CONTACTO

### **Datos del Tutor(es):**

*Ing. Rodolfo Rodríguez Molinet*

Ingeniero en Ciencias Informáticas. Graduado en la Universidad de las Ciencias Informáticas en el 2008. Ha impartido las asignaturas Matemática I y II y Álgebra Lineal. Actualmente Profesor de Investigación de Operaciones y Desarrollador de la Línea Capital Humano del Departamento Desarrollo de Productos, CEIGE, Facultad 3.

Correo: [rmolinet@uci.cu](mailto:rmolinet@uci.cu)

*Ing. William González Obregón*

Ingeniero en Ciencias Informáticas. Graduado como título de oro en la Universidad de las Ciencias Informáticas en el 2010. Se desempeña como Desarrollador de software al frente de una de las factorías de la Línea Capital Humano del Departamento Desarrollo de Productos, CEIGE, Facultad 3.

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

## RESUMEN

Actualmente, las empresas cubanas se ven en la necesidad de contar con información de alta calidad que les permita agilizar los procesos económicos basándose en las tecnologías y la información para mejorar la toma de decisiones. Dentro de estos procesos se encuentra la gestión de la disciplina laboral en el área de capital humano, que incluye el control de las violaciones cometidas en la entidad, así como la emisión de las medidas disciplinarias impuestas. El presente trabajo tiene como objetivo desarrollar el componente Disciplina Laboral del subsistema Capital Humano del Sistema Integral de Gestión Cedrux para su integración al cálculo de la nómina. Para el desarrollo del componente se utilizaron como lenguajes de programación, PHP por el lado del servidor y Java Script por el lado del cliente, como gestor de base de datos se utilizó PostgreSQL, para la estructuración y construcción de la solución se trabajó con el modelo de desarrollo del marco de trabajo del Sistema Integral de Gestión Cedrux. Se empleó el Lenguaje Unificado de Modelado (UML) para la modelación del análisis y el diseño y la Notación para el Modelado de Procesos de Negocio (BPMN) para modelar el negocio, aplicándose ambos mediante la herramienta Visual Paradigm para UML. El sistema garantizará el buen funcionamiento de la gestión de la disciplina laboral disminuyendo las dificultades para el cálculo de la nómina por parte del subsistema Capital Humano del Sistema Integral de Gestión Cedrux.

## PALABRAS CLAVE

Capital Humano; Cedrux; Disciplina laboral; ERP; Nómina

## TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....	4
1.1 INTRODUCCIÓN .....	4
1.2 CONCEPTOS GENERALES .....	4
1.3 SISTEMAS INFORMÁTICOS SIMILARES .....	7
1.4 MODELO DE DESARROLLO DE SOFTWARE .....	11
1.5 HERRAMIENTAS Y TECNOLOGÍAS .....	14
1.5.1 Lenguajes de modelado.....	14
1.5.2 Herramientas de modelado .....	16
1.5.3 Lenguajes de programación y frameworks .....	17
1.5.4 Gestor de base de datos.....	20
1.5.5 Herramientas de base de datos .....	21
1.5.6 Navegador Web .....	21
1.5.7 Control de versiones .....	21
1.5.8 Otras herramientas teóricas .....	23
1.6 CONCLUSIONES DEL CAPÍTULO.....	26
CAPÍTULO 2: MODELAMIENTO DEL NEGOCIO Y ANÁLISIS.....	27
2.1 INTRODUCCIÓN .....	27
2.2 MODELACIÓN DEL NEGOCIO .....	27
2.3 MODELO CONCEPTUAL.....	28
2.4 REQUISITOS FUNCIONALES.....	32
2.5 CONCLUSIONES DEL CAPÍTULO.....	52
CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA .....	53
3.1 INTRODUCCIÓN .....	53
3.2 DIAGRAMA DE CLASES DEL DISEÑO.....	53
3.3 DIAGRAMA DE COMPONENTES .....	55
3.4 INTEGRACIÓN ENTRE COMPONENTES .....	55
3.5 MODELO DE DATOS .....	56
3.6 CONCLUSIONES DEL CAPÍTULO.....	57

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN .....	58
4.1 INTRODUCCIÓN .....	58
4.2 MÉTRICAS PARA VALIDAR EL DISEÑO .....	58
4.2.1 Métrica Tamaño Operacional de Clases (TOC) .....	59
4.2.2 Métrica Relaciones entre Clases (RC) .....	63
4.3 MODELO DE PRUEBAS .....	67
4.3.1 Prueba de caja negra .....	68
4.3.2 Prueba de caja blanca .....	71
4.4 CONCLUSIONES DEL CAPÍTULO.....	75
CONCLUSIONES GENERALES .....	76
RECOMENDACIONES .....	77
REFERENCIAS BIBLIOGRÁFICAS.....	78

## INTRODUCCIÓN

En un mundo tan cambiante y globalizado, las empresas requieren de herramientas tecnológicas que contribuyan en la estrategia global de negocio para maximizar la rentabilidad y mejores prácticas operativas en las diferentes disciplinas de la organización. El país avanza con amplios pasos dentro del proceso de utilización ordenada y masiva de las tecnologías de la información y las comunicaciones, siendo este un factor decisivo para el desarrollo de las empresas, economía y de la sociedad cubana.

Hoy en día una de las herramientas informáticas más utilizadas son los sistemas ERP (aplicaciones de planificación de recursos empresariales), los cuales juegan un papel primordial en la vida de las empresas, ya que ayudan a mejorar procesos, reducir tiempo (horas/hombre) y ayudan a centrarse en tareas que agreguen valor. Son paquetes de software que integran toda la gestión de la empresa, proveen arquitectura que cumple con los requisitos internos y externos para soportar sistemas de planeación de recursos de la empresa, con todos los procesos integrados a través de una cadena de suministro.

Una de las áreas que se gestiona en los ERP es el Capital Humano en la cual se permite el acceso en tiempo real para todos los usuarios de una información confiable y segura, donde los procesos de recursos humanos participan en todas las funciones operativas, tácticas y estratégicas del negocio de la empresa. Tiene como premisa fundamental integrar al hombre dentro de la organización como un recurso al que hay que optimizar a partir de una visión renovada, dinámica y competitiva.

Entre las acciones que se realizan en la gestión del capital humano están las relacionadas con el control de la disciplina laboral y el cálculo de la nómina. La nómina es el instrumento que permite de una manera ordenada realizar el pago de salarios a los trabajadores, mientras que el control de la disciplina laboral tiene como objetivo buscar siempre el bienestar y evitar el incumplimiento de las normas en la empresa, por lo cual es necesario registrar las violaciones y las medidas disciplinarias a los trabajadores; algunas de estas medidas traen consigo importes económicos que afectan en el cálculo de la nómina.

En el caso de las entidades cubanas se realiza el control de las medidas disciplinarias aplicadas a los trabajadores de forma manual, esto provoca pérdida de información, duplicación de la misma, atraso en la entrega de la información correspondiente a las medidas impuestas, de esta forma el cálculo de la nómina se torna engorroso y complejo.

En la Universidad de las Ciencias Informáticas se lleva a cabo la realización del Sistema Integral de Gestión CedruX para la administración de los recursos de un grupo de entidades cubanas; el cual cuenta con un subsistema que se encarga de la gestión del capital humano.

Por lo anterior se plantea como **problema a resolver**:

La gestión manual de la Disciplina Laboral para la Administración de Capital Humano dificulta generar la nómina mediante el subsistema de Capital Humano del Sistema Integral de Gestión CedruX.

Se define por lo tanto como **objeto de estudio**: los Sistemas de Gestión del Capital Humano y como **campo de acción**: los Sistemas de Gestión de la Disciplina Laboral.

Para dar respuesta al problema antes planteado se tiene como **objetivo general**: Desarrollar el componente Disciplina Laboral que permita disminuir las dificultades para generar la nómina para su integración al subsistema de Capital Humano del Sistema Integral de Gestión CedruX.

Se definen como **objetivos específicos** los siguientes:

- Fundamentar la investigación mediante la elaboración del Marco Teórico para conocer el estado del arte.
- Proponer una solución informática para la gestión de la disciplina laboral que permita disminuir las dificultades para generar la nómina.
- Validar la solución.

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

- Estudio de los logros y limitaciones en los enfoques existentes sobre la gestión de la Disciplina Laboral para la Administración del Capital Humano.
- Evaluación del contenido de la información obtenida sobre la actividad que se investiga, establecer un diagnóstico de las tendencias actuales y tomar posición al respecto.
- Análisis de los procesos de control de la disciplina laboral.
- Realización del diseño de la propuesta de solución.



- Implementación de los requerimientos especificados para los procesos del control de la disciplina laboral.
- Realización de pruebas a la solución.

Se tiene como **Idea a defender**: Si se elabora el componente Disciplina Laboral para la administración del Capital Humano entonces deben disminuir las dificultades para generar la nómina mediante el subsistema de Capital Humano del Sistema Integral de Gestión CedruX.

En cumplimiento de las distintas tareas antes mencionadas, se ponen en práctica los siguientes **métodos de investigación**:

#### **Métodos teóricos:**

- **Analítico – sintético**: Posibilitando procesar toda la información enfocada hacia la investigación, permitiendo organizar y simplificar el análisis de la documentación; extrayendo los elementos más importantes relacionados con la Gestión del Capital Humano.
- **Histórico – lógico**: Para conocer los antecedentes y tendencias actuales en la gestión del Capital Humano en los ERP.

#### **Métodos empíricos:**

- **Experimento**: Favoreciendo el desarrollo de pruebas para la verificación de las funcionalidades implementadas, con el fin de detectar errores y comprobar su correcto funcionamiento.
- **Observación**: Para percibir y planificar como quedaría concebido el sistema.

La estructura en capítulos será la siguiente:

**Capítulo 1:** Describe un estado del arte de los sistemas que gestionan el capital humano, así como la fundamentación de las metodologías, lenguajes y herramientas a utilizar.

**Capítulo 2:** Detalla el análisis del sistema, teniendo en cuenta la modelación de los procesos de negocio y la definición de requisitos de software.

**Capítulo 3:** Detalla el diseño y la documentación de la implementación de la solución.

**Capítulo 4:** Se lleva a cabo la etapa de verificación y validación de la solución.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

En este capítulo se abarcan un grupo de conceptos y teorías relacionadas con el negocio. Se realiza una investigación en base a sistemas que gestionen el capital humano, además se analiza el modelo de desarrollo, las tecnologías y las herramientas a utilizar en el desarrollo del sistema.

### 1.2 Conceptos Generales

#### 1.2.1 Administración de Recursos Humanos

La Administración de Recursos Humanos consiste en la planeación, organización, desarrollo y coordinación, así como también control de técnicas, capaces de promover el desempeño eficiente del personal. Significa conquistar y mantener las personas en la organización, trabajando y dando el máximo de sí, con una actitud positiva y favorable. Representa todas aquellas cosas que hacen que el personal permanezca en la organización. (1)

Son todas aquellas actividades diseñadas para ocuparse y coordinar a las personas necesarias para una organización. Busca construir y mantener un entorno de excelencia en la calidad para habilitar mejor a la fuerza de trabajo en la consecución de los objetivos de calidad y de desempeño operativo de la empresa.

#### 1.2.2 Nómina

Se le llama nómina a una lista conformada por el conjunto de trabajadores de una entidad determinada a los cuales se les va a remunerar por los servicios prestados. Permite de una manera ordenada, realizar el pago de salarios a los trabajadores, así como proporcionar información contable y estadística, tanto para la empresa como para el organismo encargado de regular las relaciones laborales. Aunque en cada entidad puede variar la forma de calcular y contabilizar la nómina, existen ciertos pasos comunes a todas, como la preparación de la nómina con los nombres y las remuneraciones de los trabajadores.

#### 1.2.3 Gestión de la Disciplina Laboral

La **disciplina** en una organización se va a entender como la capacitación que corrige y moldea las actividades y la conducta de todos los empleados, para que los esfuerzos individuales de estos se encaminen mejor hacia la cooperación y el desempeño. (2)

El objetivo de la disciplina empresarial es desarrollar y mantener reglamentos de trabajo efectivos así como crear y promover relaciones de trabajo armónicas con el personal.

Las **violaciones disciplinarias** ocurren cuando las normas no se cumplen o son violadas, es decir, cuando las reglas internas que se hayan establecido con antelación no tengan un correcto cumplimiento por parte de todos los integrantes de la empresa.

En una organización, las reglas y los reglamentos tienen el mismo objeto que las leyes en la sociedad y cuando alguien infringe una de estas reglas o reglamentos será necesario disciplinarle. En este caso, la justicia tiene tres fundamentos: las reglas y los reglamentos, un sistema de sanciones progresivas y un proceso de apelación.

Se consideran violaciones de la disciplina laboral, las siguientes:

- La infracción del horario de trabajo.
- La ausencia injustificada.
- La falta de respeto a superiores, compañeros de trabajo o a terceras personas en la entidad o en ocasión del desempeño del trabajo.
- La desobediencia.
- El maltrato de obra o palabra a superiores, compañeros de trabajo o a terceras personas en la entidad o en ocasión del desempeño del trabajo.
- La negligencia.
- Las violaciones de las disposiciones vigentes en la entidad laboral sobre el secreto estatal, técnico o comercial y para la seguridad y protección física.
- El daño a los bienes de la entidad laboral o de terceras personas, en la entidad o en ocasión del trabajo.

- La pérdida, sustracción o desvío y la apropiación, mediante engaño, de bienes o valores propiedad del centro de trabajo o de terceros.
- Cometer hechos o incurrir en conductas que puedan ser constitutivas de delitos en la entidad laboral o en ocasión del desempeño del trabajo.
- La inobservancia de los reglamentos disciplinarios vigentes en las entidades laborales .
- El incumplimiento injustificado, por parte del trabajador, de los deberes que la legislación sobre protección e higiene del trabajo establece. (3)

Las **medidas disciplinarias** tienen por objeto en la mayoría de los casos la implementación de acciones correctivas a fin de que el trabajador sea notificado de su falta laboral y modifique un comportamiento desviado de las obligaciones que le impone su relación laboral con la empresa para la cual presta sus servicios.

Los tipos de medidas disciplinarias pueden ser:

- Preventivas: Se alienta a los trabajadores a que cumplan las normas y procedimientos para prevenir desviaciones.
- Correctivas: Son las medidas que se ejercen cuando se producen las desviaciones.
- Progresivas. Las sanciones se van haciendo más severas en la medida que se repiten las faltas.

El fin inmediato de las medidas disciplinarias es notificar la falta, lograr que se corrija la misma y que el trabajador no incurra en la desviación señalada nuevamente, a fin de recuperar la normalidad en sus actividades. Es importante señalar que la falta puede ser de tal gravedad que el objetivo de la medida sea la terminación del servicio, por lo tanto, de acuerdo a la falta existen diferentes medidas disciplinarias como son:

- Amonestación pública ante el colectivo del infractor.
- Multa de hasta el importe del 25 % del salario de un mes, mediante descuentos de hasta un 10 % del salario mensual.
- Inhabilitación para ser ascendido o promovido antes del transcurso de un año.

- Suspensión del derecho al cobro hasta un año, parcial o totalmente, de incentivos por los resultados del trabajo, del coeficiente económico-social u otros pagos sujetos al cumplimiento de determinados indicadores o condiciones.
- Suspensión hasta un año del derecho escalafonario cuando los turnos, rutas o condiciones de trabajo estén sujetos a elección.
- Suspensión por un período de hasta un año del derecho a ser elegido o designado en órganos o comisiones del centro de trabajo.
- Pérdida de honores otorgados por méritos en el centro de trabajo.
- Suspensión del vínculo laboral con la entidad por un término de hasta 30 días.
- Traslado temporal a otra plaza de menor remuneración o calificación, o de condiciones laborales distintas por un término de hasta un año con derecho a reintegrarse a su plaza.
- Traslado a otra plaza de menor remuneración o calificación, o de condiciones laborales distintas, con pérdida de la que ocupaba el trabajador.
- Separación definitiva de la entidad. (3)

### **1.3 Sistemas informáticos similares**

En la actualidad son muchas las empresas que hacen uso de herramientas informáticas como los sistemas de gestión y ERP para lograr un mejor desempeño en las mismas. Algunos de estos sistemas serán objeto de estudio en este trabajo para analizar las interfaces de usuario que estos tienen, observando el tipo de información que se maneja y cómo es manejada, además de poder analizar las funciones que presentan, las cuales podrían ser muy útiles en el sistema a desarrollar.

#### **VERSAT Sarasola**

VERSAT Sarasola es un paquete integrado para la gestión económica financiera. Está hecho en Delphi y como base de datos usa SQL.

Contiene 11 módulos: Configuración, Contabilidad General, Costos y Procesos, Control de Inventarios, Control de Activos Fijos, Finanzas, Caja y Banco, Facturación, Planificación Económico/Productiva, Generador de Reportes y Nómina de Salarios.

El subsistema Nómina de Salarios es lo suficientemente adaptable a los diversos sistemas de pago, estimulaciones y demás regulaciones laborales que existen en el país. Da tratamiento a cuatro modalidades de pagos a trabajadores, por tarifa (horaria, diaria), pago libre, vacaciones. Permite la configuración de impuestos, bonificaciones, condiciones anormales, retenciones, descuentos, penalizaciones, entre otras.

#### **Características del VERSAT SARASOLA:**

- Herramienta para la planificación económica, el control y el análisis de gestión.
- Diseñado para su empleo en cualquier tipo de entidad empresarial y presupuestada.
- Permite llevar el control y registro contable individual de todos los hechos económicos que se originan en las estructuras internas de las entidades, así como exponer el estado financiero y toda la información económica y contable en este universo.
- Se estructura en un grupo de subsistemas en los cuales se procesan y contabilizan los documentos primarios, donde se anotan los movimientos, los recursos materiales, laborales y financieros que se utilizan en una entidad.
- Se logra establecer un proceso de interacción usuario-sistema.
- Rapidez y fiabilidad, a partir de la configuración del proceso de contabilización de los documentos primarios y de las propias posibilidades de trabajo contenidas en cada subsistema. (4)

#### **SAP ERP HUMAN CAPITAL MANAGEMENT**

SAP ERP HUMAN CAPITAL MANAGEMENT (SAP ERP HCM) optimiza la contribución de cada empleado coordinando las aptitudes, las actividades y los incentivos de los empleados con los objetivos empresariales y las estrategias necesarias para alcanzarlos. También proporciona herramientas para gestionar, medir y recompensar las contribuciones individuales y colectivas.

SAP ERP HCM proporciona funcionalidades integradas y para toda la empresa que:

- Optimizan los procesos de HCM (human capital management) y los integran a la perfección en todas las operaciones globales.
- Proporcionan acceso en tiempo real a la información que acelera la toma de decisiones por parte del personal.
- Le permiten asignar las personas correctas a los proyectos adecuados y en el momento oportuno.
- Dan soporte tanto a los empleados como a los directivos a lo largo de todo el ciclo de vida del empleado.
- Capacitan a los empleados para gestionar procesos en un entorno colaborativo. (5)

Los principales subcomponentes son los siguientes:

**Gestión de personal:** Centraliza las actividades relacionadas con los recursos humanos y contribuye a las tareas de planificación. Incluye el proceso de contratación de personal y ofrece un marco flexible para crear y gestionar paquetes de beneficios diseñados especialmente.

**Gestión de tiempos:** Contempla los procesos que implican la planificación, registro y valoración del trabajo interno y externo realizado por los empleados, así como los tiempos de ausentismo. Proporciona información a otros procesos empresariales tales como la determinación de costos de personal y la creación de facturas.

**Cálculo de nómina:** Se utiliza para calcular la remuneración de cada empleado. Consta, además, de una serie de procesos derivados de la legislación de cada país, en términos de beneficios sociales, seguridad laboral y asistencia médica.

### **SAGE MAS 500**

Este software ofrece servicios de software de automatización de los procesos de recursos humanos y la nómina, permiten automatizar las actividades y el cumplimiento de estas por los recursos humanos, además de permitir realizar un seguimiento de la asistencia, los empleados y de la nómina.

El módulo que permite realizar el proceso de la nómina es Abra nómina. Este está integrado con el sistema de recursos humanos y permite la gestión del plan de asistencia para actualizar automáticamente con el procesamiento de nóminas. Permite producir cheques de pago en la demanda, y

hacer cambios de última hora. Es adaptable a todas las organizaciones permitiendo incluir los ingresos ilimitados, impuestos y tipos de deducción definidos por el usuario de múltiples grupos a pagar.

Sage ABRA HRMS es un conjunto integrado de soluciones de recursos humanos para ayudar de manera más eficiente y rentable la gestión de sus empleados.

Entre sus funcionalidades se encuentran:

- Simplificar las tareas rutinarias de gestión de los empleados mediante la automatización de su trabajo administrativo.
- Administrar y realizar un seguimiento de su cumplimiento de los requisitos del gobierno.
- Simplificar la contratación para que se pueda obtener el empleado correcto para el trabajo.
- Llevar un registro de asistencia de los empleados.
- Automatizar la formación de los empleados.
- Proporcionar los datos de actualizaciones empleadas y la inscripción de beneficios en línea.(6)

## **ASSETS NS**

ASSETS NS es un Sistema de Gestión Integral, una aplicación cliente-servidor programada en Visual Basic 6.0 y Microsoft SQL Server 2000. Está bastante adaptado a las necesidades del país.

Es un sistema estándar y parametrizado que permite el control de los procesos de Compras, Ventas, Producción, Taller, Inventario, Finanzas, Contabilidad, Presupuesto, Activos Fijos, Útiles y Herramientas y Recursos Humanos. (7)

El módulo Recursos Humanos de ASSETS NS está concebido para calcular las nóminas y controlar los recursos laborales de una entidad.

Entre sus funcionalidades cuenta con gran número de informes relacionados con la actividad de recursos humanos. Entre ellos existen dos fichas para los estadísticos; en donde se agrupan datos estadísticos sobre la composición y comportamiento de la fuerza de trabajo, los salarios y la disciplina laboral. El Análisis de la Disciplina Laboral es un informe del ausentismo en un período determinado, por lo que deben introducirse las fechas inicial y final. Es acumulativo y de acuerdo a las fechas introducidas se puede utilizar para analizar la disciplina de un mes o de todo un período.



### **1.3.1 Resultados del estudio de los sistemas informáticos observados**

La disciplina laboral en una empresa es de vital importancia, el cumplimiento de las políticas, normas y reglamentos de la empresa por parte de todos sus integrantes, son esenciales para trabajar en armonía y con eficiencia en la empresa. Por este motivo, los sistemas informáticos dedicados a la gestión empresarial que existen en la actualidad en el mundo, deben contar con una funcionalidad en el área de capital humano que se encargue de la gestión de la disciplina laboral.

Durante la investigación y estudio de los sistemas en cuestión, se puede observar que cuentan con funcionalidades para la gestión de los recursos humanos y para el cálculo de la nómina, pero la mayoría carece de una funcionalidad que gestione la disciplina laboral. ASSETS NS cuenta con un informe dedicado a la disciplina laboral por concepto del ausentismo de los trabajadores, el cual es descontado a la nómina por diferentes áreas, como por ejemplo subsidio, pero no por concepto de medidas disciplinarias impuestas a estos, por lo cual ninguna de estas soluciones nos brinda información relevante que se pueda utilizar en la construcción de la propuesta de solución del trabajo.

### **1.4 Modelo de desarrollo de software**

Los modelos de desarrollo son un conjunto de actividades, acciones, tareas, fundamentos y productos de trabajo que se requieren para desarrollar software de alta calidad. Son de vital importancia ya que proporcionan estabilidad, control y organización.

Existen varios modelos entre los que se destacan:

- Modelo de cascada
- Incremental
- De procesos
- En espiral
- De desarrollo concurrente
- Basado en componentes
- Iterativo

Los modelos de desarrollo de software se han ido adaptando a las necesidades y características propias de cada proyecto. De ahí que en la actualidad cada ingeniero seleccione un modelo de desarrollo para la ingeniería del software según la naturaleza del proyecto y de la aplicación, los métodos y las herramientas a utilizarse y los controles y entregas que se requieren.

El modelo de desarrollo de software que se va a utilizar describe la secuencia de actividades de alto nivel para la construcción y desarrollo de soluciones. Se logra con la combinación entre los modelos Basado en componentes, el Iterativo y el Incremental. Se emplearán las técnicas de prototipado, si son requeridas, para los requerimientos del usuario de los que no existe una visión clara por parte de estos, con el objetivo de desarrollar una definición mejorada de los requisitos del usuario para el sistema.

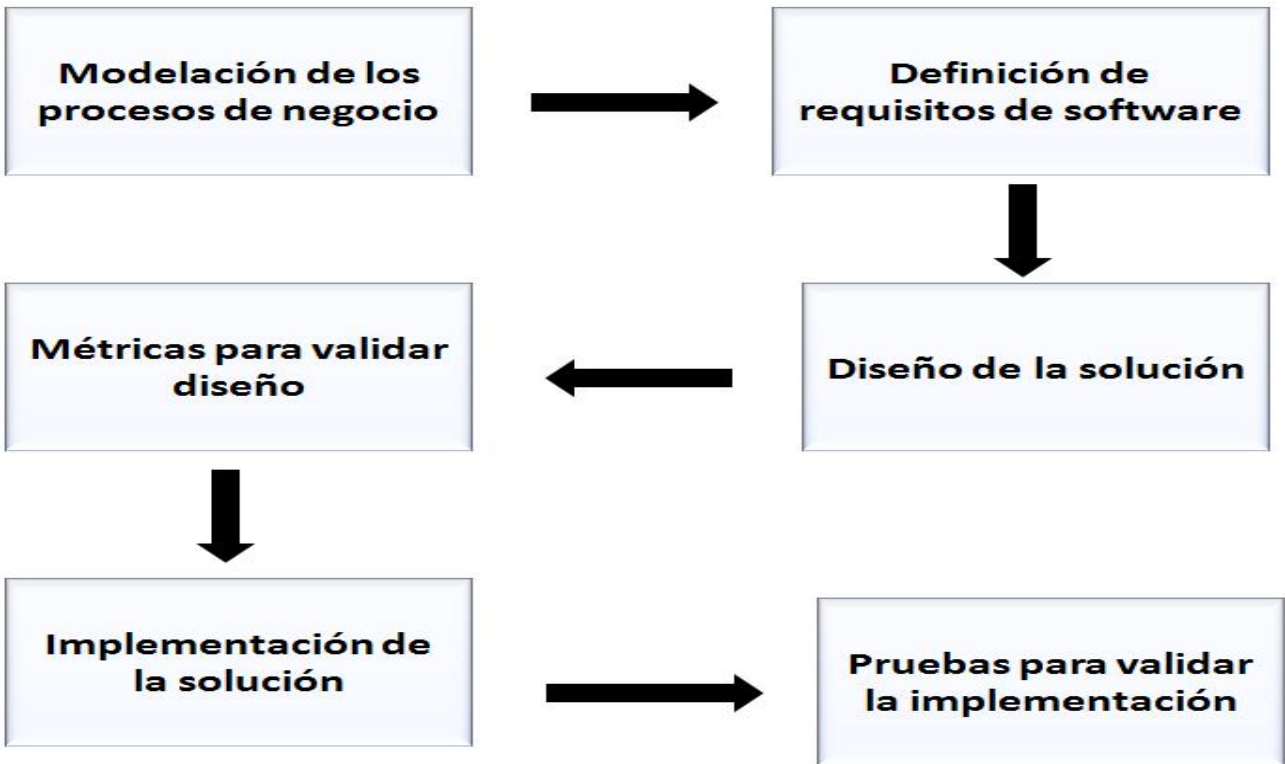
### **Desarrollo iterativo e incremental**

Es un enfoque en el que el ciclo de vida está compuesto por iteraciones, estas son pequeños procesos compuestos de varias actividades cuyo objetivo es entregar una parte del sistema parcialmente completo, probado, integrado y estable. Todo el software es integrado en cada entrega de cada iteración hasta obtener el producto de software completo en la última iteración. En cada iteración se obtiene como resultado un incremento.

### **Desarrollo basado en componentes**

Nos lleva a alcanzar un mayor nivel de reutilización de software, aún en contextos distintos a aquellos para los que fue diseñado. Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados. Cuando existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema. Dado que un componente puede ser construido y luego mejorado continuamente, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo.

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



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

A continuación se definen los artefactos que se generan en el modelo de desarrollo.

#### **Modelación de los procesos de negocio**

- Mapa de procesos.

#### **Definición de requisitos de software**

- Modelo conceptual.
- Listado de los requisitos funcionales.
- Descripción de los requisitos funcionales identificados.

#### **Diseño de la solución**

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

- Diagrama de componentes.

### **Métrica para validar diseño.**

- Resultados de la aplicación de la métrica relación entre clases.
- Resultados de la aplicación de la métrica tamaño operacional de clase.

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

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

## **1.5 Herramientas y tecnologías**

### **1.5.1 Lenguajes de modelado**

A lo largo de los años, durante el desarrollo de los proyectos de software ocurren bastantes confusiones y malas interpretaciones en los requerimientos de los clientes y usuarios, en parte debido a la abundancia de notaciones, metodologías y conceptos que hace que los desarrolladores de sistemas no se pongan de acuerdo en qué es lo que realmente están elaborando. En este sentido, es común encontrar muchos productos de software carentes de diseño formal y entendido sólo por quien lo ha construido.

En cualquier proyecto de ingeniería se requieren etapas de modelado que permitan experimentar y visualizar qué se construirá. Un modelo no es más que una representación simplificada de la realidad que ayuda a entender un sistema grande y complejo que no puede ser comprendido fácilmente en su totalidad.

Un lenguaje de modelado proporciona los elementos básicos con los que escribir un modelo. Estos en las empresas que se dedican a la creación de software han tomado una gran importancia debido a que el modelado es una parte central de todas las actividades que conducen a la producción de buen software.

A través del modelado se consigue:

- Visualizar cómo es que queremos que sea un sistema.
- Especificar la estructura o el comportamiento de un sistema.
- Proporcionar plantillas que guían en la construcción de un sistema.

- Documentar las decisiones que se toman.

#### **1.5.1.1 Lenguaje unificado de modelado (UML)**

UML es un lenguaje estándar para escribir planos de software, que se utiliza para visualizar, especificar, construir y documentar los artefactos de un mismo sistema que involucra una gran cantidad de software. (8)

UML sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura hardware donde se ejecuten.

Otro objetivo de este modelado visual es que sea independiente del lenguaje de implementación, de tal forma que los diseños realizados usando UML se puedan implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos).

UML es además un método formal de modelado. Esto aporta las siguientes ventajas:

- Mayor rigor en la especificación.
- Permite realizar una verificación y validación del modelo realizado.
- Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño de más alto nivel, de la estructura de un proyecto. (9)

#### **1.5.1.2 Notación para el modelado de los procesos del negocio (BPMN)**

La Notación para el Modelado de Procesos de Negocio es un estándar nuevo para el flujo de procesos del modelado de negocios y los servicios web. BPMN define un Diagrama de Procesos de Negocio basado en la técnica de diagramado de flujos que ajusta modelos gráficos de operación de procesos de negocio. Este diagrama está estructurado en categorías básicas de elementos, entre las que se encuentran:

- Objetos de flujo.
- Objetos de conexión.
- Artefactos.

- Carriles.

### **1.5.2 Herramientas de modelado**

En los últimos años el desarrollo del software ha propiciado que surjan herramientas que permitan a los analistas y diseñadores de software realizar estos procesos de forma eficiente, con mayor calidad y fiabilidad. Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son un ejemplo de ello.

Las Herramientas CASE son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un Software. (10)

En la actualidad existen una gran cantidad de herramientas CASE. Entre las más utilizadas se encuentran Platinun Erwin, EasyCASE, Oracle Designer, System Architect y Visual Paradigm para UML. Para la realización de la propuesta de solución del trabajo se va a utilizar Visual Paradigm para UML ya que este soporta los dos lenguajes de modelados que se utilizarán en el trabajo, los cuales son UML y BPMN.

#### **1.5.2.1 Visual Paradigm para UML**

Visual Paradigm para UML (VP-UML) es una herramienta UML de diseño UML y herramienta CASE diseñada para la ayuda al desarrollo de software. VP-UML soporta estándares claves de la industria, tales como Lenguaje de Modelado Unificado (UML), SysML, BPMN y XMI. Ofrece un completo conjunto de herramientas de los equipos de desarrollo de software necesario para la captura de requisitos, la planificación de programas, la planificación de controles, modelado de clase y modelado de datos. (11)

El uso de esta herramienta trae consigo grandes ventajas, ya que esta soporta el ciclo de vida completo del desarrollo de software, ayuda a una más rápida construcción de aplicaciones de calidad y permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Es una herramienta colaborativa porque soporta a varios usuarios trabajando en un mismo proyecto, genera la documentación del proyecto automáticamente en varios formatos como son Web o Pdf y permite control de versiones. Brinda la posibilidad de generar código a partir de los diagramas, para plataformas

como .Net, Java y PHP, así como obtener diagramas a partir de código. Esta es precisamente una gran ventaja puesto que el sistema al que se debe integrar la propuesta de solución está desarrollado en PHP.

VP-UML es multiplataforma, lo cual le permite al usuario utilizar esta herramienta en varios sistemas operativos como Windows, Linux, Unix y otros; además se encuentra disponible en distintas versiones: Enterprise, Professional, Standard, Modeler, Personal y Community.

### **1.5.3 Lenguajes de programación y frameworks**

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo. (12)

El término framework se refiere a una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Entre los objetivos principales que se persigue con su uso están: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Para la realización de la propuesta de solución del trabajo se utilizarán los lenguajes de programación y framework definidos en el marco de trabajo del subsistema capital humano de Cedrux, con el objetivo de garantizar una correcta integración con lo que se ha desarrollado hasta el momento en el subsistema. Se utilizarán PHP y JavaScript como lenguajes de programación y los frameworks ExtJS, Doctrine y Zend Framework.

#### **1.5.3.1 JavaScript**

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

#### **1.5.3.2 ExtJS**

ExtJS es un conjunto de librerías JavaScript que permite el desarrollo de aplicaciones RIA (Aplicaciones Ricas de Internet) basadas en un navegador. Ofrece al desarrollador un gran conjunto de widgets (componentes como por ejemplo grids, ventanas de diálogo) plenamente integrados y un API (interfaz de programación de aplicaciones) para conseguir interfaces web más dinámicas e interactivas con el usuario.

ExtJS usa el lenguaje JavaScript junto con HTML para la creación de las interfaces de usuario, así como para el manejo de eventos en cada una de las páginas que comprenden una aplicación desarrollada con ExtJS.

Al usar JavaScript, ExtJS trae consigo los beneficios propios del lenguaje tal como:

- La orientación a objetos,
- La manipulación del DOM (ExtJS extiende esta capacidad con su propia implementación para el manejo de DOM)
- El soporte de múltiples navegadores como Internet Explorer, Opera, Safari y Mozilla Firefox. (13)

Esta librería incluye:

- Componentes UI de alto performance y personalizables.
- Modelo de componentes extensibles.
- Un API (interfaz de programación de aplicaciones) fácil de usar.
- Licencias de código abierto y comercial. (14)

La versión 2.2 que es la que se va a utilizar, tiene una alta capacidad de soporte de navegadores, mantiene la misma apariencia independientemente del navegador y la plataforma del cliente. Se integra con aplicaciones desarrolladas utilizando la especificación JEE o con aplicaciones desarrolladas con .Net o PHP.

### **1.5.3.3 PHP**

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.



PHP es un lenguaje de script, diseñado para, entre otras cosas, aumentar e incrementar el dinamismo de las páginas web. Sus características han ido creciendo hasta convertirse en un lenguaje de programación completo, capaz de manejar entornos que integran grandes bases de datos. Su popularidad se basa, en gran parte, en su sintaxis similar a la del lenguaje de programación C, su rapidez y simplicidad. (15)

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

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

#### **1.5.3.4 Doctrine**

El Proyecto doctrina es la casa de un conjunto seleccionado de librerías PHP, se centra principalmente en la prestación de servicios de persistencia y funcionalidad.

El mapeador objeto relacional (ORM) para PHP se encuentra en la parte superior de una capa de abstracción de base de datos de gran alcance (DBAL). Una de sus principales características es la posibilidad de escribir consultas de base de datos en un dialecto orientado a objetos de propiedad SQL llamada Doctrine (Lenguaje de consulta DQL), inspirado en hibernates HQL. Esto proporciona a los desarrolladores una poderosa alternativa a SQL que mantiene la flexibilidad sin necesidad de duplicar código innecesario. La potente capa de abstracción de base de datos tiene muchas características de base de datos de la introspección de esquema, la gestión de esquema y la abstracción DOP. (16)

#### **1.5.3.5 Zend Framework**

Zend Framework se trata de un framework de código abierto para desarrollo de aplicaciones y servicios Web con PHP5. Utiliza código 100% orientado a objetos y brinda soluciones para construir sitios web modernos, robustos y seguros. La estructura de los componentes de Zend Framework es única; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura, débilmente acoplada, permite a los desarrolladores utilizar los componentes por separado.

Aunque se pueden utilizar de forma individual, los componentes de la biblioteca estándar de Zend Framework conforman un potente y extensible framework de aplicaciones web al combinarse. Además, ofrecen gran rendimiento y robusta implementación modelo-vista-controlador (MVC), abstracción de base de datos fácil de usar y un componente de formularios que implementa: la prestación de formularios HTML y validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos.

#### **1.5.4 Gestor de base de datos**

En un ERP es imprescindible contar con una base de datos para la persistencia de los datos. Por lo que es necesario el uso de una herramienta que garantice la gestión de la misma. Para esto se hará uso de un sistema gestor de base de datos.

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. Por tanto debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes. (17)

Dentro de los numerosos SGBD que se utilizan en la actualidad para la gestión de bases de datos se encuentra PostgreSQL.

##### **1.5.4.1 PostgreSQL**

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, desarrollado en la Universidad de California en Berkeley del Departamento de Ciencias de la Computación. Postgres fue pionera en muchos conceptos que sólo estuvo disponible en algunos sistemas de bases de datos comerciales mucho más tarde.

PostgreSQL es un descendiente de código abierto del código original de Berkeley. Soporta una gran parte del estándar SQL y ofrece muchas características modernas:

- consultas complejas

- claves externas
- desencadenantes
- puntos de vista
- la integridad transaccional
- control de concurrencia multiversión

Debido a la licencia de liberales, PostgreSQL puede ser usado, modificado y distribuido por cualquier persona de forma gratuita para cualquier propósito, ya sea privado, comercial o académico. (18)

### **1.5.5 Herramientas de base de datos**

#### **1.5.5.1 EMS SQL Manager**

EMS SQL Manager para PostgreSQL es una herramienta de alto rendimiento para la administración y el desarrollo de bases de datos PostgreSQL. Funciona con cualquier versión de PostgreSQL hasta la más reciente y soporta sus últimas características. (19)

#### **1.5.6 Navegador Web**

La propuesta de solución al problema expuesto será realizada como una aplicación web, por lo que es necesario contar con un navegador para interactuar con su contenido.

Un navegador o navegador web es un programa que permite ver la información que contiene una página web. El navegador interpreta el código en el que está escrita la página web y lo presenta en pantalla permitiendo al usuario interactuar con su contenido y navegar hacia otros lugares mediante enlaces o hipervínculos.

##### **1.5.6.1 Mozilla Firefox**

Se utilizará Mozilla Firefox ya que este es un navegador de software libre. Entre sus características se encuentran que presenta una forma rápida y eficiente de navegar por la web y que permite abrir varias páginas en una misma ventana mediante el empleo de pestañas separadas.

#### **1.5.7 Control de versiones**

El uso de un control de versiones para el desarrollo del sistema es de vital importancia, debido a que el sistema es programado a la vez por diferentes desarrolladores, por lo cual es necesario contar con una herramienta que guarde todos los cambios hechos a los ficheros y directorios.

Se llama control de versiones a los métodos y herramientas disponibles para controlar todo lo referente a los cambios en el tiempo de un archivo.

Difícilmente un archivo de código o un documento de texto están terminados con la primera escritura; necesita cambios o reescrituras para corregir errores, modificar su contenido, a medida que el documento cambia existen dos opciones, mantener un historial de cambios o dejar que evolucione sin memoria. El control de versiones es un método estándar para mantener esta memoria, haciendo además que sea útil para el desarrollo futuro. (20)

#### **1.5.7.1 Subversion**

Subversion es un sistema de control de versiones libre y de código fuente abierto, es decir, maneja ficheros y directorios a través del tiempo. Tiene un árbol de ficheros en un repositorio central. El repositorio es como un servidor de ficheros ordinario, excepto porque recuerda todos los cambios hechos a sus ficheros y directorios. Esto le permite recuperar versiones antiguas de sus datos, o examinar el historial de cambios de los mismos. Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos ordenadores. Entre las características que tienen estos sistemas están:

- Entendimiento con los lenguajes de programación.
- Suministro de herramientas para la construcción de software.
- Pueden ser usado para administrar cualquier conjunto de ficheros.

#### **TortoiseSVN**

TortoiseSVN es un cliente gratuito de código abierto para el sistema de control de versiones Subversion. Maneja ficheros y directorios a lo largo del tiempo. Es un sistema general que puede ser utilizado para manejar cualquier colección de ficheros, incluyendo código fuente. (21)

## **1.5.8 Otras herramientas teóricas**

### **1.5.8.1 Técnicas empleadas para la captura de requisitos**

Existen técnicas y herramientas para el levantamiento de requisitos que permiten obtener, documentar y desarrollar los requisitos de una forma más sencilla. Muchas de estas técnicas son fundamentales para capturar los requisitos de software, permitiendo entregar una especificación de requisitos de software correcta y completa del sistema a automatizar. A continuación se muestran las empleadas para la captura de los requisitos.

- Entrevistas
- Prototipos
- Glosarios
- Arqueología de documentos
- Sistemas existentes

### **1.5.8.2 Patrón de análisis**

Cada patrón es una regla compuesta por tres partes que expresa la relación entre cierto contexto, un problema y una solución. El concepto de patrón es aplicable a la arquitectura de software y también al diseño de interfaz. (22)

#### **Patrones CRUD**

El patrón CRUD (Create, Read, Update, Delete [Crear, Leer, Actualizar, Eliminar]) se utiliza en los casos donde se quiere realizar altas, bajas, cambios y consultas a alguna entidad del sistema.

Es conocido como el padre de todos los patrones de capa de acceso. Describe que cada objeto debe ser creado en la base de datos para que sea persistente. Una vez creado, la capa de acceso debe tener una forma de leerlo para poder actualizarlo o simplemente borrarlo. (23)

### **1.5.8.3 Patrón de arquitectura**

Los patrones de arquitectura están relacionados a la interacción de objetos dentro o entre niveles arquitectónicos. Resuelven problemas de adaptabilidad a requerimientos cambiantes, performance,

modularidad y acoplamiento. Dando solución a llamadas entre objetos (similar a los patrones de diseño), decisiones y criterios arquitectónicos y empaquetado de funcionalidades.

### **Modelo vista controlador (MVC)**

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. Su principal finalidad es mejorar la reusabilidad y que las modificaciones en las vistas impacten en menor medida en la lógica de negocio o de datos.

El Controlador es responsable de:

- Recibir los eventos de entrada.
- Contiene reglas de gestión de eventos, estas acciones pueden suponer peticiones al modelo o a las vistas.

Las Vistas son responsables de:

- Recibir datos del modelo y mostrarlos al usuario.
- Tienen un registro de su controlador asociado.
- Pueden dar el servicio de "Actualización()", para que sea invocado por el controlador o por el modelo cuando es un modelo activo.

El MVC es un patrón que convierte una aplicación en un paquete mantenible, modular y de desarrollo rápido. La modularidad y el diseño independiente permiten a los desarrolladores y diseñadores hacer cambios en alguna parte de la aplicación sin afectar a los demás.

#### **1.5.8.4 Patrones de diseño**

Un patrón define una posible solución correcta para un problema de diseño dentro de un contexto dado, describiendo las cualidades invariantes de todas las soluciones.

Los patrones de diseño pretenden:

- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.

- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño.

### **Patrones Grasp (Patrones generales de software para asignar responsabilidades)**

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

Existen nueve patrones Grasp los cuales son: Experto, Creador, Alta Cohesión, Bajo Acoplamiento, Controlador, Polimorfismo, Fabricación Pura, Induración y No Hables con Extraños. De estos se utilizarán Experto, Creador, Alta Cohesión, Bajo Acoplamiento y Controlador con el fin de que contribuya a que el sistema sea más robusto y flexible.

#### **Bajo Acoplamiento**

Consiste en tener las clases lo menos ligadas entre sí que se pueda, de tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización y disminuyendo la dependencia entre las clases. Mantener el bajo acoplamiento entre clases más que un patrón en una buena práctica del diseño, dado que al realizar cambios no se afectan otros componentes, es más fácil de entender de manera aislada y de esta forma se pueden reutilizar mejor las clases.

#### **Alta Cohesión**

Este patrón nos dice que la información que almacena una clase debe de ser coherente y está en la mayor medida de lo posible relacionada con la clase. Realizando un diseño donde las clases mantengan una alta cohesión se mejora la claridad y facilidad con que se entiende el diseño, se simplifica el mantenimiento y las mejoras de funcionalidad, a menudo se genera un bajo acoplamiento y soporta mayor capacidad de reutilización.

#### **Experto**

Se pone en práctica con el uso de clases que poseen responsabilidades específicas a cumplir de acuerdo con la información que manejan.

## **Creador**

Es útil contar con el principio general para la asignación de responsabilidades de creación porque permite que el diseño pueda soportar bajo acoplamiento, mayor claridad, encapsulación y reutilización. Las clases controladoras son responsables de crear el objeto de las modelos y estas a su vez de las entidades.

## **Controlador**

Se utilizan cuando la aplicación es muy extensa, de esta forma, en vez de tener un solo controlador y saturarlo, se tienen clases Controllers, que son controladores más pequeños especializados en las funcionalidades de cada componente.

## **Patrones Gof (Gang of Four)**

Los patrones Gof son patrones de diseño publicados en el libro Design Patterns: Elements of Reusable Object-Oriented Software por Gamma, Helm, Jonson y Vlissides conocidos mundialmente por Gang of Four o Pandilla de los cuatro. En este libro se encuentran recopilados un total de 23 patrones clasificados en patrones creacionales, estructurales y de comportamiento.

Se utilizará en el diseño de la solución el patrón estructural fachada.

## **Fachada**

Provee de una interfaz unificada simple para el acceso de una interfaz o grupo de interfaces de un subsistema. Un ejemplo de su aplicación es en el uso de los servicios, donde la relación existente entre las clases controladoras y los servicios permite acceder a métodos que no están implementados en el componente y que se encuentran en otros componentes pertenecientes al subsistema Capital Humano.

## **1.6 Conclusiones del capítulo**

En este capítulo se trataron conceptos muy importantes con el objetivo de tener un mejor entendimiento del negocio. Se realizó un estudio de algunos de los sistemas existentes en la actualidad, teniendo en cuenta sus interfaces y funcionalidades, así como los inconvenientes que presentan y por qué no se tienen en cuenta para la construcción de la propuesta de solución. Por último se presentan las herramientas y lenguajes a utilizar, los cuales contribuirán al desarrollo de la solución que se quiere construir. Al concluir este capítulo se puede comenzar a desarrollar la propuesta de solución del trabajo.



## CAPÍTULO 2: MODELAMIENTO DEL NEGOCIO Y ANÁLISIS

### 2.1 Introducción

En el presente capítulo se realiza un estudio preliminar del sistema. Una vez identificados los procesos de negocio se procede a realizar el modelo conceptual con el objetivo de identificar las entidades del negocio. Se identifican y describen los requisitos funcionales que debe satisfacer la solución a construir.

### 2.2 Modelación del negocio

#### 2.2.1 Descripción del negocio

Dentro del proceso de nóminas de todas las entidades laborales de la economía cubana, ya sean pertenecientes al sector empresarial como al presupuestado, se tiene en cuenta la gestión de la disciplina laboral por la incidencia directa que tienen los importes económicos impuestos a los trabajadores como medida disciplinaria.

A raíz de las investigaciones y estudios realizados se concluyó que era necesario desarrollar una solución informática que permita de forma flexible, segura y con la calidad requerida, la gestión de la disciplina laboral que cumpla con todas las exigencias y requerimientos establecidos en la legislación laboral vigente.

#### 2.2.2 Procesos del negocio

Un proceso de negocio es un conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de negocio definido.

El proceso de negocio que se identificó fue la **disciplina laboral**.

#### 2.2.3 Descripción de los procesos del negocio

La gestión de la disciplina laboral se inicia con el registro de las violaciones y las medidas disciplinarias, en la cual se registran las faltas cometidas, la gravedad de estas y las medidas impuestas. Por cada trabajador, que haya incurrido en alguna acción antirreglamentaria, se recoge en un registro de medidas disciplinarias la medida impuesta, si trae consigo un importe económico y la apelación en caso de que apele, además se puede filtrar el listado de los registro de medidas por un periodo determinado.

En la figura 2 se muestra el mapa de procesos del proceso disciplina laboral.

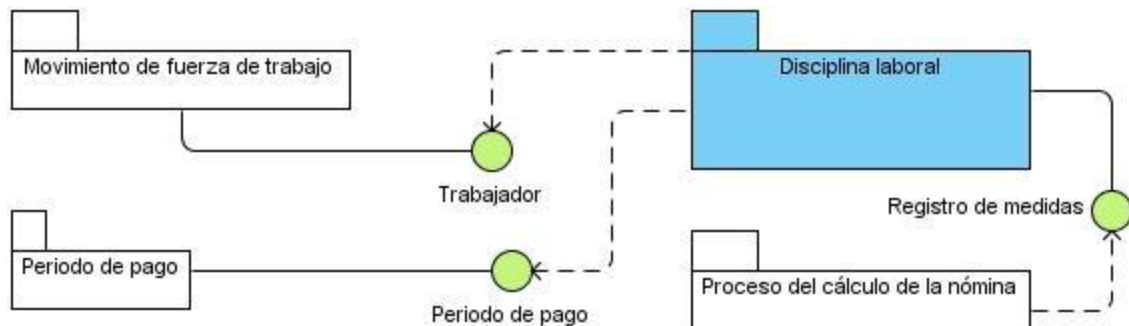


Figura 2: Mapa de procesos

### 2.3 Modelo conceptual

Un modelo conceptual es un diagrama que ilustra una serie de relaciones entre ciertos factores que se cree impactan o conducen a una condición de interés. (24)

En la figura 3 se muestra el modelo conceptual que representa las diferentes entidades existentes en el proceso de la disciplina laboral del subsistema Capital Humano del Sistema Integral de Gestión CedruX.

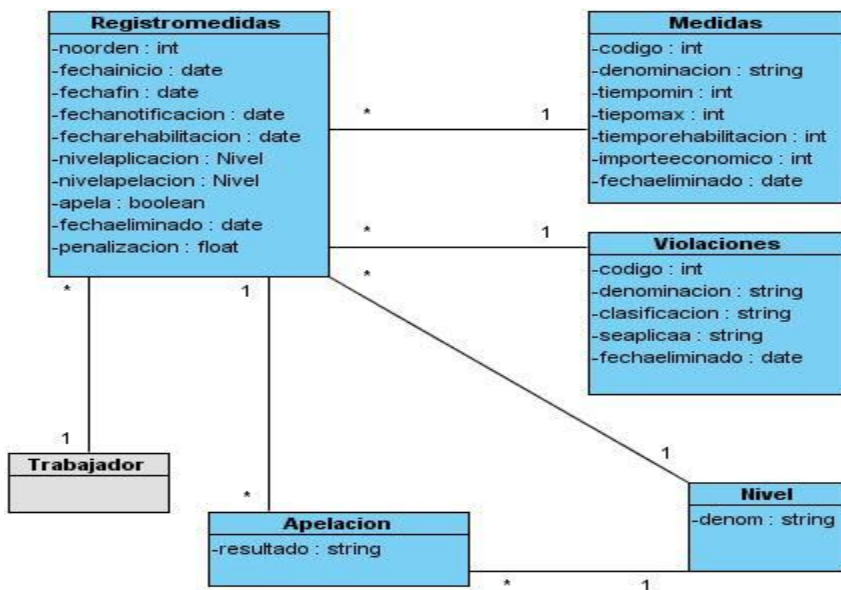


Figura 3: Modelo conceptual

A continuación se realiza una descripción de las clases presentes en el modelo conceptual.

## Violaciones

<b>Descripción</b>		Las violaciones son aquellas conductas impropias recogidas en el reglamento.						
<b>Atributos</b>								
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones			
					Clases válidas	Clases no válidas		
código	Número que identifica a la violación.	entero	No	Si	Números			
denominación	Denominación de la violación.	cadena	No	Si	Letras	y		
se aplica a	Guarda a quien se dirige la violación.	cadena	No	No	números	y		
clasificación	(Leve, grave, muy grave)	cadena	No	No	Letras	y		
					números			

Tabla 1: Descripción de la clase Violaciones

## Medidas

<b>Descripción</b>		Listado de las medidas disciplinarias de la entidad según lo establecido en el reglamento.						
<b>Atributos</b>								
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones			
					Clases válidas	Clases no válidas		
Código	Número de 3 dígitos que identifica la medida.	entero	No	Si	números			
denominación	Descripción de la medida	cadena	No	Si	Letras	y		
Tiempo mínimo	Tiempo mínimo de duración de la medida	entero	No	No	números			
Tiempo máximo	Tiempo máximo de duración de la		No	No	números			

	medida					
tiempo de rehabilitación	Después de este tiempo debe ser eliminada por completo la medida.	entero	No	No	números	
importe económico	Determina si se aplica un importe económico o no.	boolean	No	No	Si o no.	

Tabla 2: Descripción de la clase medidas

### Registro de medidas disciplinarias

<b>Descripción</b>	Registro de las medidas disciplinarias aplicadas en la entidad.					
<b>Atributos</b>						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
número de orden	Número de orden en el registro anual.	entero	No	Si	Números	
fecha de notificación	Fecha de notificación al trabajador de la medida.	fecha	No	No	fecha	
fecha de inicio	Fecha de inicio de la medida.	fecha	No	No	fecha	
fecha de fin	Fecha final de la medida.	fecha	No	No	fecha	
fecha de rehabilitación	Fecha de rehabilitación de la medida.	fecha	Si	No	fecha	
nivel de aplicación	Director general del Grupo Empresarial Director	cadena	No	No	Letras y números	

	general de la Entidad Director de la UEB					
apelación	Si se decide apelar o no.	cadena	No	No	Si o No	
nivel de apelación	Nivel al que se apela la medida.	cadena	No	No	Letras y números	
resultado de la apelación	Resultado de la apelación de la medida	cadena	No	No	Letras y números	
penalización	Importe económico impuesto al trabajador	float	Si	No	Números	

**Tabla 3: Descripción de la clase registro de medidas**

### Apelación

<b>Descripción</b>	Registro de las apelaciones					
<b>Atributos</b>						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
Resultado	Resultado de la apelación de la medida.	cadena	No	No	Clases válidas y números	Clases válidas no válidas cadena

**Tabla 4: Descripción de la clase apelación**

### Nivel

<b>Descripción</b>	Registra el nivel de aplicación y apelación de las apelaciones					
<b>Atributos</b>						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
denominación	Descripción del nivel	cadena	No	Si	Clases válidas y números	Clases válidas no válidas

**Tabla 5: Descripción de la clase nivel**

## 2.4 Requisitos funcionales

Un requerimiento es la condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. Por lo que se puede decir que los requerimientos deben ser especificados por escrito, claros y precisos. Estos se clasifican en funcionales y no funcionales. Los requisitos funcionales son capacidades o condiciones que el sistema debe tener, mientras que los no funcionales son propiedades o cualidades que el producto debe tener. A continuación se muestran los 17 requisitos funcionales identificados en el negocio:

<b>RF1</b>	<b>Gestionar violaciones disciplinarias</b>
	<ul style="list-style-type: none"> <li>• Adicionar violación</li> <li>• Modificar violación</li> <li>• Eliminar violación</li> <li>• Listar violación</li> <li>• Imprimir violación</li> </ul>
<b>RF2</b>	<b>Gestionar medidas disciplinarias</b>
	<ul style="list-style-type: none"> <li>• Adicionar medida disciplinaria</li> <li>• Modificar medida disciplinaria</li> <li>• Eliminar medida disciplinaria</li> <li>• Listar medida disciplinaria</li> <li>• Imprimir medida disciplinaria</li> </ul>
<b>RF3</b>	<b>Gestionar registro de medidas disciplinarias</b>
	<ul style="list-style-type: none"> <li>• Adicionar registro de medida disciplinaria</li> </ul>

	<ul style="list-style-type: none"> <li>• Modificar registro de medida disciplinaria</li> <li>• Eliminar registro de medida disciplinaria</li> <li>• Imprimir registro de medida disciplinaria</li> <li>• Listar registro de medida disciplinaria</li> <li>• Filtrar registro de medida disciplinaria por período</li> <li>• Registrar apelación</li> </ul>
--	--

**Tabla 6: Requisitos funcionales**

### 2.4.1 Especificaciones de requisitos

En el presente epígrafe se especifican los requisitos vistos anteriormente; el propósito fundamental de la especificación de los requisitos es guiar hacia el desarrollo correcto del sistema.

#### 2.4.1.1 Requisito funcional gestionar violaciones disciplinarias

##### Especificación del requisito adicionar violación

<b>Precondiciones</b>	Se ha registrado al menos una violación
<b>Flujo de eventos</b>	
<b>Flujo básico</b>	
1	Se introducen los datos de la violación Código: Numero que identifica a la violación Denominación: Denominación de la violación Clasificación: Descripción de la violación. Se aplica a: A quien se aplica.
2	El sistema valida (ver validación 1) los datos introducidos.
3	Si los datos son correctos el sistema los registra.
4	El sistema confirma el registro de los datos.
5	Concluye el requisito.
<b>Pos-condiciones</b>	
1	Se registró en el sistema una nueva violación.
<b>Flujos alternativos</b>	
<b>Flujo alternativo 4.a Información errónea</b>	
1	El sistema señala los datos erróneos y permite corregirlos.
2	El usuario corrige los datos.

3	Volver al paso 2 del flujo básico.	
<b>Pos-condiciones</b>		
1	N/A	
<b>Flujo alternativo 4.b Información incompleta</b>		
1	El sistema señala los datos vacíos y permite corregirlos.	
2	El usuario corrige los datos.	
3	Volver al paso 2 del flujo básico.	
<b>Pos-condiciones</b>		
1	N/A	
<b>Flujo alternativo *.a El usuario cancela la acción</b>		
1	Concluye el requisito.	
<b>Pos-condiciones</b>		
1	No se registran los datos.	
<b>Validaciones</b>		
1	Se validan los datos según lo establecido en el Modelo conceptual administración de capital humano.	
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A.
<b>Conceptos</b>	<b>Violación</b>	Ver en la interfaz: Código Denominación Clasificación Se aplica a
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

Tabla 7: Especificación del requisito adicional violación

### Especificación del requisito modificar violación

<b>Precondiciones</b>	Se ha registrado al menos una violación.
<b>Flujo de eventos</b>	
<b>Flujo básico</b>	
1	Se selecciona la violación a modificar.
2	El sistema muestra y permite editar los datos de la violación.
3	Se introducen los datos de la violación a modificar. Código: Numero que identifica a la violación Denominación: Denominación de la violación Clasificación: Descripción de la violación. Se aplica a: A quien se aplica.



4	El sistema valida (ver validación 1) los datos introducidos.	
5	Si los datos son correctos el sistema los registra.	
6	El sistema confirma el registro de los datos.	
7	Concluye el requisito.	
<b>Pos-condiciones</b>		
1	Se modificaron los datos de la violación..	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 4.a Información errónea</b>		
1	El sistema señala los datos erróneos y permite corregirlos.	
2	El usuario corrige los datos.	
3	Volver al paso 4 del flujo básico.	
<b>Pos-condiciones</b>		
1	N/A	
<b>Flujo alternativo 4.b Información incompleta</b>		
1	El sistema señala los datos vacíos y permite corregirlos.	
2	El usuario corrige los datos.	
3	Volver al paso 4 del flujo básico.	
<b>Pos-condiciones</b>		
1	N/A	
<b>Flujo alternativo *.a El usuario cancela la acción</b>		
1	Concluye el requisito.	
<b>Pos-condiciones</b>		
1	No se registran las modificaciones realizadas.	
<b>Validaciones</b>		
1	Se validan los datos según lo establecido en el Modelo conceptual Administración de capital humano.	
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A
<b>Conceptos</b>	<b>Violación</b>	Ver en la interfaz: Código Denominación Clasificación Se aplica a
<b>Requisitos especiales</b>	N/A.	
<b>Asuntos pendientes</b>	N/A.	

Tabla 8: Especificación del requisito modificar violación

### Especificación del requisito eliminar violación

<b>Precondiciones</b>	Se ha registrado al menos una violación.	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
1	Se selecciona la violación a eliminar.	
2	El sistema verifica (ver validación 1) que se pueda eliminar la violación.	
3	Se solicita confirmación para eliminar la violación.	
4	Si el usuario confirma se elimina la violación.	
5	El sistema confirma la eliminación.	
6	Concluye el requisito.	
<b>Pos-condiciones</b>		
1	Se eliminó la violación.	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 3. No se puede eliminar la violación</b>		
1	El sistema notifica por qué no puede eliminarse la violación.	
<b>Pos-condiciones</b>		
1	N/A	
<b>Flujo alternativo *.a El usuario cancela la acción</b>		
1	Concluye el requisito.	
<b>Pos-condiciones</b>		
1	No se elimina la violación.	
<b>Validaciones</b>		
1	Se validan que no se haya realizado ningún registro de violación con este tipo de incidencia.	
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A
<b>Conceptos</b>	<b>Violación</b>	Código Denominación Clasificación Se aplica a
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

Tabla 9: Especificación del requisito eliminar violación

### Especificación del requisito listar violación

<b>Precondiciones</b>	Se ha registrado al menos una violación.	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
1	El sistema muestra un listado de las violaciones.	

	Se muestra en un listado el código, la denominación, la clasificación y se aplica a.	
2	Concluye el requisito.	
<b>Pos-condiciones</b>		
1	N/A	
2		
<b>Flujos alternativos</b>		
<b>Flujo alternativo</b>		
1	N/A	
<b>Pos-condiciones</b>		
1	N/A	
<b>Validaciones</b>		
1	N/A	
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A
<b>Conceptos</b>	<b>Violación</b>	Código Denominación Clasificación Se aplica a
<b>Requisitos especiales</b>	N/A.	
<b>Asuntos pendientes</b>	N/A	

Tabla 10: Especificación del requisito listar violación

### Especificación del requisito imprimir violación

<b>Precondiciones</b>	Se ha registrado al menos una violación en el sistema.	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
1	El sistema permite seleccionar los elementos que se desean imprimir.	
2	El sistema permite seleccionar el formato en que se desea imprimir y luego se imprime.	
3	Concluye el requisito.	
<b>Pos-condiciones</b>		
1	N/A	
<b>Flujos alternativos</b>		
<b>Flujo alternativo *.a El usuario cancela la acción</b>		
1	Concluye el requisito.	
<b>Pos-condiciones</b>		
1	No se imprime la información.	

<b>Flujo alternativo 2.a El sistema permite realizar una vista previa</b>		
1	El sistema permite mostrar una vista previa de lo que se desea imprimir.	
<b>Pos-condiciones</b>		
1	Se muestra la información.	
<b>Flujo alternativo * No se pueden imprimir los datos por errores en el sistema, la red o la impresora.</b>		
1	El sistema notifica al usuario que los datos no pueden ser impresos.	
<b>Pos-condiciones</b>		
1	N/A.	
<b>Validaciones</b>		
1	N/A.	
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A
<b>Conceptos</b>	<b>Violación</b>	Código Denominación Clasificación Se aplica a
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

Tabla 11: Especificación del requisito imprimir violación

#### 2.4.1.2 Requisito funcional gestionar medidas disciplinarias

##### Especificación del requisito adicionar medida

<b>Precondiciones</b>	Se ha registrado al menos una medida en el sistema.
<b>Flujo de eventos</b>	
<b>Flujo básico</b>	
6	Se introducen los datos del registro de la medida  Código: Numero que identifica a la violación Denominación: Denominación de la violación Clasificación: Descripción de la violación. .
7	El sistema valida (ver validación 1) los datos introducidos.
8	Si los datos son correctos el sistema los registra.
9	El sistema confirma el registro de los datos.
10	Concluye el requisito.

<b>Pos-condiciones</b>		
2	Se registró en el sistema una nueva medida.	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 4.a Información errónea</b>		
4	El sistema señala los datos erróneos y permite corregirlos.	
5	El usuario corrige los datos.	
6	Volver al paso 2 del flujo básico.	
<b>Pos-condiciones</b>		
2	N/A	
<b>Flujo alternativo 4.b Información incompleta</b>		
4	El sistema señala los datos vacíos y permite corregirlos.	
5	El usuario corrige los datos.	
6	Volver al paso 2 del flujo básico.	
<b>Pos-condiciones</b>		
2	N/A	
<b>Flujo alternativo *.a El usuario cancela la acción</b>		
2	Concluye el requisito.	
<b>Pos-condiciones</b>		
2	No se registran los datos.	
<b>Validaciones</b>		
2	Se validan los datos según lo establecido en el Modelo conceptual administración de capital humano.	
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A.
<b>Conceptos</b>	<b>Medida disciplinaria</b>	Ver en la interfaz: Código Denominación Clasificación
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

Tabla 12: Especificación del requisito adicionar medida

**Especificación del requisito modificar medida**

<b>Precondiciones</b>	Se ha registrado al menos una medida.	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
8	Se selecciona la medida a modificar.	
9	El sistema muestra y permite editar los datos de la medida.	
10	Se introducen los datos del registro de la medida disciplinaria. Código: Numero que identifica a la violación Denominación: Denominación de la violación Clasificación: Descripción de la violación.	
11	El sistema valida (ver validación 1) los datos introducidos.	
12	Si los datos son correctos el sistema los registra.	
13	El sistema confirma el registro de los datos.	
14	Concluye el requisito.	
<b>Pos-condiciones</b>		
2	Se modificaron los datos de la medida.	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 4.a Información errónea</b>		
4	El sistema señala los datos erróneos y permite corregirlos.	
5	El usuario corrige los datos.	
6	Volver al paso 4 del flujo básico.	
<b>Pos-condiciones</b>		
2	N/A	
<b>Flujo alternativo 4.b Información incompleta</b>		
4	El sistema señala los datos vacíos y permite corregirlos.	
5	El usuario corrige los datos.	
6	Volver al paso 4 del flujo básico.	
<b>Pos-condiciones</b>		
2	N/A	
<b>Flujo alternativo *.a El usuario cancela la acción</b>		
2	Concluye el requisito.	
<b>Pos-condiciones</b>		
2	No se registran las modificaciones realizadas.	
<b>Validaciones</b>		
2	Se validan los datos según lo establecido en el Modelo conceptual Administración de capital humano.	
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A

<b>Conceptos</b>	<b>Medida disciplinaria</b>	Ver en la interfaz: Código Denominación Clasificación
<b>Requisitos especiales</b>	N/A.	
<b>Asuntos pendientes</b>	N/A.	

Tabla 13: Especificación del requisito modificar medida

### Especificación del requisito eliminar medida

<b>Precondiciones</b>	Se ha registrado al menos una medida.	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
7	Se selecciona la medida a eliminar.	
8	El sistema verifica (ver validación 1) que se pueda eliminar la medida.	
9	Se solicita confirmación para eliminar la medida.	
10	Si el usuario confirma se elimina la medida.	
11	El sistema confirma la eliminación.	
12	Concluye el requisito.	
<b>Pos-condiciones</b>		
2	Se eliminó la medida.	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 3. No se puede eliminar la medida.</b>		
2	El sistema notifica por qué no puede eliminarse la medida.	
<b>Pos-condiciones</b>		
2	N/A	
<b>Flujo alternativo *.a El usuario cancela la acción</b>		
2	Concluye el requisito.	
<b>Pos-condiciones</b>		
2	No se elimina la medida.	
<b>Validaciones</b>		
2	Se validan que no se haya realizado ningún registro de medida con este tipo de incidencia.	
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A

<b>Conceptos</b>	<b>Medida disciplinaria</b>	Visible en la interfaz: Código Denominación Clasificación
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

Tabla 14: Especificación del requisito eliminar medida

### Especificación del requisito listar medida

<b>Precondiciones</b>		Se ha registrado al menos una medida.
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
3		El sistema muestra un listado de las medidas. Se muestra en un listado el código, denominación y la clasificación.
4		Concluye el requisito.
<b>Pos-condiciones</b>		
3	N/A	
4		
<b>Flujos alternativos</b>		
<b>Flujo alternativo</b>		
2	N/A	
<b>Pos-condiciones</b>		
2	N/A	
<b>Validaciones</b>		
2	N/A	
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A
<b>Conceptos</b>	<b>Medida disciplinaria</b>	Visible en la interfaz: Código Denominación Clasificación
<b>Requisitos especiales</b>	N/A.	
<b>Asuntos pendientes</b>	N/A	

Tabla 15: Especificación del requisito listar medida



## Especificación del requisito imprimir medida

<b>Precondiciones</b>	Se ha registrado al menos una medida en el sistema	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
4	El sistema permite seleccionar los elementos que se desean imprimir.	
5	El sistema permite seleccionar el formato en que se desea imprimir y luego se imprime.	
6	Concluye el requisito.	
<b>Pos-condiciones</b>		
2	N/A	
<b>Flujos alternativos</b>		
<b>Flujo alternativo *.a El usuario cancela la acción</b>		
2	Concluye el requisito.	
<b>Pos-condiciones</b>		
2	No se imprime la información.	
<b>Flujo alternativo 2.a El sistema permite realizar una vista previa</b>		
2	El sistema permite mostrar una vista previa de lo que se desea imprimir.	
<b>Pos-condiciones</b>		
2	Se muestra la información.	
<b>Flujo alternativo * No se pueden imprimir los datos por errores en el sistema, la red o la impresora.</b>		
2	El sistema notifica al usuario que los datos no pueden ser impresos.	
<b>Pos-condiciones</b>		
2	N/A.	
<b>Validaciones</b>		
2	N/A.	
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A
<b>Conceptos</b>	<b>Medida disciplinaria</b>	Visible en la interfaz: Código Denominación Clasificación
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

Tabla 16: Especificación del requisito imprimir medida

### 2.4.1.3 Requisito funcional gestionar registro de medidas disciplinarias

## Especificación del requisito adicional registro de medida disciplinaria

<b>Precondiciones</b>	Se ha registrado al menos una medida disciplinaria
<b>Flujo de eventos</b>	
<b>Flujo básico</b>	
11	Se introducen los datos del registro de la medida disciplinaria. Expediente Interno: Expediente interno del trabajador. Nombre: Nombre del trabajador. Primer Apellido: Primer apellido del trabajador. Segundo Apellido: Segundo apellido del trabajador. Nº de orden: Numero de orden en el registro anual. Violación cometida: Incidencia disciplinaria cometida por el trabajador. Medida disciplinaria: Medida disciplinaria que se aplicará. Fecha de notificación: Fecha de notificación al trabajador de la medida. Fecha de inicio: Fecha de inicio de la medida. Fecha fin: Fecha fin de la medida. Fecha rehabilitación: Fecha de rehabilitación de la medida. Nivel de la aplicación: Director general del Grupo Empresarial. Director general de la Entidad. Penalización: Importe económico de la medida.
12	El sistema valida (ver validación 1) los datos introducidos.
13	Si los datos son correctos el sistema los registra.
14	El sistema confirma el registro de los datos.
15	Concluye el requisito.
<b>Pos-condiciones</b>	
3	Se registró en el sistema un nuevo registro de medidas disciplinarias.
<b>Flujos alternativos</b>	
<b>Flujo alternativo 4.a Información errónea</b>	
7	El sistema señala los datos erróneos y permite corregirlos.
8	El usuario corrige los datos.
9	Volver al paso 2 del flujo básico.
<b>Pos-condiciones</b>	
3	N/A
<b>Flujo alternativo 4.b Información incompleta</b>	
7	El sistema señala los datos vacíos y permite corregirlos.
8	El usuario corrige los datos.
9	Volver al paso 2 del flujo básico.
<b>Pos-condiciones</b>	
3	N/A
<b>Flujo alternativo *.a El usuario cancela la acción</b>	
3	Concluye el requisito.
<b>Pos-condiciones</b>	
3	No se registran los datos.

<b>Validaciones</b>		
3	Se validan los datos según lo establecido en el Modelo conceptual administración de capital humano.	
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A.
<b>Conceptos</b>	<b>Medida disciplinaria</b>	Expediente Interno Nombre Primer Apellido Segundo Apellido Nº de orden Violación cometida Medida disciplinaria Fecha de notificación Fecha de inicio Fecha fin Fecha rehabilitación Nivel de la aplicación Penalización
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

Tabla 17: Especificación del requisito adicionar registro de medida disciplinaria

### Especificación del requisito modificar registro de medida disciplinaria

<b>Precondiciones</b>	Se ha registrado al menos una medida disciplinaria
<b>Flujo de eventos</b>	
<b>Flujo básico</b>	
15	Se selecciona la medida disciplinaria a modificar.
16	El sistema muestra y permite editar los datos de la medida disciplinaria.
17	Se introducen los datos del registro de la medida disciplinaria. Expediente Interno: Expediente interno del trabajador. Nombre: Nombre del trabajador. Primer Apellido: Primer apellido del trabajador. Segundo Apellido: Segundo apellido del trabajador. Nº de orden: Numero de orden en el registro anual. Violación cometida: Incidencia disciplinaria cometida por el trabajador. Medida disciplinaria: Medida disciplinaria que se aplicará. Fecha de notificación: Fecha de notificación al trabajador de la medida. Fecha de inicio: Fecha de inicio de la medida. Fecha fin: Fecha fin de la medida. Fecha rehabilitación: Fecha de rehabilitación de la medida.

	Nivel de la aplicación: Director general del Grupo Empresarial. Director general de la Entidad. Penalización: Importe económico de la medida.	
18	El sistema valida (ver validación 1) los datos introducidos.	
19	Si los datos son correctos el sistema los registra.	
20	El sistema confirma el registro de los datos.	
21	Concluye el requisito.	
	<b>Pos-condiciones</b>	
3	Se modificaron los datos de la medida disciplinaria.	
	<b>Flujos alternativos</b>	
	<b>Flujo alternativo 4.a Información errónea</b>	
7	El sistema señala los datos erróneos y permite corregirlos.	
8	El usuario corrige los datos.	
9	Volver al paso 4 del flujo básico.	
	<b>Pos-condiciones</b>	
3	N/A	
	<b>Flujo alternativo 4.b Información incompleta</b>	
7	El sistema señala los datos vacíos y permite corregirlos.	
8	El usuario corrige los datos.	
9	Volver al paso 4 del flujo básico.	
	<b>Pos-condiciones</b>	
3	N/A	
	<b>Flujo alternativo *.a El usuario cancela la acción</b>	
3	Concluye el requisito.	
	<b>Pos-condiciones</b>	
3	No se registran las modificaciones realizadas.	
	<b>Validaciones</b>	
3	Se validan los datos según lo establecido en el Modelo conceptual Administración de capital humano.	
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A
<b>Conceptos</b>	<b>Medida disciplinaria</b>	Expediente Interno Nombre Primer Apellido Segundo Apellido Nº de orden Violación cometida Medida disciplinaria Fecha de notificación Fecha de inicio Fecha fin Fecha rehabilitación Nivel de la aplicación

Penalización

<b>Requisitos especiales</b>	N/A.
<b>Asuntos pendientes</b>	N/A.

Tabla 18: Especificación del requisito modificar registro de medida disciplinaria

**Especificación del requisito eliminar registro de medida disciplinaria**

<b>Precondiciones</b>	Se ha registrado al menos un tipo de incidencia.	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
13	Se selecciona la medida disciplinaria a eliminar.	
14	El sistema verifica (ver validación 1) que se pueda eliminar la medida disciplinaria.	
15	Se solicita confirmación para eliminar la medida disciplinaria.	
16	Si el usuario confirma se elimina la medida disciplinaria.	
17	El sistema confirma la eliminación.	
18	Concluye el requisito.	
<b>Pos-condiciones</b>		
3	Se eliminó el tipo de incidencia.	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 3. No se puede eliminar el tipo de incidencia</b>		
3	El sistema notifica por qué no puede eliminarse la medida disciplinaria.	
<b>Pos-condiciones</b>		
3	N/A	
<b>Flujo alternativo *.a El usuario cancela la acción</b>		
3	Concluye el requisito.	
<b>Pos-condiciones</b>		
3	No se elimina la medida disciplinaria.	
<b>Validaciones</b>		
3	Se validan que no se haya realizado ningún registro de medida disciplinaria con este tipo de incidencia.	
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A

<b>Conceptos</b>	<b>Medida disciplinaria</b>	Expediente Interno Nombre Primer Apellido Segundo Apellido Nº de orden Violación cometida Medida disciplinaria Fecha de notificación Fecha de inicio Fecha fin Fecha rehabilitación Nivel de la aplicación Penalización
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

Tabla 19: Especificación del requisito eliminar registro de medida disciplinaria

#### Especificación del requisito listar registro de medida disciplinaria

<b>Precondiciones</b>	Se ha registrado al menos una medida disciplinaria en el sistema	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
5	El sistema muestra un listado de las medidas disciplinarias. Se muestra en un listado el expediente interno, el nombre, el primer apellido, segundo apellido, nº de orden, violación cometida, medida disciplinaria, fecha de notificación, fecha de inicio, fecha fin, fecha rehabilitación, nivel de la aplicación, ordenados por el expediente interno.	
6	Concluye el requisito.	
<b>Pos-condiciones</b>		
5	N/A	
<b>Flujos alternativos</b>		
<b>Flujo alternativo</b>		
3	N/A	
<b>Pos-condiciones</b>		
3	N/A	
<b>Validaciones</b>		
3	N/A	
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A

<b>Conceptos</b>	<b>Medida disciplinaria</b>	Expediente Interno Nombre Primer Apellido Segundo Apellido Nº de orden Violación cometida Medida disciplinaria Fecha de notificación Fecha de inicio Fecha fin Fecha rehabilitación Nivel de la aplicación Penalización
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

Tabla 20: Especificación del requisito listar registro de medida disciplinaria

#### Especificación del requisito filtrar registro de medida disciplinaria por período

<b>Precondiciones</b>	Se ha registrado al menos una medida disciplinaria en el sistema.	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
1	Se selecciona de la interfaz la opción de Filtrar.	
2	Se selecciona el período por el cual se quiere filtrar.	
3	Se muestra en la interfaz un listado de todos los pagos realizados en ese período.	
4	Concluye el requisito.	
<b>Pos-condiciones</b>		
1	Se mostró un listado del período.	
<b>Flujos alternativos</b>		
<b>Flujo alternativo</b>		
1	N/A	
<b>Pos-condiciones</b>		
1	N/A	
<b>Validaciones</b>		
1	N/	
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A

<b>Conceptos</b>	<b>Período de pago</b>	Visibles en la interfaz: Denominación Fecha Inicio Fecha Fin
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

Tabla 21: Especificación del requisito filtrar registro de medida disciplinaria por período

### Especificación del requisito registrar apelación

<b>Precondiciones</b>	Se ha registrado al menos una apelación en el sistema.
<b>Flujo de eventos</b>	
<b>Flujo básico</b>	
1	Se introducen los datos de la apelación. Expediente Interno: Expediente interno del trabajador. Nombre: Nombre del trabajador. Primer Apellido: Primer apellido del trabajador. Segundo Apellido: Segundo apellido del trabajador. Nº de orden: Numero de orden en el registro anual. Violación cometida: Incidencia disciplinaria cometida por el trabajador. Medida disciplinaria: Medida disciplinaria que se aplicará. Nivel de la aplicación: Director general del Grupo Empresarial. Resultado: Resultado de la apelación
2	El sistema valida (ver validación 1) los datos introducidos.
3	Si los datos son correctos el sistema los registra.
4	El sistema confirma el registro de los datos.
5	Concluye el requisito.
<b>Pos-condiciones</b>	
4	Se registró en el sistema una nueva apelación.

### Flujos alternativos

#### Flujo alternativo 4.a Información errónea

10	El sistema señala los datos erróneos y permite corregirlos.
11	El usuario corrige los datos.
12	Volver al paso 2 del flujo básico.

#### Pos-condiciones

4	N/A
---	-----

#### Flujo alternativo 4.b Información incompleta

10	El sistema señala los datos vacíos y permite corregirlos.
11	El usuario corrige los datos.
12	Volver al paso 2 del flujo básico.



<b>Pos-condiciones</b>		
4	N/A	
<b>Flujo alternativo *.a El usuario cancela la acción</b>		
4	Concluye el requisito.	
<b>Pos-condiciones</b>		
4	No se registran los datos.	
<b>Validaciones</b>		
4	Se validan los datos según lo establecido en el Modelo conceptual administración de capital humano.	
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A.
<b>Conceptos</b>	<b>Trabajador</b>	Expediente Interno Nombre Primer Apellido Segundo Apellido
	<b>Registrar apelación de la medida</b>	Nº de orden Violación cometida Medida disciplinaria Nivel de la aplicación Resultado
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

Tabla 22: Especificación del requisito registrar apelación

### Especificación del requisito imprimir registro de medida disciplinaria

<b>Precondiciones</b>	Se ha registrado al menos una medida disciplinaria en el sistema
<b>Flujo de eventos</b>	
<b>Flujo básico</b>	
7	El sistema permite seleccionar los elementos que se desean imprimir.
8	El sistema permite seleccionar el formato en que se desea imprimir y luego se imprime.
9	Concluye el requisito.
<b>Pos-condiciones</b>	
3	N/A
<b>Flujos alternativos</b>	
<b>Flujo alternativo *.a El usuario cancela la acción</b>	
3	Concluye el requisito.
<b>Pos-condiciones</b>	

3	No se imprime la información.	
<b>Flujo alternativo 2.a El sistema permite realizar una vista previa</b>		
3	El sistema permite mostrar una vista previa de lo que se desea imprimir.	
<b>Pos-condiciones</b>		
3	Se muestra la información.	
<b>Flujo alternativo * No se pueden imprimir los datos por errores en el sistema, la red o la impresora.</b>		
3	El sistema notifica al usuario que los datos no pueden ser impresos.	
<b>Pos-condiciones</b>		
3	N/A.	
<b>Validaciones</b>		
3	N/A.	
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A
<b>Conceptos</b>	<b>Medida disciplinaria</b>	Expediente Interno Nombre Primer Apellido Segundo Apellido Nº de orden Violación cometida Medida disciplinaria Fecha de notificación Fecha de inicio Fecha fin Fecha rehabilitación Nivel de la aplicación Penalización
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

Tabla 23: Especificación del requisito imprimir registro de medida disciplinaria

## 2.5 Conclusiones del capítulo

En el presente capítulo se realizó la modelación del negocio con el objetivo de identificar los procesos del negocio para dar paso a la captura y especificación de requisitos. Se realizó el modelo conceptual con el objetivo de identificar las entidades del negocio. Se identificaron y describieron los requisitos funcionales que satisfacen las necesidades reales de los usuarios y clientes, siendo el punto de partida para realizar el diseño de la propuesta de solución.

## CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

### 3.1 Introducción

En el presente capítulo se muestra una representación gráfica de los diagramas de clases del diseño, realizándose una breve descripción de los mismos y el diagrama de componentes. Se especifica además el modelo de datos físico resultante de la base de datos y se comienza la construcción de la propuesta de solución.

### 3.2 Diagrama de clases del diseño

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

- Clases asociadas.
- Métodos.
- Navegabilidad.
- Dependencias.

En la figura 4 se muestra el diagrama de clases del diseño gestionar violaciones, en la cual se le da solución a los requisitos funcionales adicionar, modificar, eliminar, imprimir y listar violaciones, se representan las principales clases, operaciones y relaciones que se necesitan para darle cumplimiento a estos requerimientos, donde las clases `gestionarviolaciones.js` y `gestionarviolaciones.phtml` conforman la capa arquitectónica de presentación. La clase `GestionarviolacionesController` es la encargada de la lógica del negocio, implementando funcionalidades que garantizan el cumplimiento de los requisitos identificados y las clases `NomClasificaciondisciplinaModel`, `NomViolacionesModel`, `NomClasificaciondisciplina`, `NomViolaciones`, `BaseNomClasificaciondisciplina` y `BaseNomViolaciones` son las encargadas del acceso a los datos de las tablas `nom_clasificaciondisciplina` y `nom_violaciones`.

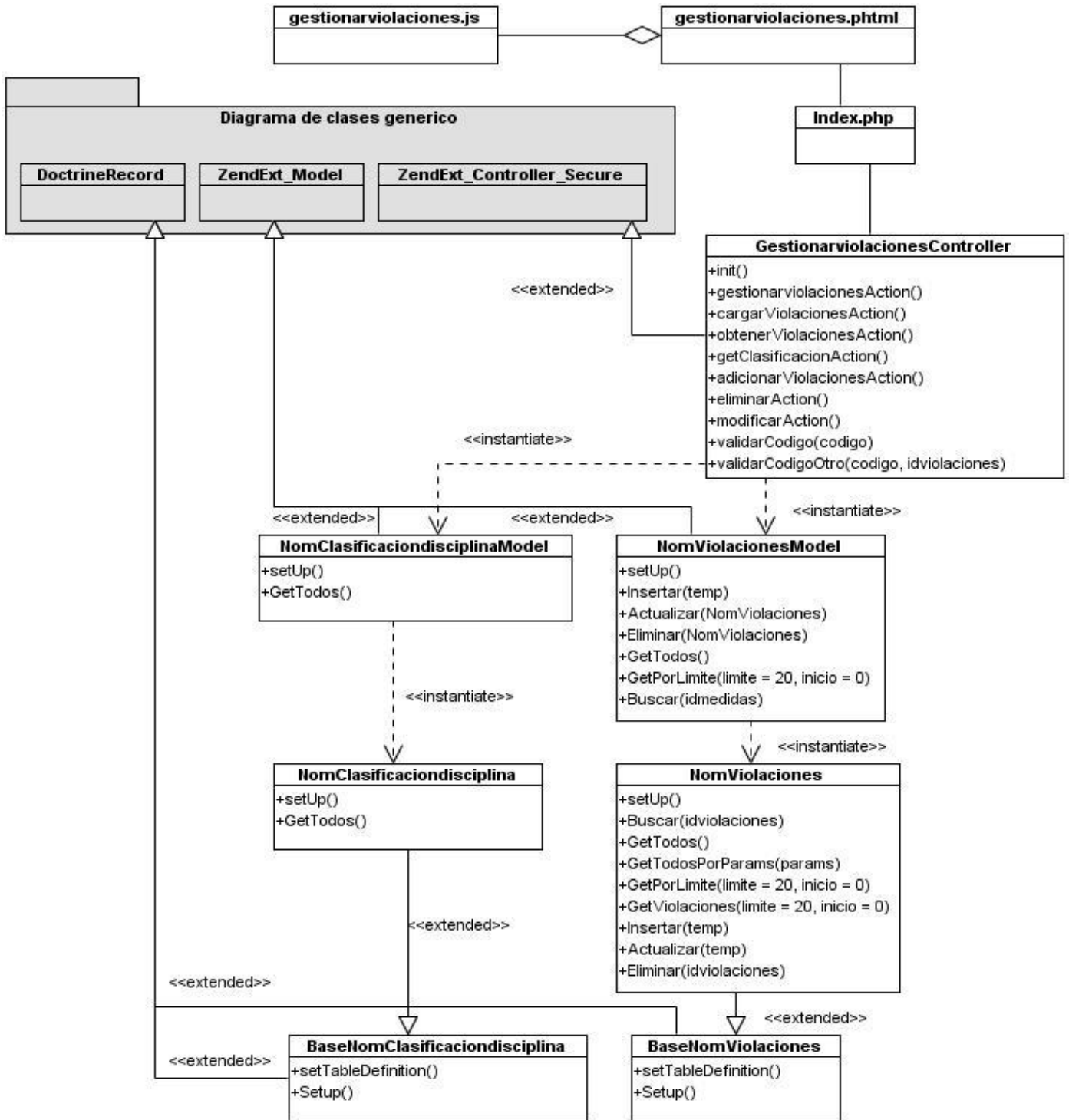


Figura 4: Diagrama de clase Gestionar violaciones

### 3.3 Diagrama de componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. En la figura 5 se muestra el diagrama de componentes que representa las relaciones de Disciplina Laboral con otros componentes del subsistema capital humano del Sistema Integral de Gestión Cedrux y los servicios que utiliza para obtener información de estos, como el que devuelve los datos de los periodo de pagos del componente periodo de pago, el que devuelve los datos de los trabajadores del componente trabajador y el servicio que le brinda a la nómina, para que esta tenga la información correspondiente de los registro de medidas.

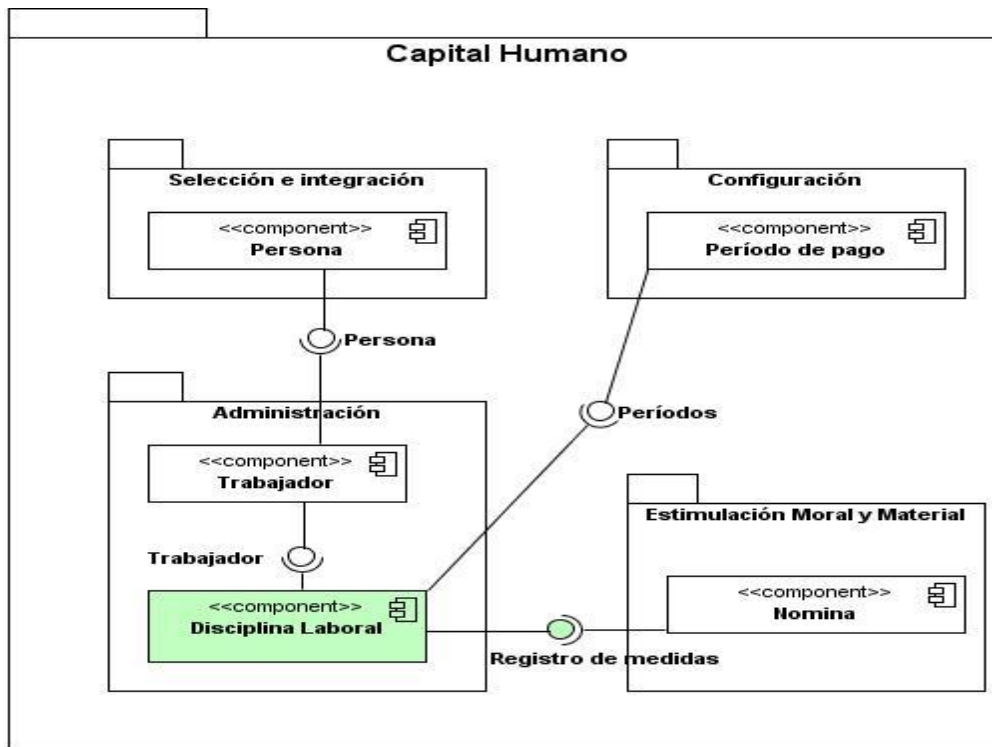


Figura 5: Diagrama de componente

### 3.4 Integración entre componentes

La integración entre los componentes está definida en una arquitectura de 3 capas: presentación (view), negocio (controller) y acceso a datos (models), consiste en el flujo de los datos desde la vista hacia la capa de datos y viceversa, a través de los diferentes elementos que la componen. Consta de 4 nodos de integración: vista-controlador, controlador-modelo, modelo-framework Doctrine y Doctrine-base de datos. La comunicación entre las capas dentro de un mismo componente se realiza mediante llamadas a métodos o eventos de forma directa.

Entre diferentes módulos y componentes, la integración se basa en el patrón Inversión de Control (IoC) y se realiza a través de un componente incluido en el framework Zend\_Ext, que permite operar sobre distintos esquemas en la base de datos realizando las transacciones adecuadas. En dicho componente se define el fichero ioc.xml que contiene la ubicación de cada uno de los componentes y los servicios que ofrecen las clases services correspondientes. De esta manera, la puerta de entrada de cada componente es el paquete de las clases de servicios que buscan en los modelos o entidades las funcionalidades requeridas.

Con la integración se persigue obtener una forma eficiente y flexible de combinar recursos internos o externos de los subsistemas usando:

- IoC Interno: para la integración entre componentes de un subsistema.
- IoC Externo: para la integración entre subsistemas.

### **3.5 Modelo de datos**

Un modelo de datos es aquel que describe la representación de los datos persistentes de la base de datos. Básicamente consiste en una descripción de algo conocido como contenedor de datos (donde se guarda la información), así como los métodos a almacenar y recuperar información de dichos contenedores. Consiste en:

- Objetos (entidades que existen y que se manipulan).
- Atributos (características básicas de estos objetos).
- Relaciones (forma en que se enlazan los distintos objetos entre sí).

En la figura 6 se muestra el modelo de datos físico resultante de las tablas de la base de datos con la que trabaja el componente disciplina laboral.

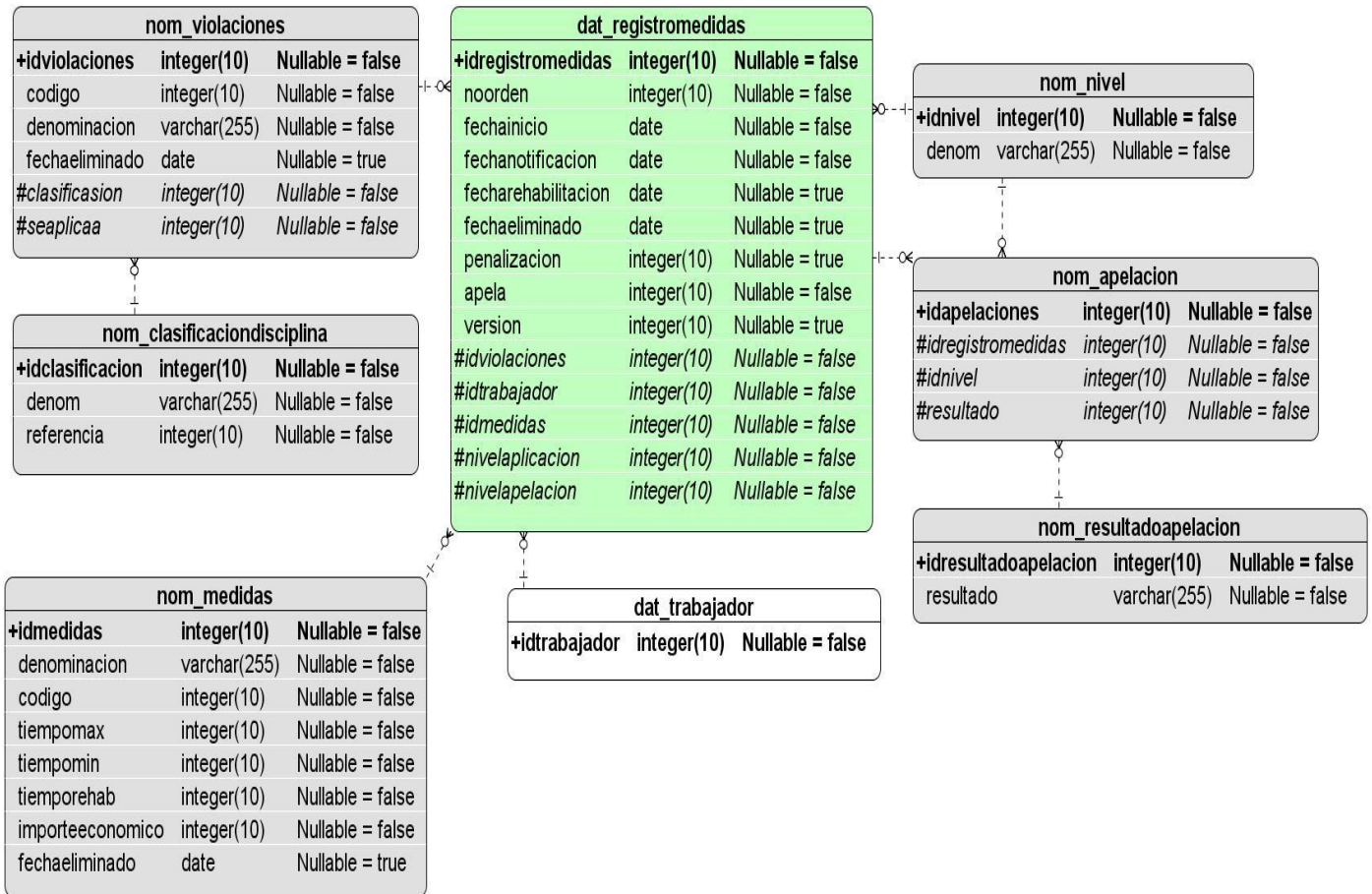


Figura 6: Modelo de datos

### 3.6 Conclusiones del capítulo

En el capítulo se realizó la modelación del diseño representándose gráficamente los diagramas de clases del diseño para describir cómo quedará la estructura del sistema. Se realizó el diagrama de componentes para representar las dependencias lógicas que tiene el componente disciplina laboral y el modelo de datos físico de la base de datos para percibir como persisten los datos en la base de datos. Se realizó la implementación del componente disciplina laboral para darle cumplimiento a los objetivos del trabajo. Al concluir este capítulo la propuesta de solución del trabajo está lista para ser validada, esto se realizará en el próximo capítulo.

## CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

### 4.1 Introducción

La dificultad para construir sistemas de software multiplica la probabilidad de que persistan errores aún después de haberse finalizado. Es imposible asegurar que un software se encuentre completamente libre de errores; sin embargo, existen formas y métodos para acercarse lo más posible a un resultado óptimo.

En este capítulo se realiza la validación de la solución propuesta. Se evalúa el diseño empleando métricas de software que proporcionan una medida de la complejidad y calidad del software. Se aplican pruebas con el objetivo de verificar la funcionalidad y estructura del componente desarrollado.

### 4.2 Métricas para validar el diseño

La realización de métricas son claves en la ingeniería del software orientada a objetos. Estas permiten tener una visión más clara y profunda y proporcionan un mecanismo para la evaluación de la calidad del software. Los objetivos principales de las métricas orientadas a objetos son:

- Comprender mejor la calidad del producto.
- Estimar la efectividad del proceso.
- Mejorar la calidad del trabajo realizado en el nivel del proyecto. (25)

Las métricas que se utilizaron para validar el diseño propuesto aplican los principales atributos de calidad de software, inspiradas en lo propuesto por Pressman (26), las métricas de tamaño de clase por Lorenz y Kidd (27) y la métrica acoplamiento de clase de la serie de métricas de Chidamber y Kemerer (28).

Las métricas orientadas a tamaños para una clase se centran en cálculos de atributos y de operaciones para una clase individual y promedian los valores para el sistema en su totalidad.

La métrica acoplamiento de clase sugiere que cuanto más acoplamiento se da en una clase, será más difícil de reutilizar. Además, las clases con excesivo acoplamiento dificultan la comprensibilidad y hacen más difícil el mantenimiento por lo que será necesario un mayor esfuerzo.

Atributos de calidad que se abarcan:



- **Responsabilidad:** Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto de la problemática propuesta.
- **Complejidad de implementación:** Consiste en el grado de dificultad que tiene implementar un diseño de clases determinado.
- **Reutilización:** Consiste en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.
- **Acoplamiento:** Consiste en el grado de dependencia o interconexión de una clase o estructura de clase con otras, está muy ligada a la característica de Reutilización.
- **Complejidad del mantenimiento:** Consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta pero fuertemente en los costes y la planificación del proyecto.
- **Cantidad de pruebas:** Consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad (Unidad) del producto (Componente, módulo, clase, conjunto de clases, etc.) diseñado.

#### 4.2.1 Métrica Tamaño Operacional de Clases (TOC)

Para medir el tamaño operacional de clases se tienen en cuenta los aspectos que se muestran en la tabla 23.

Tamaño operacional de clase (TOC)	
Descripción:	Está dado por el número de métodos asignados a una clase.
<b>Atributos que afecta:</b>	<b>Modo en que lo afecta:</b>
Responsabilidad	El aumento del TOC provoca un aumento de la responsabilidad asignada a la clase.
Complejidad de implementación	El aumento del TOC provoca un aumento de la complejidad de implementación de la clase.

Reutilización	Un aumento del TOC provoca una disminución en el grado de reutilización de la clase.
---------------	--

Tabla 23: Métrica Tamaño Operacional de Clase

En la tabla 24 se muestra las categorías por las que se va a evaluar la métrica TOC.

Atributo	Categoría	Criterio
Responsabilidad	Baja	$\leq$ Prom.(Promedio)
	Media	Entre Prom. y 2* Prom.
	Alta	$>$ 2* Prom.
Complejidad de implementación	Baja	$\leq$ Prom.
	Media	Entre Prom. y 2* Prom.
	Alta	$>$ 2* Prom.
Reutilización	Baja	$>$ 2* Prom.
	Media	Entre Prom. y 2* Prom.
	Alta	$\leq$ Prom.

Tabla 24: Rango de valores de para la evaluación técnica de los atributos de calidad relacionados con la métrica TOC.

Los resultados del instrumento de evaluación de la métrica Tamaño Operacional de clase (TOC) se muestran en las gráficas que se presentan en las figuras 7, 8, 9 y 10.

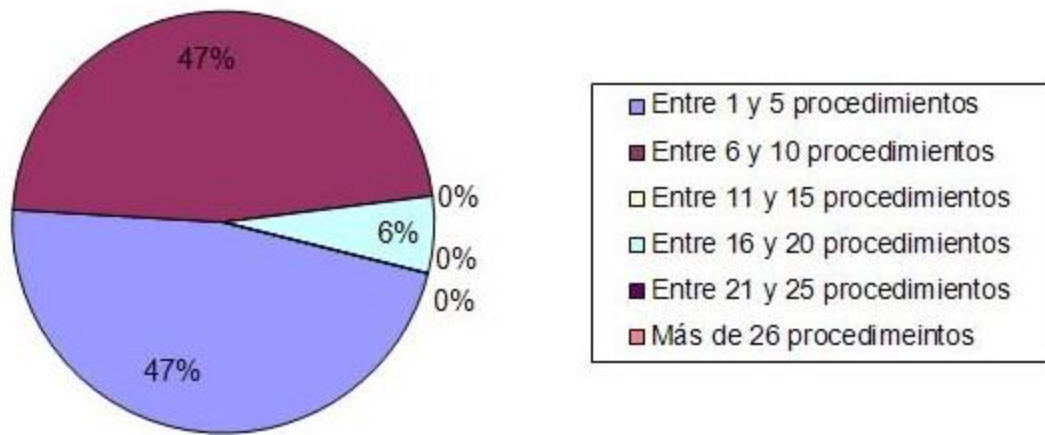


Figura 7: Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.

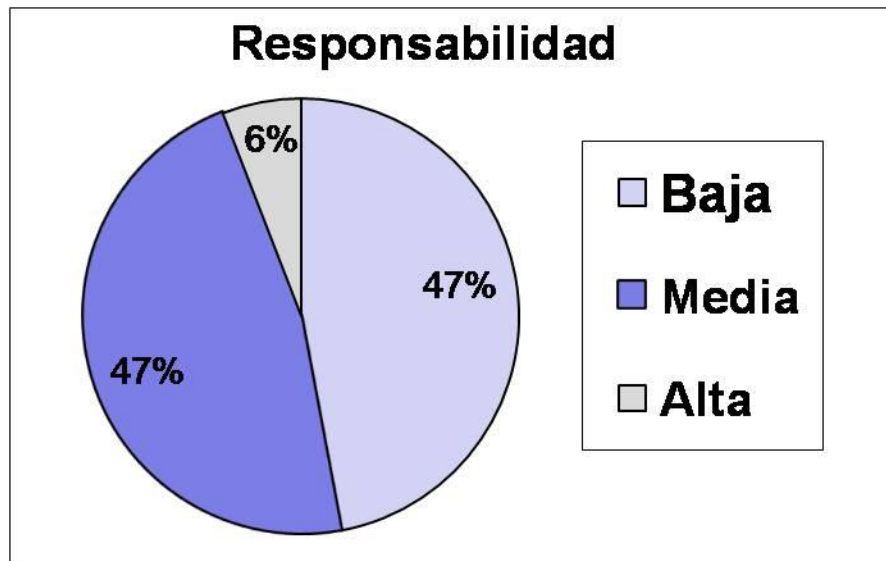


Figura 8: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.

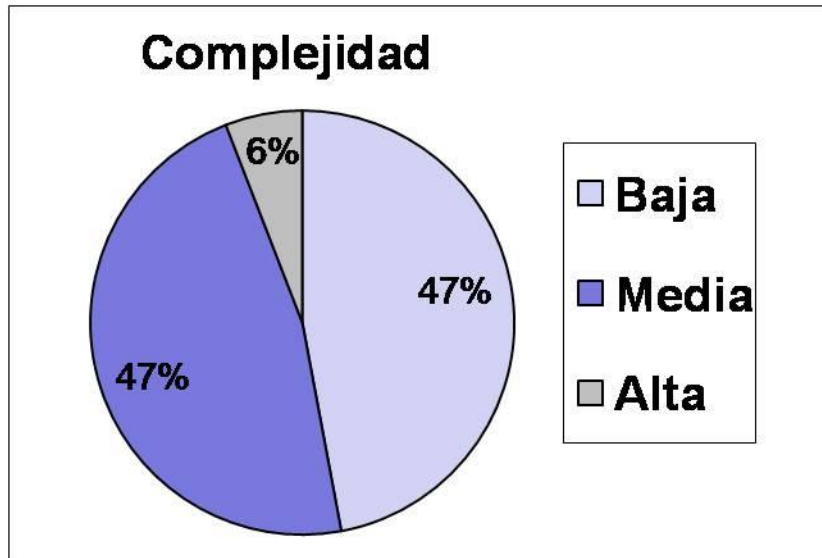


Figura 9: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de Implementación.

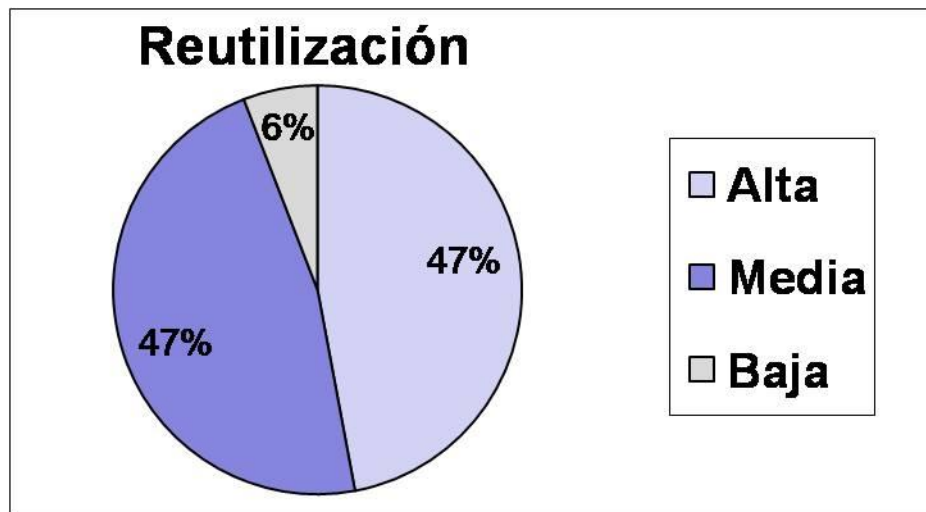


Figura 10: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.

Cuando existe un TOC alto se afectan los parámetros de calidad definidos por esta métrica. Se reduce la reutilización de las clases, la implementación se hace más compleja, las pruebas son difíciles de realizar y aumenta la responsabilidad de las clases.

La mayoría de las clases que conforman el sistema están dentro de las categorías de baja y media, para un 94 % en total, lo que demuestra la elevada reutilización, baja complejidad y responsabilidad en el diseño propuesto. Los resultados obtenidos son positivos según esta métrica.

#### 4.2.2 Métrica Relaciones entre Clases (RC)

Para realizar la métrica relaciones entre clases se tienen en cuenta los aspectos que se muestran en la tabla 25.

<b>Relaciones entre clases (RC)</b>	
Descripción:	Está dada por el número de relaciones de uso de una clase con otras.
<b>Atributos que afecta:</b>	<b>Modo en que lo afecta:</b>
Acoplamiento	El aumento del RC provoca un aumento del Acoplamiento de la clase.
Complejidad del mantenimiento	El aumento del RC provoca un aumento de la complejidad del mantenimiento de la clase.
Reutilización	El aumento del RC provoca una disminución en el grado de reutilización de la clase.
Cantidad de pruebas	El aumento del RC provoca un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.

**Tabla 25: Métrica Relaciones entre Clases (RC)**

En la tabla 26 se muestra las categorías por las que se va a evaluar la métrica TOC.

Atributo	Categoría	Criterio
Acoplamiento	Ninguna	0
	Baja	1
	Media	2
	Alta	>2
Complejidad y Mantenimiento	Baja	$\leq$ Prom.
	Media	Entre Prom. y $2^*$ Prom.
	Alta	$> 2^*$ Prom.
Cantidad de Pruebas	Baja	$\leq$ Prom.
	Media	Entre Prom. y $2^*$ Prom.
	Alta	$> 2^*$ Prom.
Reutilización	Baja	$> 2^*$ Prom.
	Media	Entre Prom. y $2^*$ Prom.
	Alta	$\leq$ Prom.

**Tabla 26: Rango de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica RC.**

Los resultados del instrumento de evaluación de la métrica Relaciones entre clases (RC) se muestran en las gráficas que se presentan en las figuras 11, 12, 13, 14 y 15.

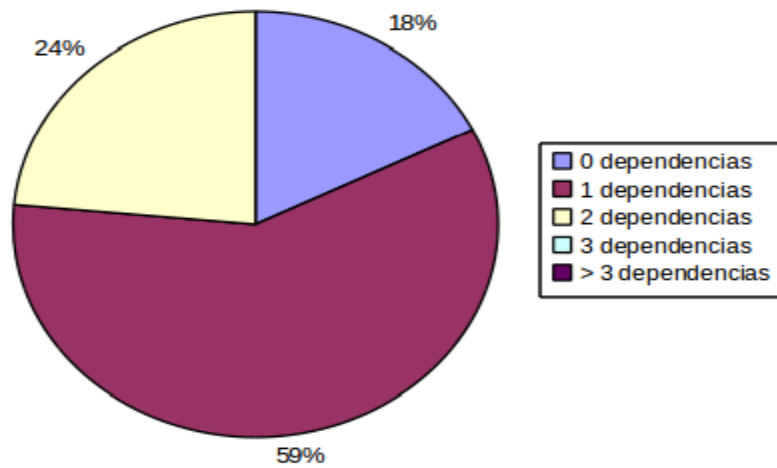


Figura 11: Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.

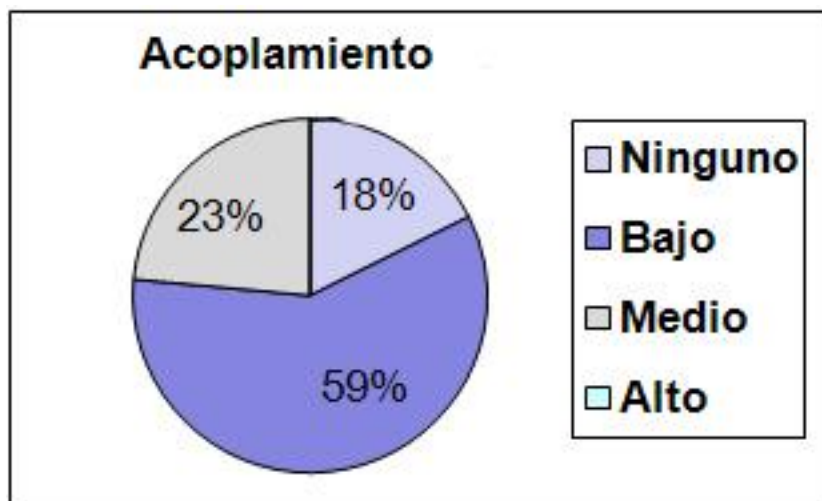


Figura 12: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento.



Figura 13: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de Mantenimiento.

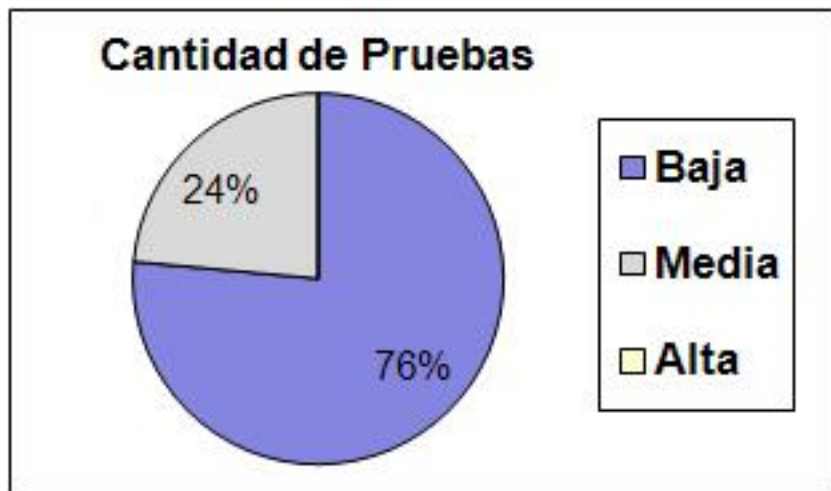
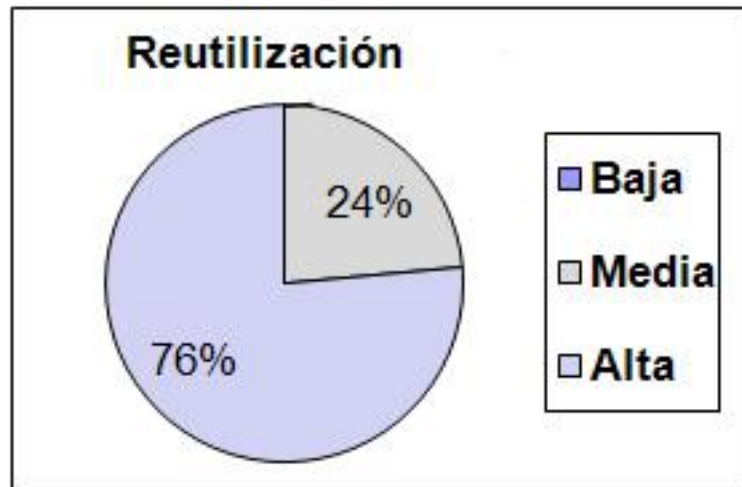


Figura 14: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de Pruebas.





**Figura 15: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización.**

Los resultados obtenidos durante la evaluación del instrumento de medición de la métrica RC demuestran que el diseño propuesto para el componente Disciplina Laboral se encuentra dentro de los niveles aceptables de calidad, mostrando que el 100% de las clases poseen menos de 3 dependencias entre clases. Los atributos de calidad fueron evaluados satisfactoriamente para el 100% de las clases, confirmando la elevada reutilización y bajo acoplamiento, complejidad y cantidad de pruebas en el diseño propuesto.

Las métricas de software aplicadas posibilitaron estimar la calidad de los atributos internos del producto, demostrando una aceptable calidad de diseño. La solución propuesta contribuirá a la disminución de disturbios durante la implementación del componente, garantizando la reutilización y agilidad en el proceso de desarrollo de software.

### **4.3 Modelo de pruebas**

Las pruebas del software son un elemento crítico para la garantía de la calidad del software. Las pruebas son realizadas con el objetivo de detectar errores en el sistema, por lo que se llevan a cabo durante todo el ciclo de vida del producto.

Para llevar a cabo el proceso de pruebas al componente se definen estrategias de pruebas con el propósito de garantizar la calidad del software.

### 4.3.1 Prueba de caja negra

Para verificar que la aplicación se comporta según los requerimientos establecidos por el cliente, se realizan método de caja negra utilizando casos de pruebas. En las tablas 27, 28 y 29 se especifica el caso de prueba para el requisito eliminar medida.

#### Condiciones de ejecución

- Se debe identificar y autenticar ante el sistema y además debe tener los permisos para ejecutar esta acción.
- Se debe seleccionar el subsistema Capital humano/Administración de capital humano/Disciplina laboral/Gestionar violaciones.

#### Requisito a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1: Eliminar medida.	El sistema debe permitir eliminar medida.	EP 1.1: Eliminar medida.	<ul style="list-style-type: none"> <li>– Se muestra un mensaje confirmando si realmente se desea eliminar la medida.</li> <li>– Se presiona el botón <b>Aceptar</b>.</li> <li>– Se muestra un mensaje de información.</li> <li>– Se presiona el botón <b>Aceptar</b>.</li> </ul>
		EP 1.2: Cancelar la operación.	<ul style="list-style-type: none"> <li>– Se muestra un mensaje confirmando si realmente se desea eliminar la medida.</li> </ul>

– Se presiona el botón **Cancelar**.

**Tabla 27: Descripción del caso de prueba para el requisito Eliminar medida**

**Descripción de variables**

No	Nombre de campo	Tipo	Válido	Inválido	Inválido	Inválido	Inválido
1	Código	Cadena.	Números.	Letras, Caracteres especiales.	NA	NA	NA
2	Denominación	Cadena.	Letras y números.	Carácteres especiales.	NA	NA	NA
3	Mínimo	Campo numérico.	Números.		NA	NA	NA
4	Máximo	Campo numérico.	Números.		NA	NA	NA
5	Tiempo de rehabilitación	Campo de Selección (No	Números.		NA	NA	NA
6	Importe económico	Cuadro de selección	Verdadero o falso		NA	NA	NA

**Tabla 28: Descripción de variables del caso de prueba para el requisito Eliminar medida**

**Juego de datos a probar**

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
------------------	-----------	-----------------------	------------------------

EP 1.1 Eliminar medida.	El sistema muestra el mensaje: “¿Desea eliminar la medida?”. El sistema muestra el mensaje: “Se ha eliminado la medida satisfactoriamente.”.	NA
EP 1.2 Cancelar la operación.	El sistema muestra el mensaje: “¿Desea eliminar la medida?”. El sistema cierra el mensaje de confirmación sin realizar ninguna operación.	NA

**Tabla 29: Datos de prueba del caso de prueba para el requisito Eliminar medida**

Las pruebas de caja negra a la solución fueron desarrolladas por el Grupo de Aseguramiento de la Calidad del Centro de Informatización para la Gestión de Entidades (CEIGE). Estas se realizaron en tres iteraciones de prueba, donde finalmente se comprobó en la tercera iteración que el componente estaba libre de no conformidades, para así lograr su liberación. La tabla 30 muestra los resultados de cada una de las iteraciones realizadas.

Agrupación de requisitos	No Conformidades		
	Iteración 1	Iteración 2	Iteración 3
Gestionar violaciones	2	1	0
Gestionar medidas	3	1	0
Gestionar registro de medidas	2	1	0

TOTAL	7	3	0
-------	---	---	---

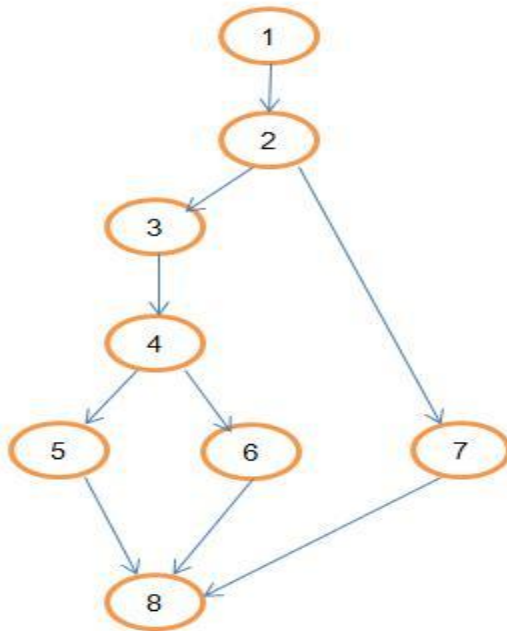
**Tabla 30: Resultado de las iteraciones de las pruebas de caja negra realizadas por el grupo de aseguramiento de la calidad.**

### 4.3.2 Prueba de caja blanca

Para realizar la prueba de caja blanca, específicamente la prueba del camino básico, es necesario calcular antes la complejidad ciclomática del algoritmo o fragmento de código a analizar. En la figura 16 se enumeran las sentencias de código del procedimiento realizado sobre el método buscarTrabPorExpedienteAction(). Este método es seleccionado por ser considerado crítico ya que se encarga de buscar los datos del trabajador que se necesita para registrar los registro de medidas, por lo que un mal funcionamiento de este, traería consigo que no se puedan realizar correctamente las funcionalidades que garantizan la agrupación de requisitos funcionales gestionar registro de medidas.

```
function buscarTrabPorExpedienteAction(){
    $exp = $this->_request->getPost('expediente'); 1
    if($exp){ 2
        $params->numeroexpediente = $exp; 3
        $trab = $this->plIntegrator->trabajador->BuscarTrabajador($params); 3
    if($trab){ 4
        $param->idpersona=$trab[0]->idpersona; 5
        $pers = $this->plIntegrator->persona->BuscarPersonaPorParam($param); 5
        $result = array('Nombre' =>$pers[0]->nombre, 'Apellido1' => $pers[0]->papel,
        'Apellido2' => $pers[0]->sapel, 'idtrabajador'=>$trab[0]->idtrabajador); 5
        echo json_encode($result); 5
    }
    else
        echo json_encode(array()); 6
    }
    else
        echo json_encode(array()); 7
    } 8
}
```

**Figura 16: Método buscarTrabPorExpedienteAction()**



**Figura 17: Grafo de flujo asociado al algoritmo buscarTrabPorExpedienteAction()**

### **Cálculo de la complejidad ciclomática**

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

Para efectuar el cálculo de la complejidad ciclomática del código es necesario tener varios parámetros como son la cantidad total de aristas del grafo y la cantidad total de nodos, los cuales se obtienen del grafo de flujo que se muestra en la figura 17 y se utilizan en la siguiente fórmula:

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

$$V(G) = (9 - 8) + 2$$

$$V(G) = 3$$

Siendo: A la cantidad total de aristas y N la cantidad total de nodos. Se puede usar también:

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

$$V(G) = 2 + 1 \quad V(G) = 3$$

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

$$V(G) = R$$

$$V(G) = 3$$

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

El cálculo efectuado mediante las tres fórmulas ha dado el mismo valor, por lo que se puede plantear que la complejidad ciclomática del código es de 3, lo que significa que existen tres posibles caminos por donde el flujo puede circular, este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento tratado. Seguidamente es necesario representar los caminos básicos por los que puede recorrer el flujo, los cuales se muestran en la tabla 31. En las tablas 32, 33 y 34 se muestran los casos de pruebas asociados al algoritmo `buscarTrabPorExpedienteAction()` por cada camino básico.

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

**Tabla 31: Caminos básicos del flujo**

<b>Caso de prueba para el camino básico # 1 (1 – 2 – 3 – 4 – 5 – 8)</b>	
Descripción	Se busca los datos de un trabajador por un número de expediente determinado.
Condición de ejecución	Debe existir al menos un trabajador registrado en el sistema. Se debe introducir el campo de búsqueda requerido.
Entrada	<code>numeroexpediente = \$exp;</code>

Resultados esperados	Se deben obtener los datos del trabajador solicitado.
Resultados	Se obtienen los datos del trabajador.
Salida	<code>\$result = array('Nombre', 'Apellido1', 'Apellido2', 'idtrabajador');</code>

**Tabla 32: Caso de prueba para el algoritmo buscarTrabPorExpedienteAction(): camino básico #1**

<b>Caso de prueba para el camino básico # 2 (1 – 2 – 3 – 4 – 6 – 8)</b>	
Descripción	Se busca los datos del trabajador por un número de expediente determinado.
Condición de ejecución	Debe existir al menos un trabajador registrado en el sistema. Se debe introducir el campo de búsqueda requerido.
Entrada	<code>numeroexpediente = \$exp;</code>
Resultados esperados	Se deben obtener los datos del trabajador solicitado.
Resultados	No se obtienen los datos del trabajador porque no hay ningún trabajador con el número de expediente asignado.
Salida	<code>\$result =null;</code>

**Tabla 33: Caso de prueba para el algoritmo buscarTrabPorExpedienteAction(): camino básico #2**

<b>Caso de prueba para el camino básico # 3 (1 – 2 – 7 – 8)</b>	
Descripción	Se busca los datos del trabajador por un número de expediente determinado.
Condición de ejecución	Debe existir al menos un trabajador registrado en el sistema. Se debe introducir el campo de búsqueda requerido.



Entrada	numeroexpediente = null;
Resultados esperados	Se deben obtener los datos del trabajador solicitado.
Resultados	No se obtienen los datos del trabajador porque no se introdujo ningún número de expediente.
Salida	\$result =null;

**Tabla 34: Caso de prueba para el algoritmo buscarTrabPorExpedienteAction(): camino básico #3**

Con la realización de las pruebas de caja blanca a diferentes métodos significativos de la aplicación, fue posible apreciar como estos respondían satisfactoriamente a los resultados que se esperaban por cada caso de prueba. Por lo que se puede observar que los códigos implementados cumplen con lo que se necesita para lograr una buena implementación de los diferentes requisitos que se definieron en el negocio.

#### **4.4 Conclusiones del capítulo**

Durante el desarrollo de este capítulo se ejecutaron métricas de diseño para caracterizar numéricamente los distintos aspectos del desarrollo y se realizaron pruebas que permitieron evaluar todos los elementos del software. Fueron validadas las funcionalidades implementadas a través de diferentes pruebas de caja negra, mostrando cómo respondían adecuadamente a los requisitos funcionales y garantizando la satisfacción plena de las necesidades reales de los usuarios y demandas del cliente. De igual forma, se aplicaron pruebas de caja blanca para efectuar las revisiones al código, las cuales dieron resultados positivos.

## CONCLUSIONES GENERALES

Al culminar la investigación se le dio cumplimiento a los objetivos planteados, se logró la implementación de un componente capaz de cubrir los procesos de la gestión de la disciplina laboral de forma correcta que facilita el cálculo de la nómina por parte del subsistema capital humano del Sistema de Gestión Integral Cedrux. Lo anterior se ve demostrado a través de lo siguiente:

- Se analizaron los fundamentos teóricos y las principales aplicaciones vinculadas al campo de acción, tanto a nivel nacional como internacional, demostrando la necesidad del nuevo sistema.
- Se realizó un estudio acerca de las tecnologías y herramientas utilizadas en el marco de trabajo del subsistema capital humano, reafirmando que eran las idóneas para ser utilizadas.
- Se logró obtener el modelo de negocio, el análisis, el diseño y la implementación de la solución, validada con resultados positivos, que disminuye las dificultades para generar la nómina.

## RECOMENDACIONES

Después de realizadas las conclusiones de este trabajo se recomienda:

- Continuar el estudio del tema con el objetivo de incluir nuevas funcionalidades en versiones posteriores del sistema.
- Realizar el despliegue del componente propuesto como parte del subsistema de Capital Humano del Sistema Integral de Gestión Cedrux.

## REFERENCIAS BIBLIOGRÁFICAS

1. gestiopolis. [En línea] [Citado el: 5 de febrero de 2011.] <http://www.gestiopolis.com/recursos/documentos/fulldocs/rrhh1/admonrrhhlari.htm>.
2. gestiopolis. [En línea] [Citado el: 5 de febrero de 2011.] <http://www.gestiopolis.com/canales/derrhh/articulos/13/nocastigues.htm>.
3. mtss. [En línea] [Citado el: 5 de febrero de 2011.] [http://www.mtss.cu/decretos\\_leyes/DECRETO-LEY%20No.%20176%2015-8-97%20%20SISTEMA%20DE%20JUSTICIA%20LABORAL.doc](http://www.mtss.cu/decretos_leyes/DECRETO-LEY%20No.%20176%2015-8-97%20%20SISTEMA%20DE%20JUSTICIA%20LABORAL.doc).
4. ecured. [En línea] [Citado el: 5 de febrero de 2011.] [http://www.ecured.cu/index.php/Versat\\_Sarasola](http://www.ecured.cu/index.php/Versat_Sarasola).
5. Sap. [En línea] [Citado el: 5 de febrero de 2011.] <http://www.sap.com/spain/solutions/business-suite/erp/hcm.epx>.
6. Sageabra. [En línea] [Citado el: 5 de febrero de 2011.] [www.sageabra.com/hrms.aspx](http://www.sageabra.com/hrms.aspx).
7. AssetsNS. [En línea] [Citado el: 5 de febrero de 2011.] <http://assets.co.cu/assets.asp>.
8. [En línea] [Citado el: 6 de febrero de 2011.] <http://www.tecnologicocomfenalco.edu.co/iacademica/sistemas/Agora/articulos/UML.pdf>.
9. [En línea] [Citado el: 6 de febrero de 2011.] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
10. [En línea] [Citado el: 6 de febrero de 2011.] <http://www.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>.
11. visual-paradigm. [En línea] [Citado el: 6 de febrero de 2011.] <http://www.visual-paradigm.com/product/vpuml/>.
12. [En línea] [Citado el: 7 de febrero de 2011.] <http://es.kioskea.net/contents/langages/langages.php3>.
13. [En línea] [Citado el: 7 de febrero de 2011.] <http://bibdigital.epn.edu.ec/bitstream/15000/2110/1/CD-2887.pdf>.
14. [En línea] [Citado el: 8 de febrero de 2011.] <http://blogs.antartec.com/desarrolloweb/2008/10/>.
15. [En línea] [Citado el: 8 de febrero de 2011.] <http://es.wikibooks.org/wiki/>

Programaci%C3%B3n\_en\_PHP.

16. doctrine-project. [En línea] [Citado el: 8 de febrero de 2011.] <http://www.doctrine-project.org/>.

17. [En línea] [Citado el: 8 de febrero de 2011.] [http://www.error500.net/garbagecollector/archives/categorias/bases\\_de\\_datos/sistema\\_gestor\\_de\\_base\\_de\\_datos\\_sgbd.php](http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php).

18. postgresql.org. [En línea] [Citado el: 8 de febrero de 2011.] <http://www.postgresql.org/docs/current/static/intro-what-is.html>.

19. sqlmanager. [En línea] [Citado el: 8 de febrero de 2011.] <http://www.sqlmanager.net/products/postgresql/manager/>.

20. [En línea] [Citado el: 10 de febrero de 2011.] [http://www.wikilearning.com/tutorial/control\\_de\\_versiones\\_con\\_subversion-el\\_control\\_de\\_versiones/19266-1](http://www.wikilearning.com/tutorial/control_de_versiones_con_subversion-el_control_de_versiones/19266-1).

21. [En línea] [Citado el: 10 de febrero de 2011.] <https://forja.rediris.es/docman/view.php/123/117/TortoiseSVN-1.4.1-es.pdf>.

22. [En línea] [Citado el: 25 de marzo de 2011.] <http://www.grancomo.com/2006/05/02/patrones-para-el-diseno-de-interfaz/>.

23. [En línea] [Citado el: 25 de marzo de 2011.] <http://arquitectura-de-software.blogspot.com/2006/05/orm-object-relational-mapping-ii-parte.html>.

24. EVA. [En línea] [Citado el: 25 de marzo de 2011.] [http://eva.uci.cu/file.php/102/Curso\\_2010-2011/Clases/Semana\\_10/Conferencia\\_10/Materiales\\_complementarios/Introduccion\\_a\\_la\\_Disciplina\\_Analisis\\_y\\_Diseno.pdf](http://eva.uci.cu/file.php/102/Curso_2010-2011/Clases/Semana_10/Conferencia_10/Materiales_complementarios/Introduccion_a_la_Disciplina_Analisis_y_Diseno.pdf).

25. [En línea] [Citado el: 23 de Mayo de 2011.] [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/gonzalez\\_d\\_h/capitulo6.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_d_h/capitulo6.pdf).

26. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque práctico.* 1998.

27. **Lorenz M., Kidd J.** *Object Oriented Software Metrics.* Prentice Hall. 1994.

28. **Kemerer. Shyam R., Chidamber. Chris F.** *A metric suite for object oriented design.* 1994.