

Universidad de las Ciencias Informáticas



Trabajo de Diploma

*Para optar por el título de Ingeniero en Ciencias
Informáticas*

TÍTULO

*Análisis y Diseño del componente de Incidencias
del Sistema Integral de Gestión CEDRUX.*

Autores:

Olga Yarisbel Rojas Grass.

Irela Sosa Vila.


Tutor: Ing. Dayana Méndez Alayo.

Co-tutor: Ing. Javier Heredia Ruiz.

Consultante: Ing. Crescencio Campos Hung.

Asesor: Ing. José Ramón Fernández Pérez.

Ciudad de la Habana, junio 2009



"El mundo camina hacia la era electrónica... Todo indica que esta ciencia se constituirá en algo así como una medida del desarrollo; quien la domine será un país de vanguardia. Vamos a volcar nuestros esfuerzos en este sentido con audacia revolucionaria"

Ernesto Che Guevara.



DATOS DE CONTACTO

Síntesis del Tutor Ing. Dayana Mendez Alayo

Graduado de Ingeniero en Ciencias Informáticas en el año 2007 en la Universidad de las Ciencias Informáticas. Especialista del Bloque 8 de la IP, Centro UCID.

Años de graduado: 2

E-mail: daymendez@uci.cu

Síntesis del Co-Tutor Ing. Javier Heredia Ruíz

Graduado de Ingeniero en Ciencias Informáticas en el año 2007 en la Universidad de las Ciencias Informáticas. Con categoría de Adiestrado y un año de experiencia laboral.

Años de graduado: 2

E-mail: jheredia@uci.cu

Síntesis del Consultante Ing. Crescencio Campos Hung

Graduado de Ingeniería Eléctrica en el año 1989, en la ex URRS, entre los años 1991 y 1995 se desempeñó en el cargo de Especialista en Redes y Sistemas en la Unidad presupuestada "Las Camariocas". De 1996 hasta el 2001 ocupó el cargo de Técnico en Inversiones. De 2001 al 2005 pasó a trabajar en la Dirección de Recursos Laborales de la Oficina Central del Ministerio de la Industria Básica. Desde el 2006 ocupa el Cargo de Especialista Principal del Grupo de Proyección Estratégica del Grupo Empresarial Cubaniquel y se desempeña actualmente como Especialista Funcional del Ministerio de Trabajo y Seguridad Social.

Años de graduado: 19.

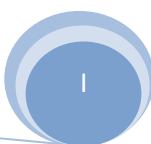
E-mail: ccampos@cubaniquel.moa.minbas.cu

Síntesis del Asesor Ing. José Ramón Fernández Pérez

Graduado de Ingeniero en Ciencias Informáticas en el año 2007, en la Universidad de las Ciencias Informáticas. Profesor del Dpto. de Sistemas Digitales de la Facultad 10, Líder de Tareas productivas en el proyecto Fábrica de Portales.

Años de graduado: 2

E-mail: jfernandez@uci.cu





AGRADECIMIENTOS

Le agradezco a todos los que me han apoyado para lograr el sueño de verme graduada.

En especial a mis padres Olga y Omar por estar siempre ahí para mí, brindándome su amor y dándoselo a mi pequeño como me lo dieron a mí, mientras me formo profesionalmente.

A mi querido tatico por ayudarme a encontrar el camino del estudio nuevamente y por sus regaños para que estudiara.

A mi pequeño por estar lejos de mí y portarse como un hombrecito.

A mi familia en general que todos se han preocupado por mí.

A mis compañeros de clases, a mis amistades, y en especial a Yani y Yami por compartir alegres y tristes momentos conmigo.

A mi tutora Dayana, mi co-tutor Javier y a Chencho, por su apoyo y dedicación.

A mi compañera de tesis Irela.

A los profes que ayudaron en mi formación profesional.

Al proyecto ERP-Cuba por este último año en el que he aprendido muchas cosas y en el que he hecho nuevas amistades.

A la Revolución.

A Fidel, nuestro siempre Comandante en Jefe.

A todos ustedes muchas gracias.

Yari

Muchas han sido las personas involucradas en el logro de este sueño.

Quiero agradecer en primer lugar a mis padres por quererme con tanta intensidad y darme todo lo que tuvieron a su alcance, a quienes no solo agradezco sus consejos y el apoyo constante, sino además, el privilegio de ser su hija.

A mis familiares.

A mis amigos por ayudarme en todo este tiempo y compartir conmigo alegrías y tristezas.

A todos los profesores que han formado en mí los valores necesarios para ser una buena profesional.

A Dayana por ser una tutora ejemplar, ayudarnos tanto y estar ahí cuando la necesitamos.

A mi co-tutor Javier y a Chéncho.

A Olga, mi compañera de tesis, por tanto sacrificio y dedicación.

A Yoelkys que, aunque esta lejos de mí, no ha dejado de preocuparse y de alentarme para llegar adonde he llegado, me ha ayudado mucho y se lo agradeceré eternamente.

Por último quiero agradecer especialmente a la Revolución por darme la oportunidad de estudiar en la UCI que ha resultado tan positiva para mi formación personal y profesional.

A todos los que de una forma u otra han contribuido a la realización de este trabajo de diploma, de corazón, muchas gracias.

Irela



DEDICATORIA

A mis padres que lo han dado todo para verme graduada, a mi hombrecito de 5 añitos José Alejandro por aguantar estar lejos de mí, a mi esposo Jose (tatico) al que quiero tanto, a mis hermanos Omarito y Danielito, a mi suegra Esther, a la memoria de mi abuelo Rigo y a mis consentidores Vicente y Lili. Son lo más grande que tengo, a ustedes les dedico uno de los momentos más importantes de mi vida.

Yari

A mis padres que siempre confiaron en mí y me brindaron su apoyo para que viera realizados mis sueños, quiero regalarles este momento y honrarlos por tanto amor y dedicación. Los quiero mucho.

Irela

Las entidades empresariales y presupuestadas se han insertado en el mundo de la informatización, debido a la necesidad de gestionar un gran volumen de recursos que necesitan ser controlados. En el presente trabajo se estudian los procesos de negocio de las entidades empresariales y presupuestadas relacionados con la gestión de incidencias de los trabajadores, que a pesar de la existencia de varios sistemas informáticos nacionales y extranjeros, se detecta que no todos se adecuan a las particularidades propias de las entidades cubanas, y en su mayoría la gestión de las incidencias está enfocada solo al pago de las nóminas.

El objetivo del trabajo es proponer como solución el análisis y diseño de un componente que gestione las incidencias de los trabajadores, brinde en tiempo real la información necesaria para procesar la nómina, sea desarrollado sobre tecnologías libres, sea adaptable a todas las entidades cubanas y se ajuste a las exigencias establecidas en la legislación laboral vigente.

PALABRAS CLAVE

Análisis, diseño, componente, gestión, procesos, incidencia.

| | |
|--|-----------|
| INTRODUCCIÓN | 1 |
| Capítulo1: Fundamentación teórica de la investigación. | 5 |
| 1.1 Introducción. | 5 |
| 1.2 Gestión de Incidencias. Principales conceptos..... | 5 |
| 1.3 Sistemas informáticos relacionados con la gestión de incidencias. | 6 |
| 1.3.1 Sistemas informáticos en el mundo. | 7 |
| 1.3.2 Sistemas informáticos en Cuba. | 8 |
| 1.4 Tendencias y desarrollo del software libre..... | 11 |
| 1.5 Estudio de Lenguajes y Tecnologías..... | 11 |
| 1.5.1 Lenguaje de programación y tecnologías del lado del cliente..... | 11 |
| 1.5.2 Lenguaje de programación del lado del servidor. | 13 |
| 1.5.3 Sistema gestor de base de datos. | 13 |
| 1.6 Metodología y herramienta para la modelación del software. | 14 |
| 1.6.1 Metodología. | 14 |
| 1.6.2 Lenguajes de Modelado. | 15 |
| 1.6.3 Herramienta CASE..... | 16 |
| 1.7 Fundamentación de las herramientas, lenguajes y tecnologías..... | 16 |
| 1.8 Conclusiones. | 18 |
| Capítulo 2. Descripción del sistema. | 19 |
| 2.1 Introducción. | 19 |
| 2.2 Reglas del Negocio. | 19 |
| 2.3 Mapa de procesos y descripción del negocio. | 19 |
| 2.4 Técnicas utilizadas en las actividades de la ingeniería de requisitos..... | 20 |
| 2.5 Representación del Modelo Conceptual..... | 22 |
| 2.5.1 Conceptos del modelo conceptual y sus descripciones. | 23 |
| 2.6 Patrón utilizado para los requisitos..... | 23 |
| 2.7 Especificación de requisitos. | 24 |
| 2.7.1 Requisitos funcionales del sistema..... | 24 |
| 2.7.2 Especificaciones de los requisitos. | 25 |
| 2.7.3 Reportes generados de la gestión de incidencias..... | 41 |
| 2.8 Prototipos de interfaz de usuarios propuestos..... | 41 |
| 2.9 Conclusiones. | 42 |

| | |
|---|-----------|
| Capítulo 3. Diseño del sistema | 43 |
| 3.1 Introducción. | 43 |
| 3.2 Patrones utilizados..... | 43 |
| 3.2.1 Patrones de asignación de responsabilidades..... | 43 |
| 3.2.2 Patrón GOF..... | 44 |
| 3.2.3 Patrón de arquitectura Modelo Vista Controlador..... | 44 |
| 3.3 Diagramas de clases del diseño..... | 45 |
| 3.3.1 Descripciones de las clases de diseño..... | 50 |
| 3.4 Diagrama de componentes. | 61 |
| 3.5 Conclusiones. | 63 |
| Capítulo 4. Validación de los requisitos y del diseño. | 64 |
| 4.1 Introducción. | 64 |
| 4.2 Pasos para la validación de requisitos. | 64 |
| 4.3 Técnicas de validación de los requisitos. | 64 |
| 4.4 Métricas para la validación de los requisitos. | 65 |
| 4.5 Métricas para la validación del diseño..... | 67 |
| 4.6 Conclusiones. | 73 |
| CONCLUSIONES | 74 |
| RECOMENDACIONES | 75 |
| BIBLIOGRAFÍA | 76 |
| GLOSARIO DE TÉRMINOS | 78 |
| ANEXOS | 79 |

| | | |
|----------|--|----|
| Tabla 1 | Conceptos y descripciones. | 23 |
| Tabla 2 | Especificación del requisito Adicionar Tipo de Incidencia..... | 25 |
| Tabla 3 | Especificación del requisito Modificar Tipo de Incidencia. | 27 |
| Tabla 4 | Especificación del requisito Eliminar Tipo de Incidencia. | 28 |
| Tabla 5 | Especificación del requisito Adicionar Tipo de Impuesto. | 29 |
| Tabla 6 | Especificación del requisito Modificar Tipo de Impuesto..... | 31 |
| Tabla 7 | Especificación del requisito Eliminar Tipo de Impuesto. | 32 |
| Tabla 8 | Especificación del requisito Adicionar asociación incidencia-impuesto..... | 33 |
| Tabla 9 | Especificación del requisito Eliminar asociación incidencia-impuesto..... | 34 |
| Tabla 10 | Especificación del requisito Adicionar Registro de Incidencia. | 35 |
| Tabla 11 | Especificación del requisito Modificar Registro de Incidencia..... | 37 |
| Tabla 12 | Especificación del requisito Eliminar Registro de Incidencia..... | 40 |
| Tabla 13 | Descripción de la clase GestIncidenciaController..... | 50 |
| Tabla 14 | Descripción de la clase NomIncidenciumModel..... | 51 |
| Tabla 15 | Descripción de la clase Nomincidencium..... | 51 |
| Tabla 16 | Descripción de la clase NomTipoincidencium..... | 52 |
| Tabla 17 | Descripción de la clase GestimpuestoController..... | 53 |
| Tabla 18 | Descripción de la clase NomimpuestoModel..... | 54 |
| Tabla 19 | Descripción de la clase Nomimpuesto..... | 54 |
| Tabla 20 | Descripción de la clase DatRegistroimpuestoModel..... | 55 |
| Tabla 21 | Descripción de la clase DatRegistroimpuesto..... | 57 |
| Tabla 22 | Descripción de la clase GestIncidenciaTrabController..... | 57 |
| Tabla 23 | Descripción de la clase DatRegistroincidenciumModel..... | 59 |
| Tabla 24 | Descripción de la clase DatRegistroincidencium..... | 59 |
| Tabla 25 | Descripción de la clase AsociarimpincController..... | 60 |
| Tabla 26 | Descripción de otras clases..... | 61 |
| Tabla 27 | Cantidad de procedimientos por clases..... | 67 |
| Tabla 28 | Criterio para calcular la Responsabilidad..... | 68 |
| Tabla 29 | Criterio para calcular la Complejidad de implementación..... | 68 |
| Tabla 30 | Criterio para calcular la Reutilización..... | 68 |
| Tabla 31 | Responsabilidad y Complejidad de las clases..... | 69 |
| Tabla 32 | Reutilización de las clases..... | 69 |
| Tabla 33 | Relaciones de uso por clases..... | 70 |
| Tabla 34 | Acoplamiento de las clases..... | 71 |
| Tabla 35 | Reutilización de las clases..... | 71 |



ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 1. Mapa de Procesos | 20 |
| Figura 2. Modelo Conceptual. | 22 |
| Figura 3. Diagrama de clases del diseño de Tipo de Incidencia..... | 46 |
| Figura 4. Diagrama de clases del diseño de Tipo de Impuesto..... | 47 |
| Figura 5. Diagrama de clases del diseño de Registrar Incidencias. | 48 |
| Figura 6. Diagrama de clases del diseño de Asociación incidencia-impuesto..... | 49 |
| Figura 7. Diagrama de Componentes..... | 62 |
| Figura 8 Representación de la responsabilidad y complejidad. | 69 |
| Figura 9 Representación de la reutilización. | 69 |
| Figura 10 Representación del acoplamiento. | 71 |
| Figura 11 Representación de la reutilización..... | 71 |

INTRODUCCIÓN

La economía cubana a partir de la desintegración del campo socialista de Europa del Este y el arreciamiento del bloqueo por parte del gobierno estadounidense, se vio imposibilitada de mantenerse al tanto de los adelantos tecnológicos que en el mundo han ido surgiendo. El país en aras de revertir esta situación ha destinado grandes cantidades de recursos de acuerdo a las posibilidades económicas para el desarrollo informático.

De este modo, durante los primeros años de este siglo, el país se ha propuesto informatizar la sociedad, lo que trajo consigo la creación de nuevos sistemas a la altura de las nuevas tecnologías. Las entidades empresariales y presupuestadas no están ajenas a este proceso, debido a la existencia de muchos recursos que deben ser controlados de forma eficaz, para un mejor funcionamiento y desarrollo de las mismas. Uno de ellos son los Recursos Humanos, que de una forma u otra contribuyen a que el desarrollo y la eficiencia de las entidades sean aún mayores. Para tener un mejor control de cada uno de los recursos humanos es necesario mantener por ejemplo, un registro de asistencia, donde se pueda manejar el tiempo de producción de cada trabajador y poder definir mediante los diferentes conceptos de pago, el salario que este devengará en el período de pago correspondiente, entre otros.

La importancia que poseen los departamentos de los Recursos Humanos en cualquier entidad empresarial y presupuestada en la compleja realidad de las organizaciones modernas, es indiscutible. Los mismos han incrementado su papel en la empresa, adquiriendo nuevas responsabilidades y desarrollando varias tareas desde la selección e integración de los trabajadores, hasta organización del trabajo, administración de capital humano, capacitación y desarrollo, evaluación y desempeño, seguridad y salud, remuneración moral y material, entre otras.

En este sentido, es indispensable contar con un software de gestión que dé soporte al volumen de procesos que tienen lugar en estos departamentos para la correcta adquisición, mantenimiento, desarrollo, gestión del activo humano y los resultados derivados de su trabajo.

El país no cuenta con un sistema que gestione todos los procesos de los Recursos Humanos en una entidad empresarial y presupuestada, dichos procesos son seguidos de forma manual, la información puede ser duplicada o perdida por el deterioro de los documentos, haciéndose lo suficientemente engorroso como para que existan errores que afectan su funcionamiento. Se dificulta la organización y recuperación en tiempo de la información necesaria que es utilizada por otros procesos que sí están automatizados, como el procesamiento de las nóminas. Es indispensable tener un sistema que recupere la información con mayor rapidez, fiabilidad y seguridad para gestionar las incidencias de los trabajadores, entendiéndose por incidencia a todo lo que le suma o le resta dinero al salario que estos devengan. Un buen control de las mismas le permite conocer al trabajador y al área de Contabilidad, la solicitud y aprobación de las vacaciones, el importe de los descuentos, así como los pagos a efectuar por conceptos de licencias o subsidios, sirviendo de base para la confección de las nóminas.

Problema a resolver:

¿Cómo contribuir a mejorar el control de las incidencias de los trabajadores en las entidades empresariales y presupuestadas?

Por tanto el **objeto de estudio** corresponde a los procesos de gestión de la Administración de los Recursos Humanos.

Definiendo el **campo de acción** como los procesos de gestión de incidencias de los trabajadores.

Se plantea como **objetivo general** realizar el análisis y diseño de un sistema para la gestión de las incidencias de los trabajadores.

Objetivos específicos:

- ✓ Realizar la fundamentación teórica de la investigación.
- ✓ Definir los procesos de negocio y describir los requisitos funcionales del sistema.
- ✓ Diseñar la propuesta de un sistema que gestione las incidencias.
- ✓ Validar los requisitos y el diseño de clases.

Para darle cumplimiento a los objetivos planteados se han trazado las siguientes **tareas de la investigación**:

- ✓ Realizar el diseño teórico de la investigación.
- ✓ Investigar y evaluar los sistemas informáticos empleados en los procesos relacionados con las incidencias.
- ✓ Estudiar y fundamentar la metodología, herramienta y lenguajes de modelado a utilizar.
- ✓ Entrevistar a los funcionales para conocer y entender el funcionamiento actual de los Recursos Humanos.
- ✓ Definir y especificar los requisitos de software.
- ✓ Realizar el diseño del sistema.
- ✓ Aplicar métricas para la validación de los requisitos y el diseño.

Como **idea a defender** se plantea que si se realiza el análisis y diseño de un sistema informático para la gestión de las incidencias de los trabajadores, se mejorarían los procesos para gestionar las incidencias y el procesamiento de las nóminas.

El trabajo está estructurado en cuatro capítulos.

Capítulo 1. Fundamentación teórica de la investigación: Este capítulo constituye la fundamentación teórica del presente trabajo. Su lectura le ofrecerá información referente a los principales conceptos tratados. Lo pondrá al tanto del estado del arte del tema y por último podrá conocer la herramienta, la metodología y los lenguajes de modelado a utilizar para dar solución al problema en cuestión.

Capítulo 2. Descripción del sistema: Este capítulo ofrece el mapa de procesos de negocio, el modelo conceptual, la especificación de los requerimientos de software que están presente para el modelado del sistema y la descripción de los reportes que pueden ser generados a partir del sistema.

Capítulo 3. Diseño del sistema: Este capítulo inicia con una breve conceptualización del diseño y de los patrones a utilizar, mostrando luego la realización de los diagramas de clases con sus descripciones y el diagrama de componentes.

Capítulo 4. Validación de los requisitos y del diseño: Este capítulo ofrece las técnicas y métricas utilizadas para la validación de los requisitos y del diseño así como los resultados de las métricas aplicadas.

Capítulo1: Fundamentación teórica de la investigación.

1.1 Introducción.

En el presente capítulo se abordan los principales conceptos a tratar relacionados con la gestión de las incidencias, se estudia la existencia de sistemas o modelos informáticos empleados en los procesos relacionados con la misma. Se estudian las tecnologías, los lenguajes de programación, el sistema gestor de base de datos, las herramientas, metodología y lenguajes con los cuales se realizará la propuesta del sistema.

1.2 Gestión de Incidencias. Principales conceptos.

Incidencia:

Las incidencias son la caracterización del pago en términos de su naturaleza y del tratamiento de los parámetros de tiempo, importe, acumulación de vacaciones, aplicación de impuestos y procedimientos de contabilización. Son incidencias todo aquello que implica variación en la utilización del fondo de tiempo laboral y los importes a devengar por los trabajadores (ausencias, impuntualidades, vacaciones, licencias, subsidios, penalizaciones de trabajo, trabajo extraordinario, etc.).

Son ejemplos de incidencias las vacaciones, las licencias de maternidad, los días de conmemoración nacional, los días feriados no laborados, las horas extras, las ausencias injustificadas, llegadas tardes, etc.

Control de Asistencia:

El control de asistencia representa un registro primario obtenido por métodos mecánicos o manuales donde se anotan diariamente los datos relacionados con la asistencia de los trabajadores, así como las incidencias relativas al cumplimiento de la disciplina laboral. Este modelo sirve de fuente de información de la Pre-nómina (modelo SC-4-05). (1)

Elaboración de la Pre-nómina:

En la pre-nómina (modelo SC-4-05) se relacionan para cada trabajador, a partir del control de asistencia, todas las incidencias (tiempo correspondiente a ausencias, impuntualidades, vacaciones, licencias, subsidios, penalizaciones de trabajo, trabajo extraordinario, etc.) que implican variaciones en la utilización del fondo de tiempo laboral y los importes a devengar por los trabajadores. (2)

Esta información es transferida al área de contabilidad para el procesamiento de las diferentes nóminas.

Análisis sobre la Utilización del Fondo de Tiempo Laboral:

El histórico de las incidencias reflejadas en la pre-nómina permite realizar diferentes estudios sobre el aprovechamiento del fondo de tiempo, así como el cálculo de los principales indicadores de ausentismo y otros relacionados con la utilización del fondo de tiempo laboral. (3)

Modelo de Control de Ausencia y sus Causales:

Anualmente debe elaborarse un modelo para el control de las ausencias y sus causales, el cual se archiva en el expediente laboral del trabajador, este modelo se obtiene del histórico de ausencias.

Impuestos:

Consisten en el por ciento sobre el salario que debe aportar la entidad al presupuesto del estado (Contribución a la seguridad social y el Impuesto sobre la fuerza de trabajo). (4)

Capital Humano:

Conjunto de conocimientos, experiencias, habilidades, sentimientos, actitudes, motivaciones, valores y capacidad para hacer, portados por los trabajadores para crear más riquezas con eficiencia. Es además, conciencia, ética, solidaridad, espíritu de sacrificio y heroísmo.

1.3 Sistemas informáticos relacionados con la gestión de incidencias.

Con el avance de las tecnologías de la información y las comunicaciones, tanto las empresas internacionales como las entidades empresariales y presupuestadas han tratado de nutrirse con su uso, ya que se hace casi imposible que estas puedan por sí mismas controlar de forma manual las actividades contables.

1.3.1 Sistemas informáticos en el mundo.

Sistema de gestión Aqua Business Entities.

Aqua Business Entities, un ERP español que está creado alrededor de dos sólidos pilares tecnológicos de Microsoft como .Net y SQL Server 2005, controla y gestiona los recursos necesarios para alcanzar el máximo rendimiento y productividad en el entorno de trabajo. Ejemplo de los módulos que contiene están:

- ✓ Aqua Sales, Marketing & Services – Control de los procesos comerciales.
- ✓ Aqua CRM – Gestión de la relación con clientes.
- ✓ Aqua Financials – Gestión financiera.
- ✓ Aqua SCM – Control de la cadena de suministro.
- ✓ Aqua Manufacturing – Producción.
- ✓ Aqua Human Resources – Recursos Humanos.
- ✓ Aqua Quality – Gestión y control de calidad.

La solución Human Resources lleva a cabo la gestión de los recursos humanos con los que cuenta una empresa y así mantiene organizada la información con los movimientos, incidencias y modificaciones que se produzcan. Los datos que ofrece Human Resources suponen una gran ayuda para los responsables de la gestión del capital humano de la empresa, ya que facilitan la planificación y mejoran la eficacia de la empresa desde el punto de vista de las personas.

Aqua Human Resources permite la clasificación de las ausencias de los trabajadores por tipos: vacaciones, bajas laborales, formación, visitas médicas, etc. Si se trata de una ausencia de varios días, como es el caso de las vacaciones, se puede introducir un rango de fechas que permitirá controlar el tiempo que ese trabajador se ausentará de su puesto de trabajo. Del mismo modo, ofrece la posibilidad de insertar un número de horas si la ausencia no fuera de una jornada completa. (5)

ASSETS NS, Sistema de Gestión Integral.

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 y está siendo utilizado en las Universidades. 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. Se facilita el uso de la parametrización para adaptarse a las exigencias de cada entidad en particular, garantizando que sus reportes tengan la forma y el contenido que el usuario les defina.

El módulo Recursos Humanos de ASSETS NS está concebido para calcular las nóminas y controlar los recursos laborales de una entidad. Ambas actividades comparten una base de datos única, por lo que se reduce considerablemente la posibilidad de ocurrencia de errores. (6)

El sistema puede calcular y contabilizar nóminas de salario para cualquier tipo de pago (sueldo fijo, jornal, pago por rendimiento), incluyendo el pago de vinculaciones colectivas, horas extras, condiciones laborales anormales e interrupciones. De igual forma, calcula y contabiliza las nóminas de vacaciones y subsidios y nominillas de diferentes tipos (salario, vacaciones, subsidios, reintegros, estimulación). Todos los procesos automáticos se realizan siguiendo los criterios y restricciones establecidos por la legislación laboral vigente en el país. Está habilitado para considerar diferentes fondos de tiempo mensuales y diarios que pueden ser definidos por el usuario de acuerdo a las características de la entidad, así como manejar estructuras salariales móviles: pago de idoneidad fija, móvil y retribuciones complementarias, pagos por resultados finales y medidas salariales. (7)

1.3.2 Sistemas informáticos en Cuba.**El VERSAT-Sarasola: Sistema cubano de Gestión Contable-Financiero.**

Sus creadores definen al VERSAT-Sarasola como “un paquete integrado para la gestión económica financiera que permite enviar información eficaz, de forma inmediata, desde lugares apartados, a la vez que ofrece mayor organización, control y disciplina en cada gestión”. Está hecho en Delphi y como base de datos usa SQL.

Se distingue por ser el primer sistema de contabilidad cubano certificado, según las nuevas normativas establecidas por los Ministerios de Finanzas y Precios y de la Informática y las Comunicaciones, para este tipo de Software. (8) (9)

Contiene 10 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, Nómina de Salarios.

Nómina de Salarios. El subsistema es lo suficientemente adecuado 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), como sueldista, pago libre, vacaciones. Permite la configuración de Impuestos, bonificaciones, condiciones anormales, retenciones, descuentos, penalizaciones, etc.

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. (8)

Al igual que otros sistemas en explotación, en VERSAT el tratamiento de la incidencia nada más está dirigido al pago de las nóminas y se obvia su utilidad para realizar estudios sobre el aprovechamiento del fondo de tiempo laboral, no cuenta con un módulo de recursos humanos.

SISCONT5 sistema de Gestión Contable del (MINBAS).

SISCONT5 es un sistema certificado que se aviene a las definiciones y conceptos del Ministerio de la Industria Básica (MINBAS), es desarrollado y comercializado por TECNOMATICA y sus acciones contables financieras le han permitido ser utilizado en otras

entidades nacionales fuera del entorno del MINBAS. Entre sus módulos sobresalen los de Efectivo en Caja y Bancos, Inventarios, Cobros y Pagos, Facturación, Activos Fijos Tangibles, Nómina y Contabilidad. (10)

Puede ser explotado en régimen monousuario y multiusuario. Gestiona los conceptos de pagos permitiendo adicionar, modificar y eliminar dichos conceptos, cuenta con un nomenclador que define el tipo de trabajador, pero todo lo referido al trabajador y a los pagos a efectuarle se realizan en el módulo de Nómina, no cuenta con un módulo de Recursos Humanos.

Este sistema a pesar de encontrarse en explotación en mas de 200 entidades del país y los especialistas de TECNOMATICA encontrarse trabajando en una nueva versión de su módulo de nóminas, no tienen previsto por el momento desarrollar un módulo de recursos humanos. El tema relacionado con la gestión de las incidencias es muy limitado y solo está enfocado al pago de las nóminas por lo que no permite realizar estudios y análisis sobre la utilización del fondo de tiempo laboral.

El estudio de este pequeño grupo de sistemas informáticos, ya sea de factura nacional o extranjera, que se encuentran en explotación en el país, permite concluir que los mismos fueron desarrollados para un entorno muy específico, el tratamiento de la gestión de las incidencias por lo general está enfocado solo al pago de las nóminas y no garantizan la posibilidad de realizar estudios y análisis sobre la utilización del fondo de tiempo laboral. Se tratará de lograr que exista una relación directa entre los trabajadores y las incidencias que le son reportadas y que le ofrezca a nómina la información necesaria en tiempo real, para efectuar el pago, el sistema que se pretende diseñar tomará como idea la forma en que son gestionados los conceptos de pagos y la asociación de los impuestos a los mismos, del sistema Versat Sarasola.

La necesidad de encontrar y explotar las reservas de productividad en el sistema empresarial cubano unido a las exigencias establecidas en la legislación laboral vigente exige la utilización de sistemas informáticos que garanticen:

- ✓ Independencia tecnológica.
- ✓ Flexibilidad ante el entorno laboral cambiante.

1.4 Tendencias y desarrollo del software libre.

En Cuba se ha desencadenado un auge nacional del software libre, principalmente en los órganos legislativos del estado y gobierno para garantizar la seguridad y la integridad de la información confidencial. El país se encuentra enfrascado en realizar un crecimiento en la industria del software, que da sus primeros pasos y cuya única alternativa sostenible para su avance, es la migración al software libre dándole un impulso tremendo a esta corriente de desarrollo. (11)

Existen ejemplos como el de la Universidad de las Ciencias Informáticas (UCI), que acertadamente ha dedicado un por ciento de su matrícula al desarrollo de las tecnologías libres, y hoy es pilar en este campo, con un trabajo ascendente. Ya cuenta con una distribución de GNU/Linux orientada a Escritorio llamada Nova que responde a las necesidades de las instituciones cubanas. (12)

Hoy el desarrollo de software con la utilización de las tecnologías libres hará que Cuba sea respetada y obtenga como valores fundamentales el logro de un estado de soberanía e independencia tecnológica.

1.5 Estudio de Lenguajes y Tecnologías.

1.5.1 Lenguaje de programación y tecnologías del lado del cliente.

JavaScript: Es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Con javascript se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador. Entre las acciones típicas que se pueden realizar en javascript están: por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo, por el otro, javascript permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se pueden crear páginas interactivas con programas como calculadoras, agendas, o tablas de

cálculo. Javascript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. (13)

AJAX (Asynchronous JavaScript and XML): Esta tecnología, o para ser más preciso, este conjunto de tecnologías aplicadas de forma concreta, permite crear "aplicaciones web" más eficientes en la interacción con el usuario. AJAX aúna todas estas tecnologías:

- ✓ Estructura y presentación de la información basada en estándares mediante XHTML y CSS.
- ✓ DOM para interactuar dinámicamente con los datos.
- ✓ Intercambio y manipulación de datos usando XML and XSLT.
- ✓ Recuperación de datos asincrónica usando XMLHttpRequest y JavaScript. (14)

Ajax puede utilizarse hoy en día como una opción más rápida y eficiente de navegación web, no como el único método de navegación y es bueno utilizarse en webs que intentan ser "aplicaciones".

EXT JS es un framework que utiliza el lenguaje javascript para simular aplicaciones de escritorio.

Se trata de una excelente librería para crear interfaces de usuario, EXT Js facilita mucho el trabajo a los desarrolladores, pues en unas cuantas líneas de javascript pueden hacer interfaces realmente buenas. Ofrece una gran cantidad de componentes para crear interfaces de usuarios complejas y permite construir aplicaciones de escritorio en la web.

Ventajas:

- ✓ Código reutilizable.
- ✓ Independiente o adaptable a frameworks diferentes.
- ✓ Orientada a la programación de interfaces tipos desktop en la web.
- ✓ El API es homogenizado independientemente del adaptador usado. Los controles siempre se verán igual.
- ✓ Soporte Comercial.
- ✓ Una extensa comunidad de Usuarios. (15)

Navegadores compatibles.

Ext JS soporta y es compatible con la mayoría de navegadores actuales.

- ✓ Internet Explorer.
- ✓ Firefox.
- ✓ Safari.
- ✓ Opera. (16)

1.5.2 Lenguaje de programación del lado del servidor.

PHP (Hypertext Processor): PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas.

Características.

- ✓ Es un lenguaje multiplataforma.
- ✓ Soporte para varios servidores Web.
- ✓ Sintaxis clara y bien definida.
- ✓ Bastante sencillo de aprender y utilizar.
- ✓ Permite las técnicas de Programación Orientada a Objetos.
- ✓ Tiene manejo de excepciones (desde PHP5).
- ✓ Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad: MySQL, PostgreSQL, Oracle, MSSQL Server, entre otros.
- ✓ Ofrece una solución simple y universal para las paginaciones dinámicas de la web de fácil programación.
- ✓ Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- ✓ Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- ✓ El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP. (17)

1.5.3 Sistema gestor de base de datos.

PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente sólo

se podían ver en productos comerciales de alto calibre. Es usado para manejar grandes cantidades de información y está basado en el modelo relacional, aunque incorpora conceptos del modelado orientado a objeto. Se pueden definir consultas anidadas, vistas, crear funciones por el usuario, no sólo en el lenguaje natural SQL, sino en varios más, entre ellos C, lenguaje nativo PostgreSQL, Perl, PHP y Java. Es multiplataforma, soporta múltiples transacciones, integridad de datos, presenta una estabilidad muy alta, gran seguridad de los datos, soporta la réplica y procedimientos almacenados. Propone un tamaño ilimitado para las base de datos, lo que da la medida de un gestor de base de datos robusto, y con grandes funcionalidades. (18)

1.6 Metodología y herramienta para la modelación del software.

1.6.1 Metodología.

En los primeros momentos de la historia del software no existían guías, procedimientos, ni metodologías para el desarrollo del mismo, ya que las demandas de los clientes eran muy pobres y el software no jugaba un papel importante en la sociedad. A medida que fue evolucionando el desarrollo de la informática y la exigencia de los clientes fue cada vez mayor, los especialistas se dieron la tarea de desarrollar diferentes metodologías para organizar el trabajo, basándose en métodos comunes para el desarrollo. Estas metodologías fueron diseñadas bajo un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para lograr la calidad, que es el principal objetivo estratégico en las organizaciones.

Modelo de Desarrollo orientado a componentes del proyecto ERP –Cuba.

El Modelo de Desarrollo orientado a componentes del proyecto ERP-Cuba está orientado a las necesidades y artefactos generados por dicho proyecto, el cual se creó como un híbrido de diversas metodologías de desarrollo, que tiene como características fundamentales:

- ✓ Se modela el negocio por procesos, no por casos de usos.
- ✓ La ingeniería de requisitos es más clara que en las demás metodologías de desarrollo.

- ✓ Es orientada a componentes, posibilitando la independencia de funciones del sistema a la hora de mantener o modificar el sistema funcional.
- ✓ Utiliza como lenguajes de modelado BPMN para el negocio y UML para el diseño.
- ✓ Se definen claramente los roles que intervienen y las responsabilidades de cada uno de ellos, las actividades de desarrollo que se deben llevar a cabo, bien descritas y que roles las realizan, así como los artefactos y los roles que lo generan. (19)

Se mantiene el objetivo fundamental de todo proceso en el que se define “quién” está haciendo “qué”, “cuándo” y “cómo” para alcanzar un determinado objetivo. Un Proceso de Desarrollo de Software es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto. (20)

1.6.2 Lenguajes de Modelado.

BPMN (Business Process Management Notation): Es una notación que modela los procesos de negocio, basada en diagramas de flujo fácil de entender. Está diseñada para cubrir varios tipos de modelado, permite la creación tanto de segmentos de procesos, como procesos de negocio de comienzo a fin en diferentes niveles de representatividad. BPMN es gráficamente más rico, con menos símbolos fundamentales, pero con más variaciones de éstos, lo que facilita su comprensión por parte de personas no expertas.

- ✓ Provee una notación que es fácilmente entendida por todos los usuarios, desde el analista de negocio, el desarrollador técnico y hasta la propia gente del negocio.
- ✓ Crea un puente estandarizado para el vacío existente entre el diseño del proceso de negocio y su implementación.
- ✓ Asegura que los lenguajes para la ejecución de los procesos de negocio puedan ser visualizados con una notación común. (21)

UML (Unified Modeling Language): Es un lenguaje que permite visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. (22) UML es un conjunto de herramientas, que permite modelar (analizar y diseñar) sistemas orientados a objetos. (23)

Se ha convertido en el estándar de facto de la industria. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otra rama. UML cuenta con varios tipos de diagramas que son la representación gráfica de un conjunto de elementos y sus relaciones que visualizan el sistema desde diferentes perspectivas. (24)

1.6.3 Herramienta CASE.

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son el mejor soporte para el proceso de desarrollo de software, es 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. (25)

Visual Paradigm para UML es una herramienta CASE multiplataforma de modelado visual UML, muy potente y fácil de utilizar. Soporta el ciclo de vida completo del desarrollo de software, ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

- ✓ Soporte de UML versión 2.1.
- ✓ Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
- ✓ Ingeniería inversa - Código a modelo, código a diagrama.
- ✓ Generación de código - Modelo a código, diagrama a código.
- ✓ Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- ✓ Poderosa herramienta de generación de PDF/HTML a partir de diagramas UML.
- ✓ Sincronización entre el código fuente y el modelo en tiempo real. (26)

1.7 Fundamentación de las herramientas, lenguajes y tecnologías.

Todas las herramientas, metodología, lenguajes y tecnologías relacionadas en el presente trabajo, fueron seleccionadas por la dirección de desarrollo de software del proyecto ERP-Cuba y están definidas en el Manual del Marco de Trabajo (ARQUCID). Para la modelación del sistema de gestión de incidencias se utilizará como herramienta de modelado el Visual Paradigm, que aunque es propietaria, la universidad posee una licencia para el trabajo con

dicha herramienta ya que es una de las más potentes que existen en la actualidad y es multiplataforma. Como metodología a emplear para la construcción del mismo se utilizará el Modelo de Desarrollo orientado a componentes del proyecto ERP –Cuba, para que cada uno de los equipos de desarrollo posean un modelo estandarizado, así como una definición clara y precisa de las responsabilidades de cada uno de los roles que se ven involucrados en el desarrollo de la solución y como lenguaje de modelado se empleará BPMN para el negocio y UML para el diseño.

Se utilizará PHP como lenguaje de programación del lado del servidor por las grandes ventajas que este presenta, es un lenguaje sencillo pero muy potente, multiplataforma, rápido y probado por sus seguidores, diseñado para soportar perfectamente sitios con tráfico muy denso y es código abierto. Como lenguaje de programación del lado del cliente JavaScript, utilizándose la tecnología AJAX y el framework EXT JS que utiliza el lenguaje javascript para simular aplicaciones de escritorio. Y como gestor de base de datos, PostgreSQL.

1.8 Conclusiones.

Una vez analizados los sistemas existentes, las condiciones y los problemas que caracterizan a los procesos de gestión de incidencia de los trabajadores, se determinó la necesidad de una aplicación web, soportada sobre software libre, que sustituya los sistemas existentes y cumpla con las necesidades de las entidades empresariales y presupuestadas. La propuesta de solución es realizar el análisis y diseño de un sistema con las características expresadas anteriormente y que gestione las incidencias de los trabajadores.

Capítulo 2. Descripción del sistema.

2.1 Introducción.

El presente capítulo muestra las características que tendrá el sistema, partiendo de la representación del negocio mediante el proceso que interviene y su descripción, el modelo conceptual, la definición de los requerimientos del sistema y la especificación de los requisitos de software.

2.2 Reglas del Negocio.

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. (27)

En este caso se explicará la regla de acción de flujo para la realización de la pre-nómina.

Para realizar la pre-nómina se debe hacer un resumen de todas las incidencias de los trabajadores del período en el que se efectuará el pago a los mismos. Inicialmente se debe tener un conjunto de trabajadores y a cada uno de ellos se les debe controlar cada una de las incidencias que han tenido en un período de pago fijado, cuando se toman este conjunto de incidencias de los trabajadores se procede a realizar la pre-nómina para luego procesar la nómina y efectuar el pago.

2.3 Mapa de procesos y descripción del negocio.

Para una mayor comprensión del sistema que se desea desarrollar es necesario partir de la modelación del negocio, en este caso se modelará por procesos. El modelado de procesos del negocio consiste en describir la realidad de manera que esta pueda ser entendida y de ser necesario modificada con el fin de incorporarle mejoras. Es importante contar con una notación que permita modelar con la mayor claridad posible la esencia del negocio, permitiendo obtener una visión del funcionamiento de una empresa u organización determinada, para ello se utiliza la notación BPMN.

En la mayoría de las entidades empresariales y presupuestadas, la gestión de las incidencias de los recursos humanos es llevada a cabo de forma manual. Existe un proceso llamado utilización del fondo de tiempo laboral que abarca una serie de actividades interrelacionadas entre sí, que permiten efectuar el pago de los trabajadores en dependencia del tiempo realmente laborado, así como realizar estudios sobre el aprovechamiento del tiempo laboral, el mismo comprende:

- ✓ Control de asistencia.
- ✓ Elaboración de la pre-nómina.
- ✓ Análisis sobre la utilización del fondo de tiempo laboral.
- ✓ Confección del modelo de control de ausencias y sus causales.

En el cual se encuentra incluido el proceso de gestión de las incidencias que en su mayoría es generado del control de asistencia y que representa la entrada para la realización de la pre-nómina y el procesamiento de las nóminas.

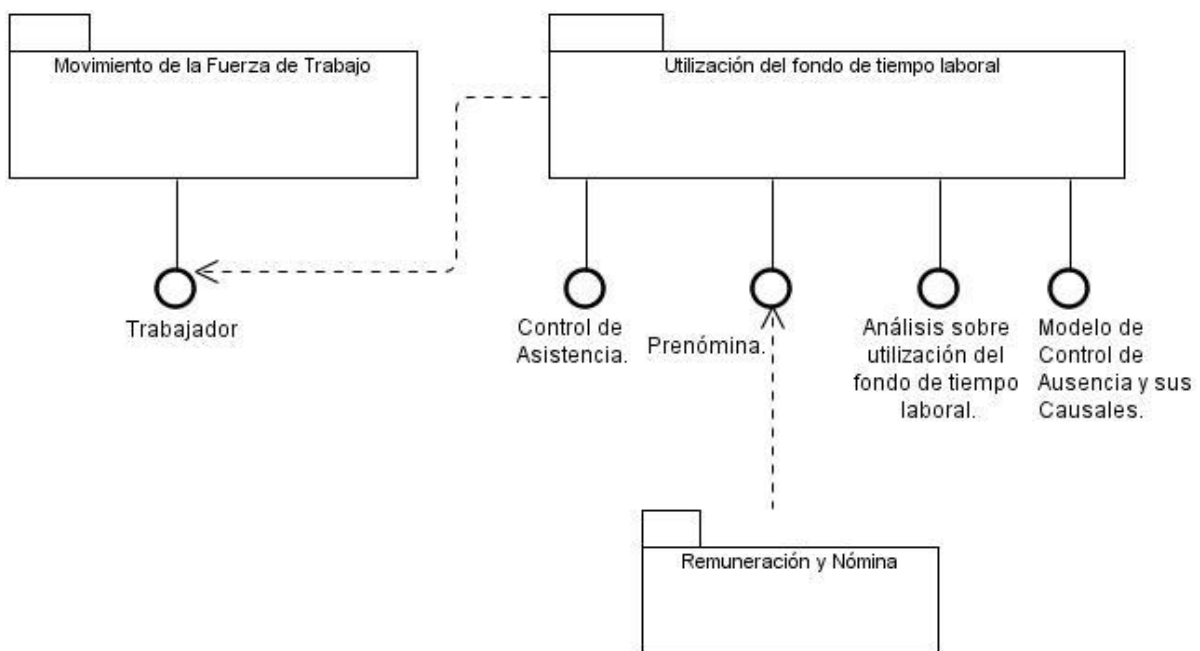


Figura 1. Mapa de Procesos

2.4 Técnicas utilizadas en las actividades de la ingeniería de requisitos.

Entrevistas y Cuestionarios

Las entrevistas y cuestionarios se emplean para reunir información proveniente de personas o de grupos. Durante la entrevista, el analista conversa con el encuestado; el cuestionario consiste en una serie de preguntas relacionadas con varios aspectos de un sistema. Por lo común, los encuestados son usuarios de los sistemas existentes o usuarios en potencia del sistema propuesto. En algunos casos, son gerentes o empleados que proporcionan datos para el sistema propuesto o que serán afectados por él. El éxito de esta técnica, depende de la habilidad del entrevistador y de su preparación para la misma.

Sistemas existentes

Esta técnica consiste en analizar distintos sistemas ya desarrollados que están relacionados con el sistema a ser construido. Por un lado, se pueden analizar las interfaces de usuario, observando el tipo de información que se maneja y cómo es manejada, por otro lado también es útil analizar las distintas salidas que los sistemas producen (listados, consultas, etc.), porque siempre pueden surgir nuevas ideas sobre la base de estas.

Prototipos

Durante la actividad de extracción de requerimientos, puede ocurrir que algunos requerimientos no estén demasiado claros o que no se esté muy seguro de haber entendido correctamente los requerimientos obtenidos hasta el momento, todo lo cual puede llevar a un desarrollo no eficaz del sistema final. Entonces, para validar los requerimientos hallados, se construyen prototipos. Los prototipos son simulaciones del posible producto, que luego son utilizados por el usuario final, permitiendo conseguir una importante retroalimentación en cuanto a si el sistema diseñado con base a los requerimientos recolectados le permite al usuario realizar su trabajo de manera eficiente y efectiva. El desarrollo del prototipo comienza con la captura de requerimientos. Desarrolladores y clientes se reúnen y definen los objetivos globales del software, identifican todos los requerimientos que son conocidos, y señalan áreas en las que será necesaria la profundización en las definiciones. Luego de esto, tiene lugar un “diseño rápido”. El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles al usuario (por ejemplo, entradas y formatos de las salidas). El diseño rápido lleva a la construcción de un prototipo.

Glosario

El glosario es una simple lista de términos en donde se explica su significado. En esta lista se incluyen y definen todos los términos que requieren explicación, mejorando así la comunicación intergrupala y la comunicación con el cliente, y mitigando el riesgo de malos entendidos. Los términos que se incluyen provienen de todas las áreas del proyecto. El glosario se va actualizando durante el transcurso del proceso de Ingeniería de Requisitos, perfeccionándolo en cada nuevo ciclo.

Arqueología de documentos

Con la aplicación de esta herramienta se tratan de determinar posibles requerimientos sobre la base de inspeccionar la documentación utilizada por la empresa. Esta herramienta sirve más que nada como complemento de las demás técnicas, y ayuda a obtener información

que de otra manera sería sumamente difícil conseguir. Se debe recolectar cualquier formulario o documento que sea utilizado para registrar o enviar información. (28)

Todas las técnicas antes mencionadas se tuvieron en cuenta para realizar el levantamiento de requisitos.

2.5 Representación del Modelo Conceptual.

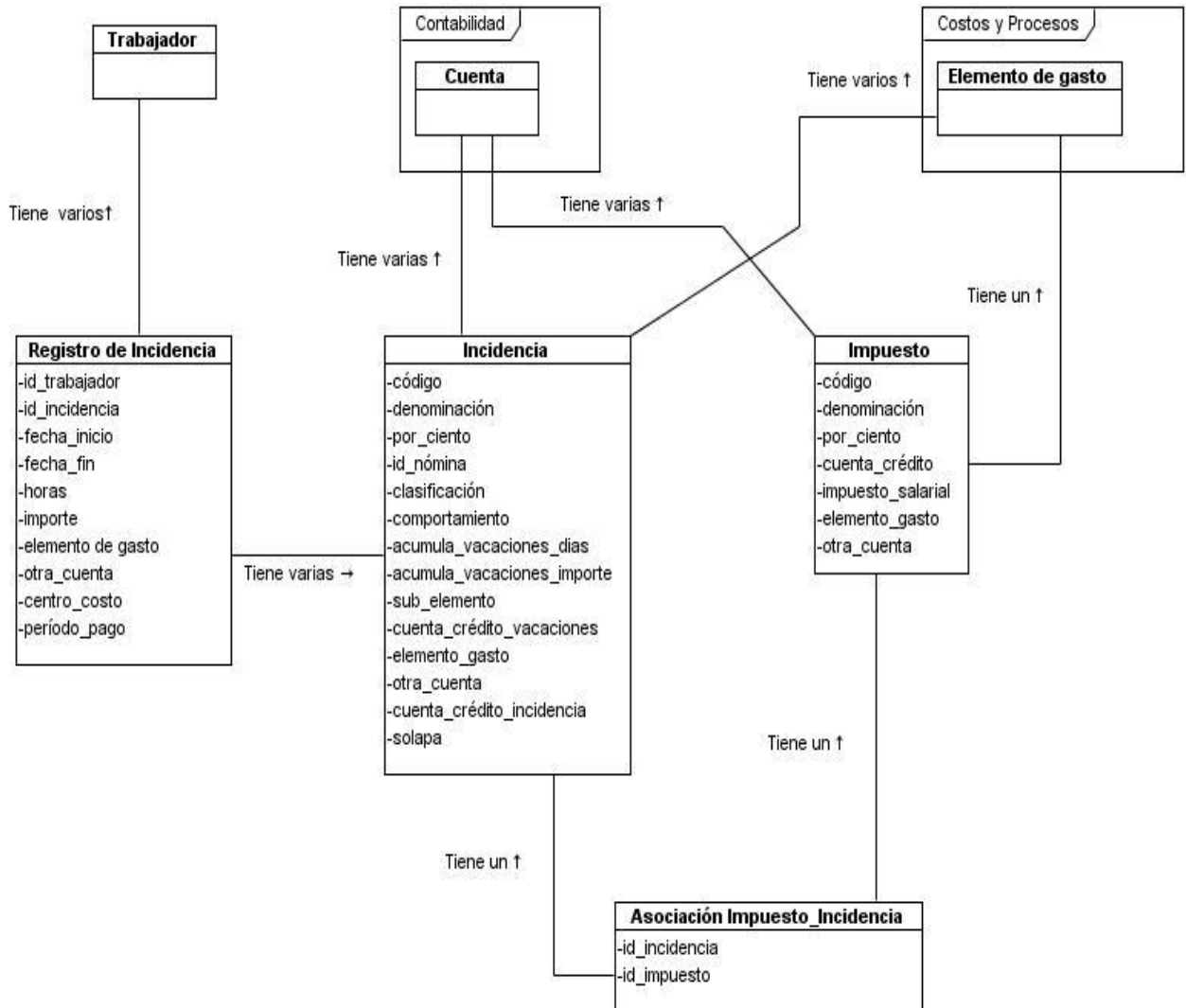


Figura 2. Modelo Conceptual.

2.5.1 Conceptos del modelo conceptual y sus descripciones.**Tabla 1 Conceptos y descripciones.**

| Concepto | Descripción |
|---------------------------------|---|
| Incidencia: | En esta entidad se definen los distintos tipos de incidencias. |
| Impuesto: | En esta entidad se definen los distintos conceptos de impuestos empresariales que afectan a la empresa. |
| Registro de Incidencia: | Es el registro de las incidencias de un trabajador en un período de pago. |
| Asociación impuesto-incidencia: | En esta entidad se registra por cada incidencia el o los impuestos asociados que debe pagar la empresa por la misma en caso de que lo(s) tenga. |
| Elemento de gasto: | En esta entidad se definen los distintos elementos de gastos que permiten la cuantificación de los gastos. |
| Cuenta: | En esta entidad se definen las distintas cuentas que permiten realizar el registro contable de un hecho económico. |
| Trabajador: | En esta entidad se registran los trabajadores de la empresa. |

2.6 Patrón utilizado para los requisitos.

¿Qué es un patrón?

El patrón es una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas. (29)

Patrón CRUD (Creating, Reading, Updating, Deleting)

Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual.

Completo

Este patrón consta de un caso de uso, llamado Información CRUD o gestionar información modela todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico, tales como creación, lectura, actualización y eliminación. Suele ser utilizado cuando todos los flujos contribuyen al mismo valor del negocio, y estos a su vez son cortos y simples.

Parcial

Este patrón alternativo modela una de las vías de los casos de uso como un caso de uso separado. Es preferiblemente utilizado cuando una de las alternativas de los casos de uso es mas significativa, larga o más compleja que las otras.

En los requerimientos identificados se pone de manifiesto el patrón CRUD, que aunque está dirigido a casos de uso están presente tanto el completo como el parcial en los requisitos de software definidos.

2.7 Especificación de requisitos.

Para dar cumplimiento al objetivo planteado y conocidos ya los conceptos que rodean al objeto de estudio, se deben definir una serie de requisitos funcionales que sean capaces de resolver el problema existente.

2.7.1 Requisitos funcionales del sistema.**RF1. Requisito funcional Gestionar tipos de incidencia.**

- RF1.1 Adicionar tipo de incidencia.
- RF1.2 Modificar tipo de incidencia.
- RF1.3 Eliminar tipo de incidencia.

RF2. Requisito funcional Gestionar tipos de impuestos.

- RF2.1 Adicionar tipo de impuesto.
- RF2.2 Modificar tipo de impuesto.
- RF2.3 Eliminar tipo de impuesto.

RF3. Requisito funcional Gestionar asociación de incidencia-impuesto.

- RF3.1 Adicionar asociación de incidencia-impuesto.
- RF3.2 Eliminar asociación de incidencia-impuesto.

RF4. Requisito funcional Registrar incidencias.

- RF4.1 Adicionar registro de incidencia.
- RF4.2 Modificar registro de incidencia.
- RF4.3 Eliminar registro de incidencia.

2.7.2 Especificaciones de los requisitos.**RF1. Requisito funcional Gestionar tipos de incidencia.**

Este requisito gestiona las posibles definiciones de los tipos de incidencias que afectan al salario del trabajador, ya sea para sumarle o restarle, se utiliza para el cálculo del salario a devengar. Según el tipo de incidencia puede constituir gastos o no a las cuentas de Contabilidad y a los elementos de gastos de Costo y Procesos por lo que las mismas son contabilizadas.

RF1.1 Especificación del requisito Adicionar tipo de incidencia.

Tabla 2 Especificación del requisito Adicionar Tipo de Incidencia.

| Conceptos tratados | Conceptos | Atributos |
|---------------------------|--|---|
| | Tipo de Incidencia. | Código, denominación, comportamiento, clasificación, porciento, solapado, acumula tiempo vacaciones, acumula importe vacaciones, cuenta crédito vacaciones, subelemento, elemento de gasto, otra cuenta, cuenta crédito incidencia. |
| Precondiciones | Precondiciones | Pre-requisito |
| | No procede. | No procede. |
| Descripción | <p>El sistema permite adicionar los siguientes datos:</p> <p>Código: Número definido por el usuario que identifica el tipo de incidencia.</p> <p>Denominación: Nombre que el usuario le asigna al tipo de incidencia.</p> <p>Comportamiento: Seleccionar si es de suma o de resta en dependencia de la incidencia, es decir si suma o resta al salario que será devengado.</p> <p>Clasificación: Seleccionar la clasificación de la incidencia que pueden ser de tipo: vacaciones, subsidio, pago y otra.</p> <p>Porciento: Un número que representa el porciento.</p> | |

| | |
|--|---|
| | <p>Solapado: Es para que cuando se marque no ocurra otra incidencia al mismo tiempo.</p> <p>Acumula tiempo vacaciones: Marcado significa que acumula tiempo de vacaciones, desmarcado significa que no acumula.</p> <p>Acumula importe vacaciones: Define la forma de tratamiento de la acumulación del importe de vacaciones. Los valores posibles son: acumular, no acumular y acumular y liquidar.</p> <p>Subelemento del gasto: Generalmente el importe de las vacaciones acumuladas junto con el salario devengado constituyen un gasto que se contabiliza a la cuenta y el centro de costo que se ponga en la nómina. En ese caso la acumulación de vacaciones se representa con el subelemento que se fije. Si este campo se deja en blanco se está asumiendo que ambos importes se van a debitar a una cuenta que no es de gastos.</p> <p>Cuenta crédito de vacaciones: Es la cuenta de Provisión de Vacaciones. Generalmente las entidades definen una sola cuenta con este propósito pero de forma excepcional pueden utilizarse dos o más cuentas de provisión, esta se usará para las vacaciones.</p> <p>Elementos de gastos: Se selecciona el elemento de gasto del nomenclador de Costos y procesos.</p> <p>Otra cuenta: Al ser marcada esta cuenta el sistema limpia el campo del subelemento de gastos y al hacer la contabilización de la nómina desconoce el subelemento asociado a la acumulación de vacaciones.</p> <p>Cuenta de crédito de la incidencia: Seleccionar la cuenta crédito del nomenclador cuenta de contabilidad es la cuenta a la cual se va a acreditar el salario devengado, que generalmente es la cuenta de Nóminas por Pagar.</p> <p>Indica adicionar el tipo de incidencia.</p> |
|--|---|

| | |
|-------------------------|--|
| | <p>Se validan los datos.</p> <p>En caso de error en la entrada de datos informar al usuario y permitir corregirlos.</p> <p>Si los datos son correctos informar al usuario que se ha adicionado satisfactoriamente el tipo de incidencia.</p> |
| Validaciones | <p>El sistema valida los datos según lo descrito en el Modelo Conceptual: <u>ERP- CH-Modelo Conceptual v1.7.doc</u></p> <p>Si la clasificación de la incidencia es del tipo Vacaciones, se bloquea la opción de insertar estos atributos:</p> <p>Acumula tiempo vacaciones.</p> <p>Acumula importe vacaciones.</p> <p>Sub_Elemento.</p> <p>Cuenta Crédito Vacaciones</p> |
| Post-condiciones | Se ha adicionado un nuevo tipo de incidencia. |
| Post-requisito | No procede. |

RF1.2 Especificación del requisito Modificar tipo de incidencia.

Tabla 3 Especificación del requisito Modificar Tipo de Incidencia.

| Conceptos tratados | Conceptos | Atributos |
|---------------------------|-----------------------|---|
| | Tipo de incidencia. | Código, denominación, comportamiento, clasificación, porciento, solapado, acumula tiempo vacaciones, acumula importe vacaciones, cuenta crédito vacaciones, subelemento, elemento de gasto, otra cuenta, cuenta crédito incidencia. |
| Precondiciones | Precondiciones | Pre-requisito |
| | Que se haya | Adicionar tipo de incidencia. |

| | | |
|-------------------------|--|--|
| | adicionado un tipo de incidencia. | |
| Descripción | <p>El sistema permite modificar el tipo de incidencia.</p> <p>Se escoge el tipo de incidencia a modificar, se deben deshabilitar los datos que no se pueden modificar como el código.</p> <p>Se modifican los datos del tipo de incidencia.</p> <p>Indica modificar el tipo de incidencia.</p> <p>El sistema valida los datos.</p> <p>En caso de error en la entrada de datos informar al usuario y permitir corregirlos.</p> <p>Si los datos son correctos informar al usuario que se ha modificado satisfactoriamente el tipo de incidencia.</p> | |
| Validaciones | <p>El sistema valida los datos según lo descrito en el Modelo Conceptual: <u>ERP- CH-Modelo Conceptual v1.7.doc</u></p> <p>Se debe garantizar que los datos que no sean posibles modificar aparezcan deshabilitados. Ejemplo: código.</p> | |
| Post-condiciones | Se ha modificado un tipo de incidencia. | |
| Post-requisito | No procede. | |

RF1.3 Especificación del requisito Eliminar tipo de incidencia.

Tabla 4 Especificación del requisito Eliminar Tipo de Incidencia.

| Conceptos tratados | Conceptos | Atributos |
|---------------------------|---------------------|--|
| | Tipo de incidencia. | Código, denominación, comportamiento, clasificación, porciento, solapado, acumula tiempo vacaciones, acumula importe vacaciones, cuenta crédito vacaciones, subelemento, elemento de gasto, otra |

| | | |
|-------------------------|--|------------------------------------|
| | | cuenta, cuenta crédito incidencia. |
| Precondiciones | Precondiciones | Pre-requisito |
| | Que se haya adicionado un tipo de incidencia. | Adicionar tipo de incidencia. |
| Descripción | <p>Se escoge el tipo de incidencia que se quiere eliminar.</p> <p>Indica eliminar el tipo de incidencia.</p> <p>Se muestra un mensaje al usuario de confirmación. Ejemplo: ¿Desea eliminar el tipo de incidencia?</p> <p>Si se confirma se elimina el tipo de incidencia seleccionada y se le muestra un mensaje al usuario.</p> | |
| Validaciones | No se pueden eliminar tipos de incidencias de las cuales se hayan registrado incidencias. | |
| Post-condiciones | Se ha eliminado el tipo de incidencia. | |
| Post-requisito | No procede. | |

RF2. Requisito funcional Gestionar Tipos de impuestos.

En el requisito se gestionan los tipos de impuestos que se tendrán en una entidad y que luego serán asociados a una determinada incidencia, ejemplo de tipos de impuestos son: impuesto por utilización de fuerza de trabajo, contribución especial a la seguridad social, etc.

RF2.1 Especificación del requisito Adicionar tipo de impuesto.

Tabla 5 Especificación del requisito Adicionar Tipo de Impuesto.

| Conceptos tratados | Conceptos | Atributos |
|---------------------------|-------------------|--|
| | Tipo de impuesto. | Código, denominación, porciento, cuenta de crédito, impuesto salarial, elemento del gasto y otra cuenta. |

| Precondiciones | Precondiciones | Pre-requisito |
|--------------------|--|---------------|
| | No procede. | No procede. |
| Descripción | <p>El sistema permite adicionar los siguientes datos del impuesto:</p> <p>Código: Número definido por el usuario que identifica el tipo de impuesto.</p> <p>Denominación: Nombre que el usuario le asigna al tipo de impuesto.</p> <p>Porcentaje: Un número que representa el porcentaje.</p> <p>Cuenta de crédito: Es la cuenta a la cual se va a acreditar el salario devengado que generalmente es la cuenta de Nóminas por Pagar.</p> <p>Impuesto salarial: Marcado significa que es un impuesto salarial, sin marcar significa que no lo es.</p> <p>Elemento de Gasto: Si se marca se está asumiendo que la contabilización del gasto se va a hacer a la combinación de Cuenta de Gastos y Centro de Costo que se asuma o se especifique en la Nómina.</p> <p>Otra cuenta: Si se marca otra cuenta se asume que ésta no es de gasto, y que por lo tanto no se requiere de Elemento del Gasto para la contabilización.</p> <p>Si el impuesto es un impuesto salarial se deben deshabilitar los campos:</p> <p>Elemento de Gasto.</p> <p>Otra cuenta.</p> <p>Se indica adicionar el tipo de impuesto.</p> <p>El sistema valida los datos.</p> <p>En caso de error en la entrada de datos informar al usuario y permitir corregirlos.</p> <p>Si los datos son correctos informar al usuario que se ha adicionado</p> | |

| | |
|-------------------------|--|
| | satisfactoriamente el tipo de impuesto. |
| Validaciones | El sistema valida los datos según lo descrito en el Modelo Conceptual: <u>ERP- CH-Modelo Conceptual v1.7.doc</u> Si el tipo de impuesto es un impuesto salarial se bloquea la opción de insertar estos atributos: Elemento del Gasto. Otra cuenta. |
| Post-condiciones | Se ha adicionado un nuevo tipo de impuesto. |
| Post-requisito | No procede. |

RF2.2 Especificación del requisito Modificar tipo de impuesto.

Tabla 6 Especificación del requisito Modificar Tipo de Impuesto.

| Conceptos tratados | Conceptos | Atributos |
|---------------------------|--|--|
| | Tipo de impuesto. | Código, denominación, porciento, cuenta de crédito, impuesto salarial, elemento del gasto y otra cuenta. |
| Precondiciones | Precondiciones | Pre-requisito |
| | Que se haya adicionado un tipo de impuesto. | Adicionar tipo de impuesto. |
| Descripción | <p>Se escoge el tipo de impuesto a modificar, se deben deshabilitar los datos que no se pueden modificar como el código.</p> <p>Se modifica el o los datos del tipo de impuesto.</p> <p>Se indica modificar el tipo de impuesto.</p> <p>El sistema valida los datos.</p> <p>En caso de error en la entrada de datos informar al usuario y permitir</p> | |

| | |
|-------------------------|--|
| | <p>corregirlos.</p> <p>Si los datos son correctos informar al usuario que se ha modificado satisfactoriamente el tipo de impuesto.</p> |
| Validaciones | <p>El sistema valida los datos según lo descrito en el Modelo Conceptual: <u>ERP- CH-Modelo Conceptual v1.7.doc</u></p> <p>Se debe garantizar que los datos que no sean posibles modificar aparezcan deshabilitados. Ejemplo: el código.</p> |
| Post-condiciones | Se ha modificado el tipo de impuesto. |
| Post-requisito | No procede. |

RF2.3 Especificación del requisito Eliminar tipo de impuesto.

Tabla 7 Especificación del requisito Eliminar Tipo de Impuesto.

| Conceptos tratados | Conceptos | Atributos |
|---------------------------|---|--|
| | Tipo de impuesto. | Código, denominación, porciento, cuenta de crédito, impuesto salarial, elemento del gasto y otra cuenta. |
| Precondiciones | Precondiciones | Pre-requisito |
| | Que se haya adicionado un tipo de impuesto. | Adicionar tipo de impuesto. |
| Descripción | <p>El sistema permite exportar datos del tipo de impuesto.</p> <p>Permite buscar la ubicación física hacia donde se exportará el fichero y nombrar el mismo.</p> <p>Al indicar la carpeta se muestra en pantalla un listado con los datos a exportar, si se confirma se exportan los datos.</p> <p>Se acepta.</p> | |

| | |
|-------------------------|---|
| | Se notifica al usuario. |
| Validaciones | No procede. |
| Post-condiciones | Se exportan los datos del tipo de impuesto. |
| Post-requisito | No procede. |

RF3. Requisito funcional Gestionar asociación de incidencia-impuesto.

Existen incidencias que llevan asociados impuestos por lo que es necesario hacer la asociación del impuesto y la incidencia para que cuando se aplique una incidencia al trabajador ya tenga el impuesto asociado.

RF3.1 Especificación del requisito Adicionar asociación de incidencia-impuesto.

Tabla 8 Especificación del requisito Adicionar asociación incidencia-impuesto.

| Conceptos tratados | Conceptos | Atributos |
|--------------------|--|---|
| | Tipo de impuesto. | Código, denominación, porciento, cuenta de crédito, impuesto salarial, elemento del gasto y otra cuenta. |
| | Tipo de Incidencia. | Código, denominación, comportamiento, clasificación, porciento, solapado, acumula tiempo vacaciones, acumula importe vacaciones, cuenta crédito vacaciones, subelemento, elemento de gasto, otra cuenta, cuenta crédito incidencia. |
| Precondiciones | Precondiciones | Pre-requisito |
| | Que se haya adicionado al menos un tipo de incidencia. | Adicionar tipo de incidencia. |
| | Que se haya | Adicionar tipo de impuesto. |

| | | |
|-------------------------|---|--|
| | adicionado al menos un tipo de impuesto. | |
| Descripción | El sistema permite seleccionar un tipo de incidencia e indicar adicionar tipo de impuesto. Muestra el listado de tipos de impuestos y permite adicionar los asociados a la incidencia. | |
| Validaciones | El sistema valida los datos según lo descrito en el Modelo Conceptual: <u>ERP- CH-Modelo Conceptual v1.7.doc.</u> | |
| Post-condiciones | Se ha asociado un tipo de impuesto a un tipo de incidencia. | |
| Post-requisito | No procede. | |

RF3.2 Especificación del requisito Eliminar asociación de incidencia-impuesto.

Tabla 9 Especificación del requisito Eliminar asociación incidencia-impuesto.

| Conceptos tratados | Conceptos | Atributos |
|---------------------------|--|---|
| | Tipo de impuesto. | Código, denominación, porciento, cuenta de crédito, impuesto salarial, elemento del gasto y otra cuenta. |
| | Tipo de Incidencia. | Código, denominación, comportamiento, clasificación, porciento, solapado, acumula tiempo vacaciones, acumula importe vacaciones, cuenta crédito vacaciones, subelemento, elemento de gasto, otra cuenta, cuenta crédito incidencia. |
| Precondiciones | Precondiciones | Pre-requisito |
| | Tiene que existir al menos una asociación incidencia - impuesto. | Adicionar asociación de incidencia-impuesto. |

| | |
|-------------------------|---|
| Descripción | <p>Se selecciona la opción eliminar asociación incidencia-impuesto, el sistema permite escoger la asociación a eliminar.</p> <p>Se indica eliminar el impuesto asociado a la incidencia.</p> <p>Se muestra un mensaje de confirmación.</p> <p>Si se confirma se elimina el impuesto asociado a la incidencia.</p> |
| Validaciones | <p>El sistema valida los datos según lo descrito en el Modelo Conceptual: <u>ERP- CH-Modelo Conceptual v1.7.doc</u></p> <p>Al eliminar una asociación, pasará a un estado inactivo para mantener los datos anteriormente registrados.</p> |
| Post-condiciones | Se ha eliminado la asociación de un impuesto a una incidencia. |
| Post-requisito | No procede. |

RF4. Requisito funcional Registrar incidencias.

Este requisito permite hacer un registro de todas las incidencias que tiene un trabajador, en un período de pago que es reflejado en el salario ya sea para sumarle o restarle al mismo.

RF4.1 Especificación del requisito Adicionar registro de incidencia

Tabla 10 Especificación del requisito Adicionar Registro de Incidencia.

| Conceptos tratados | Conceptos | Atributos |
|---------------------------|--------------------------|--|
| | Tipo Incidencia. | Código, denominación. |
| | Trabajador. | Expediente laboral (CI), nombre, primer apellido, segundo apellido. |
| | Registro de Incidencias. | Fecha inicio, fecha fin, importe, horas, elemento de gasto, centro de costo, período de pago, otra cuenta. |
| Precondiciones | Precondiciones | Pre-requisito |

| | | |
|--------------------|---|-------------------------------|
| | Se ha adicionado un tipo de incidencia. | Adicionar tipo de incidencia. |
| | Se ha adicionado un trabajador. | Emitir movimiento de alta. |
| Descripción | <p>Seleccionar la opción Registro de incidencias.</p> <p>Se busca al trabajador mediante alguno de los criterios de búsqueda siguientes:</p> <p>Expediente laboral (CI).</p> <p>Nombre.</p> <p>Primer apellido.</p> <p>Segundo apellido.</p> <p>Si la búsqueda resulta insatisfactoria, se debe mostrar un mensaje que advierta al usuario.</p> <p>Después de encontrado el trabajador se habilitan los tipos de incidencias, donde se escoge el tipo de incidencia a asociarle al trabajador.</p> <p>Luego se llenan los datos para hacer el registro de la incidencia al trabajador:</p> <p>Fecha inicio: Es la fecha en el que se inicia la incidencia.</p> <p>Fecha fin: Es la fecha en la que termina la incidencia.</p> <p>Importe: Para los tipos de incidencia en las que el importe es requerido el sistema debe validar que se complete este atributo antes de registrar los datos.</p> <p>Horas: Para los tipos de incidencia en las que las horas son requeridas el sistema debe validar que se complete este atributo antes de registrar los datos.</p> <p>Elementos de gastos: Se muestra por defecto la que tiene registrada</p> | |

| | |
|-------------------------|--|
| | <p>el tipo de incidencia y se permite modificar.</p> <p>Otra cuenta: Se muestra por defecto la que tiene registrada el tipo de incidencia y se permite modificar.</p> <p>Centro de costo: Se muestra por defecto el que tiene el trabajador y se permite modificar.</p> <p>Se indica adicionar el registro de la incidencia.</p> <p>Se validan los datos.</p> <p>En caso de error en la entrada de datos informar al usuario y permitir corregirlos.</p> <p>Si los datos son correctos informar al usuario que se ha adicionado satisfactoriamente el registro de incidencia del trabajador.</p> |
| Validaciones | <p>La fecha de inicio no debe ser mayor que la fecha de fin.</p> <p>La suma de todas las horas de las incidencias que no se solapan tiene que ser menor o igual a la cantidad de horas de trabajo del período de pago en el que se encuentra.</p> <p>El sistema valida los datos según lo descrito en el Modelo Conceptual: <u>ERP- CH-Modelo Conceptual v1.7.doc</u></p> |
| Post-condiciones | Se le reportó una incidencia al trabajador en el registro de incidencias. |
| Post-requisito | No procede. |

RF4.2 Especificación del requisito Modificar registro de incidencia

Tabla 11 Especificación del requisito Modificar Registro de Incidencia.

| Conceptos tratados | Conceptos | Atributos |
|--------------------|------------------|---|
| | Tipo Incidencia. | Código, denominación. |
| | Trabajador. | Expediente laboral (CI), nombre, primer apellido, segundo apellido. |

| | | |
|-----------------------|--|--|
| | Registro de Incidencias | Fecha inicio, fecha fin, importe, horas, elemento de gasto, otra cuenta, período de pago, centro de costo. |
| Precondiciones | Precondiciones | Pre-requisito |
| | Se ha adicionado un tipo de incidencia. | Adicionar un tipo de incidencia. |
| | Se ha adicionado un trabajador. | Movimiento de alta. |
| | Se ha adicionado una incidencia al trabajador en el registro. | Adicionar registro de incidencia. |
| Descripción | <p>Seleccionar la opción Registro de Incidencia.</p> <p>Se busca al trabajador mediante alguno de los criterios de búsqueda siguientes:</p> <p>Nombre.</p> <p>Primer apellido.</p> <p>Segundo apellido.</p> <p>Expediente laboral (CI).</p> <p>Si la búsqueda resulta insatisfactoria, se debe mostrar un mensaje que advierta al usuario.</p> <p>Se muestran las incidencias que tiene registradas el trabajador.</p> <p>Se escoge el registro a modificar, se modifican los datos del registro de la incidencia.</p> <p>Se indica modificar el registro de incidencia.</p> <p>El sistema valida los datos:</p> | |

| | |
|-------------------------|--|
| | <p>Para los tipos de incidencia en las que el importe es requerido el sistema debe validar que se complete este atributo antes de registrar los datos.</p> <p>En caso de error en la entrada de datos informar al usuario y permitir corregirlos.</p> <p>Si los datos son correctos informar al usuario que se ha modificado satisfactoriamente el registro de incidencia.</p> |
| Validaciones | <p>La fecha de inicio no debe ser mayor que la fecha de fin.</p> <p>No se pueden modificar incidencias que hayan sido procesadas en una nómina.</p> <p>El sistema valida los datos según lo descrito en el Modelo Conceptual: <u>ERP- CH-Modelo Conceptual v1.7.doc</u></p> |
| Post-condiciones | <p>Se ha modificado el registro de incidencia.</p> <p>Se ha modificado la relación del tipo de incidencia al trabajador si el usuario lo ha indicado.</p> |
| Post-requisito | No procede. |

RF4.3 Especificación del requisito Eliminar registro de incidencia

Tabla 12 Especificación del requisito Eliminar Registro de Incidencia.

| Conceptos tratados | Conceptos | Atributos |
|--------------------|---|--|
| | Tipo Incidencia. | Código, denominación |
| | Trabajador. | Expediente laboral (CI), nombre, primer apellido, segundo apellido. |
| | Registro de Incidencias. | Fecha inicio, fecha fin, importe, horas, elemento de gasto, otra cuenta, período de pago, centro de costo. |
| Precondiciones | Precondiciones | Pre-requisito |
| | Se ha adicionado un tipo de incidencia. | Adicionar un tipo de incidencia. |
| | Se ha adicionado un trabajador. | Movimiento de alta. |
| | Se ha adicionado una incidencia al trabajador en el registro. | Adicionar registro de incidencia. |
| Descripción | <p>Se selecciona la opción Registro de incidencia.</p> <p>Se busca al trabajador mediante alguno de los criterios de búsqueda siguientes:</p> <p>Nombre.</p> <p>Primer apellido.</p> <p>Segundo apellido.</p> <p>Expediente laboral (CI).</p> <p>Si la búsqueda resulta insatisfactoria, se debe mostrar un mensaje</p> | |

| | |
|-------------------------|--|
| | <p>que advierta al usuario.</p> <p>Se muestran las incidencias que tiene registradas el trabajador.</p> <p>Se escoge el registro de incidencia a eliminar.</p> <p>Se indica eliminar el registro de incidencia.</p> <p>Se muestra un mensaje al usuario de confirmación. Ejemplo: ¿Desea eliminar el registro de incidencia?</p> <p>Si se confirma, se elimina el registro de incidencia seleccionada.</p> |
| Validaciones | No se pueden eliminar incidencias que hayan sido procesadas en una nómina. |
| Post-condiciones | Se le eliminó el registro de incidencia al trabajador. |
| Post-requisito | No procede. |

2.7.3 Reportes generados de la gestión de incidencias.

La generación de reportes es fundamental en los sistemas informáticos que utilizan bases de datos, ya que mediante ellos se obtiene la información de una forma más rápida y organizada. En muchas ocasiones, los reportes sirven como herramientas para la toma de decisiones de acuerdo al análisis de la información que se haya obtenido de los datos que se almacenen en la base de datos.

Los reportes que son generados del sistema para la gestión de incidencias son los siguientes (Ver anexo):

2.8 Prototipos de interfaz de usuarios propuestos.

Los prototipos de interfaz de usuario ayudan a identificar, comunicar y probar un producto antes de crearlo. La realización de los mismos, logra un entendimiento común entre el cliente y los desarrolladores, solucionando así la mayoría de los cambios posteriores en los proyectos de desarrollo de software. A continuación los prototipos de interfaz de usuario de los requisitos definidos (Ver anexo):

2.9 Conclusiones.

En este capítulo se representó el mapa de procesos de negocio, se determinaron los requisitos funcionales del sistema y se describieron los mismos. Partiendo de este punto se le puede dar comienzo a la construcción de la propuesta de solución, velando que a cada requerimiento se le de cumplimiento.

Capítulo 3. Diseño del sistema.

3.1 Introducción.

En el presente capítulo se realizará el diseño del sistema donde se traducirán los requisitos a una especificación que describe cómo implementar el sistema. Se transforman los requisitos al diseño del futuro sistema utilizando una arquitectura correcta, logrando adaptar el diseño para que sea consistente con el entorno de implementación.

3.2 Patrones utilizados.

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Identifica: clases, instancias, roles, colaboraciones y la distribución de responsabilidades.

3.2.1 Patrones de asignación de responsabilidades.

Para la construcción del diseño, usualmente se emplean un grupo de patrones o modelos, que no son más que soluciones concretas y técnicas para lograr objetivos específicos. Dentro de los más conocidos y usados se encuentran los GRASP (General Responsibility Assignment Software Patterns, Patrones Generales de Software para Asignación de Responsabilidades), que describen los principios fundamentales de la asignación de responsabilidades a objetos.

El empleo de los patrones GRASP tiene como objetivo la descripción de los principios fundamentales del diseño de objetos para la asignación general de responsabilidades. Los utilizados fueron:

- ✓ Alta cohesión.

Una clase con mucha cohesión es útil porque es bastante fácil darle mantenimiento, entenderla y reutilizarla. Su alto grado de funcionalidad, combinada con una reducida cantidad de operaciones, también simplifica el mantenimiento y los mejoramientos. La ventaja que significa una gran funcionalidad también soporta un aumento de la capacidad de reutilización.

El patrón alta cohesión presenta semejanzas con el mundo real. Se sabe que si alguien asume demasiadas responsabilidades, sobre todo las que debería delegar, no sería eficiente. (29)

- ✓ Bajo acoplamiento.

El bajo acoplamiento estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento.

El bajo acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad. (29)

3.2.2 Patrón GOF.

El patrón fachada clasificado como estructural dentro de los patrones GOF proporciona una interfaz sencilla unificada para un conjunto de clases o subsistemas, siendo más fácil de usar. Permite reducir la complejidad y minimizar las dependencias, el acceso de los clientes a los subsistemas es por medio de la clase fachada, ella es la encargada de reenviar las peticiones a los objetos de los subsistemas, por lo que no se accede directamente a los mismos, ocultando la complejidad de ellos. Este patrón favorece a un Bajo Acoplamiento entre los clientes y los subsistemas, respondiendo a uno de los patrones GRASP, permite variar las clases internas, de manera transparente a los clientes que las usan. (29)

Mediante el patrón fachada se logró proporcionar una interfaz de servicios simple para un subsistema tan complejo como lo es Capital Humano, al cual pertenece el componente a diseñar. Su uso en el nivel de abstracción más alto, obedece al diseño general de la arquitectura del sistema, puesto que la existencia de dependencias entre el subsistema Capital Humano y otros como por ejemplo: Contabilidad y Costos y Procesos, solicita la implementación de una clase fachada donde se publiquen los servicios necesarios para los otros subsistemas que facilite la interacción. A nivel interno también se emplea este patrón para disminuir el grado de dependencia entre los componentes.

3.2.3 Patrón de arquitectura Modelo Vista Controlador.

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

Modelo: Esta capa administra el comportamiento y los datos del dominio de la aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde

la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).

Vista: Esta capa maneja la visualización de la información, es decir que presenta el modelo en un formato adecuado para interactuar, que usualmente es la interfaz de usuario.

Controlador: Esta capa controla el flujo de datos entre la vista y el modelo; es decir que responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. (30)

Esta separación permite construir y probar el modelo, independientemente de la representación visual. En el caso de los diagramas de clases realizados para el diseño la página phtml representa la vista, las clases controller el controlador, y en el modelo se representan las clases de la lógica del negocio y las de dominio que son las de acceso a datos, quedando así hecha la representación de este patrón y logrando que cualquier cambio que se realice en la vista no afecte ni la lógica del negocio, ni el dominio.

3.3 Diagramas de clases del diseño.

En el diseño se modela el sistema y se encuentra su forma, incluyendo la arquitectura, para que soporte todos los requisitos y otras restricciones relacionadas con el entorno de la implementación, tiene impacto en el sistema a desarrollar. Además, el modelo de diseño sirve de abstracción de la implementación del sistema y es utilizado como entrada fundamental de las actividades de implementación.

Las siguientes ilustraciones muestran los diagramas de clases del diseño:

Tipo de incidencia

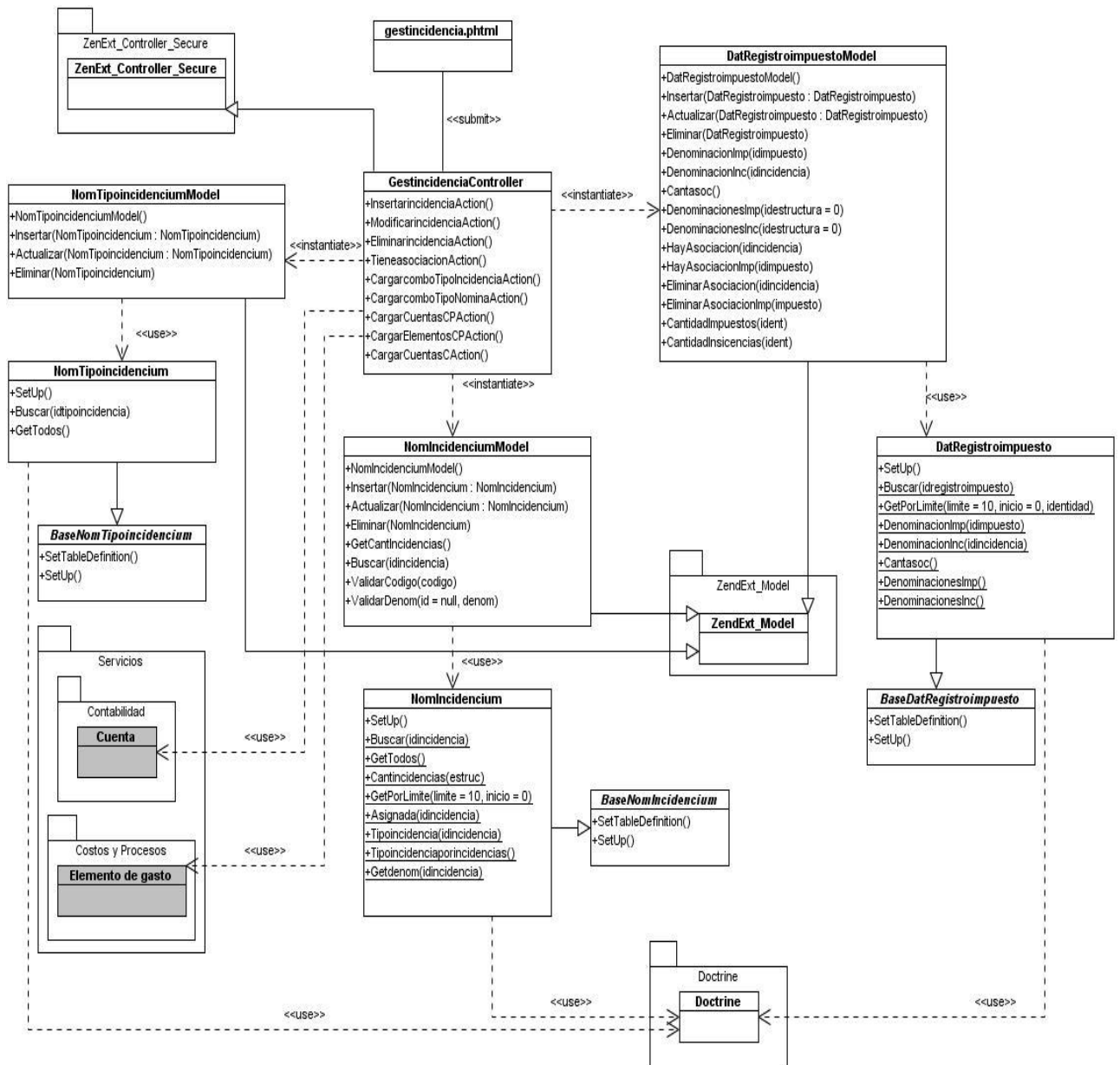


Figura 3. Diagrama de clases del diseño de Tipo de Incidencia.

Tipo de impuesto

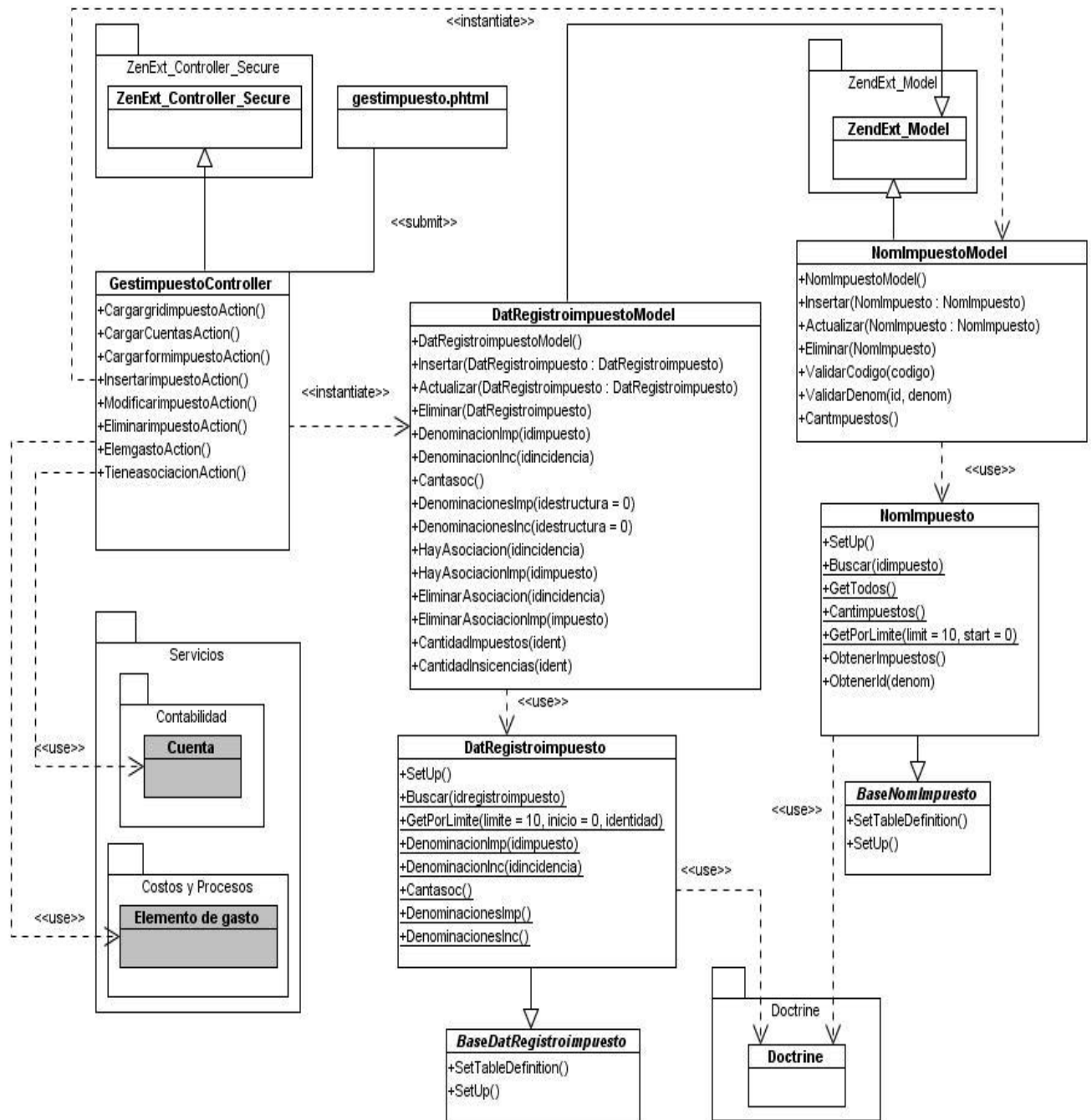


Figura 4. Diagrama de clases del diseño de Tipo de Impuesto.

Registrar incidencias

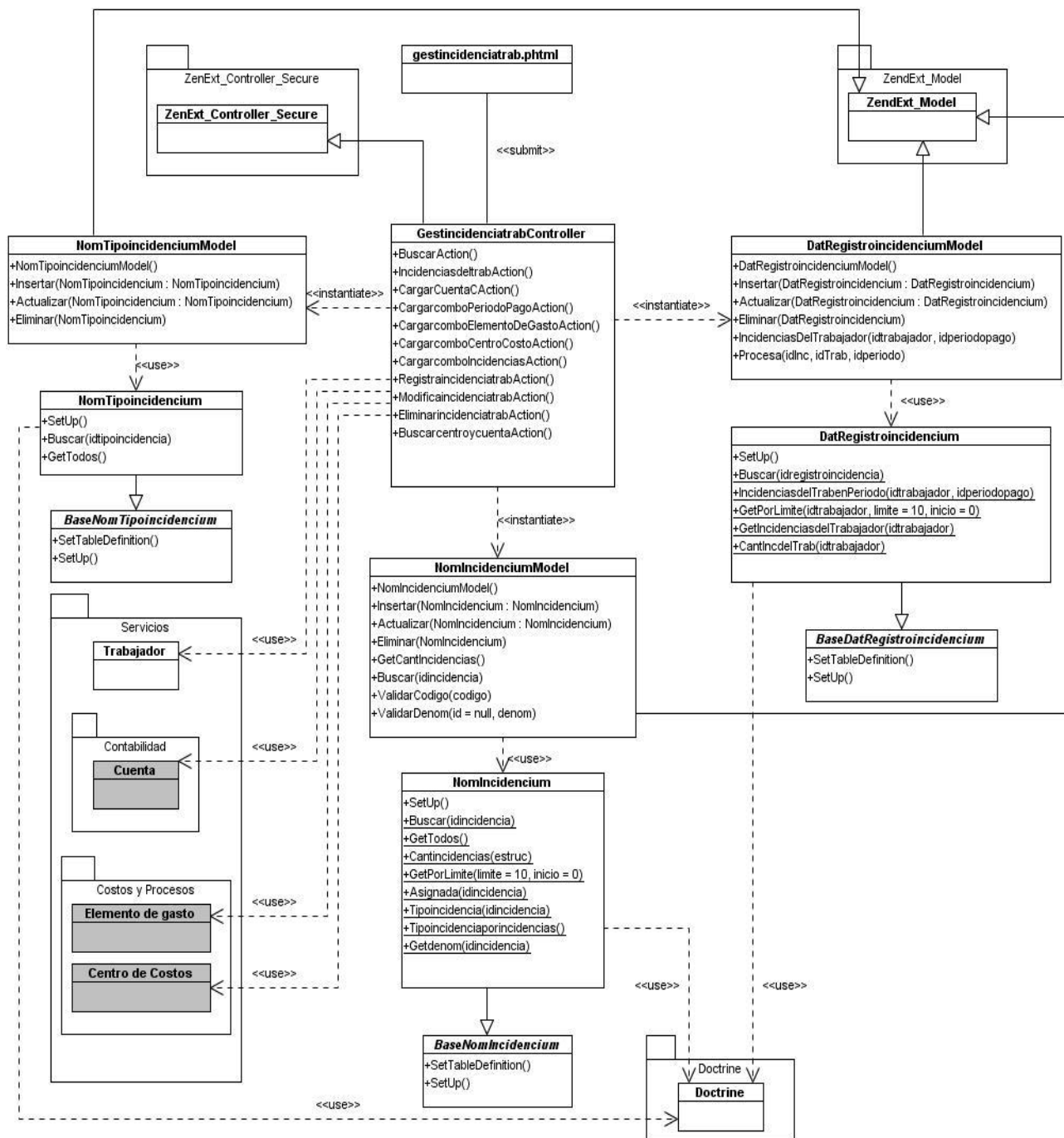


Figura 5. Diagrama de clases del diseño de Registrar Incidencias.

Asociación incidencia-impuesto.

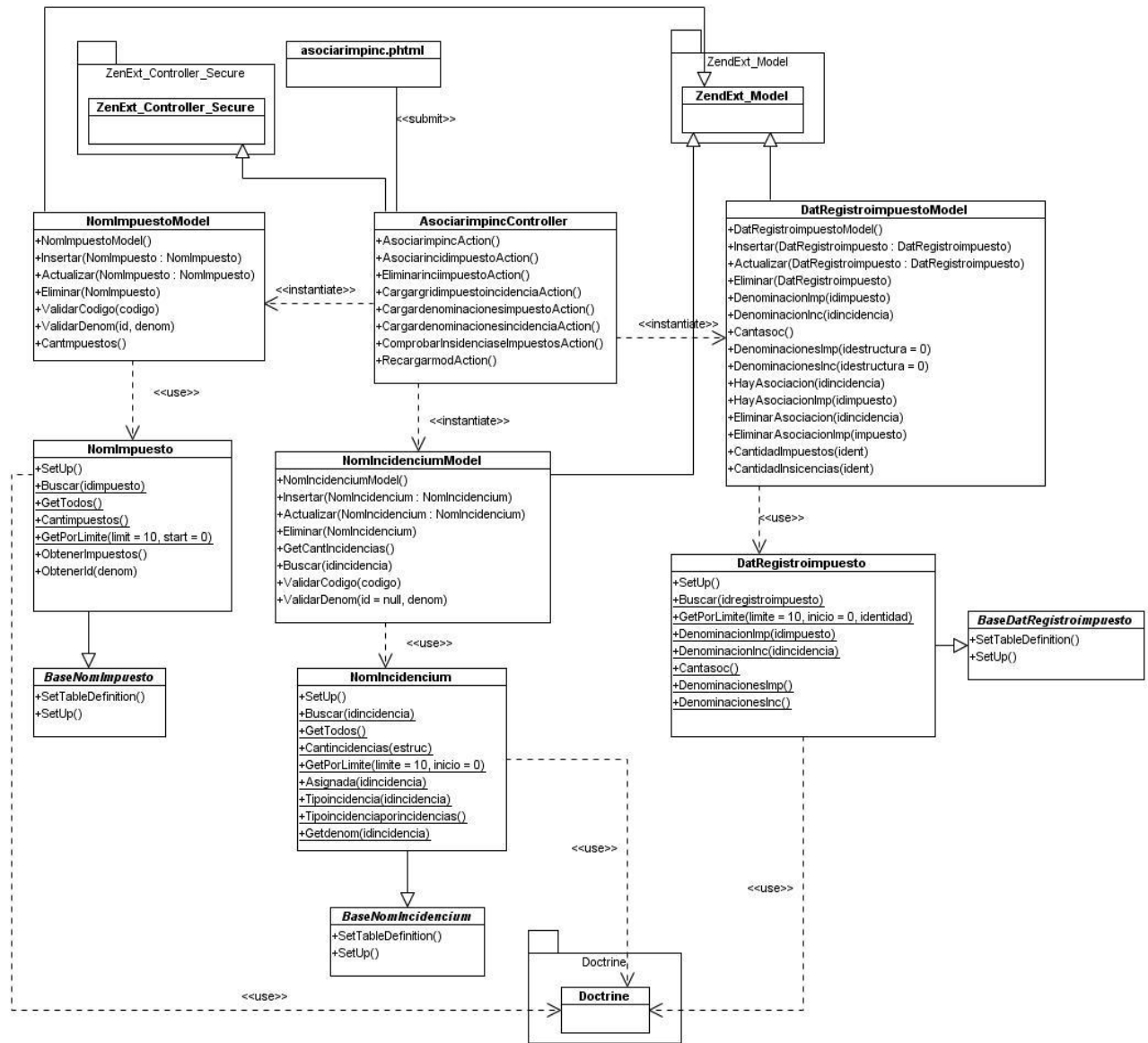


Figura 6. Diagrama de clases del diseño de Asociación incidencia-impuesto.

3.3.1 Descripciones de las clases de diseño.

Tabla 13 Descripción de la clase GestIncidenciaController.

| | |
|----------------------------------|---|
| Nombre: GestIncidenciaController | |
| Tipo de clase: Controladora | |
| Para cada responsabilidad: | |
| Nombre: | InsertarincidenciaAction() |
| Descripción: | Inserta un tipo de incidencia. |
| Nombre: | ModificarincidenciaAction() |
| Descripción: | Modifica un tipo de incidencia. |
| Nombre: | EliminarincidenciaAction() |
| Descripción: | Elimina un tipo de incidencia. |
| Nombre: | TieneasociacionAction() |
| Descripción: | Verifica si la incidencia esta afectada por algún tipo de impuesto. |
| Nombre: | CargarcomboTipoIncidenciaAction() |
| Descripción: | Carga los datos de la clasificación del tipo de incidencia. |
| Nombre: | CargarcomboTipoNominaAction() |
| Descripción: | Carga los datos del tipo de nómina. |
| Nombre: | CargarCuentasCPAction() |
| Descripción: | Carga los datos de las cuentas. |
| Nombre: | CargarElementosCPAction() |
| Descripción: | Cargar los datos de los elementos de gastos. |
| Nombre: | CargarCuentasCAAction() |
| Descripción: | Carga los datos de las cuentas. |

Tabla 14 Descripción de la clase NomIncidenciumModel.

| | |
|---------------------------------|--|
| Nombre: NomIncidenciumModel | |
| Tipo de clase: Modelo (negocio) | |
| Para cada responsabilidad: | |
| Nombre: | NomIncidenciumModel() |
| Descripción: | Constructor de la clase. |
| Nombre: | Insertar(Nomincidencium : Nomincidencium) |
| Descripción: | Insertar un tipo de incidencia. |
| Nombre: | Actualizar(Nomincidencium : Nomincidencium) |
| Descripción: | Actualizar un tipo de incidencia. |
| Nombre: | Eliminar(Nomincidencium) |
| Descripción: | Eliminar un tipo de incidencia. |
| Nombre: | GetCantIncidencias() |
| Descripción: | Obtener la cantidad de tipos de incidencias. |
| Nombre: | Buscar(idincidencia) |
| Descripción: | Buscar un tipo de incidencia. |
| Nombre: | ValidarCodigo(código) |
| Descripción: | Verificar que el código no se repita |
| Nombre: | ValidarDenom(id=null, denom) |
| Descripción: | Verificar que la denominación no se repita |

Tabla 15 Descripción de la clase Nomincidencium.

| | |
|--|--|
| Nombre: Nomincidencium | |
| Tipo de clase: Modelo (Dominio): tablas de la base de datos mapeadas por doctrine. | |
| Para cada responsabilidad: | |

| | |
|--------------|---|
| Nombre: | SetUp() |
| Descripción: | Se reconocen las relaciones con otras tablas. |
| Nombre: | Buscar(idincidencia) |
| Descripción: | Busca un tipo de incidencia. |
| Nombre: | GetTodos() |
| Descripción: | Devuelve todos los tipos de incidencia. |
| Nombre: | Cantincidencias(estruc) |
| Descripción: | Devuelve los tipos de incidencias definidos para una estructura. |
| Nombre: | GetPorLimite(limite= 10, inicio =0) |
| Descripción: | Devuelve una cantidad de tipos de incidencias. |
| Nombre: | Asignada(idincidencia) |
| Descripción: | Verifica si se le asignó a algún trabajador una incidencia de ese tipo. |
| Nombre: | Tipoincidencia(incidencia) |
| Descripción: | Devuelve un tipo de incidencia. |
| Nombre: | Tipoincidenciaporincidencias() |
| Descripción: | La clasificación de cada tipo de incidencia. |
| Nombre: | Getdenom(idincidencia) |
| Descripción: | Devuelve la denominación de un tipo de incidencia. |

Tabla 16 Descripción de la clase NomTipoincidencium.

| | |
|---|---|
| Nombre: NomTipoincidencium: se guarda la clasificación de los tipos de incidencias. | |
| Tipo de clase: Modelo (dominio) tablas de la base de datos mapeadas por doctrine. | |
| Para cada responsabilidad: | |
| Nombre: | setUp() |
| Descripción: | Se reconocen las relaciones con otras tablas. |

| | |
|--------------|---|
| Nombre: | Buscar(idincidencia) |
| Descripción: | Buscar la clasificación del tipo de incidencia. |
| Nombre: | GetTodos() |
| Descripción: | Obtener todas las clasificaciones. |

Tabla 17 Descripción de la clase GestimpuestoController.

| | |
|--------------------------------|---|
| Nombre: GestimpuestoController | |
| Tipo de clase: Controladora | |
| Para cada responsabilidad: | |
| Nombre: | CargargridimpuestoAction() |
| Descripción: | Devuelve los tipos de impuesto |
| Nombre: | CargarCuentasCAction() |
| Descripción: | Carga los datos de las cuentas. |
| Nombre: | InsertarimpuestoAction() |
| Descripción: | Inserta un tipo de impuesto. |
| Nombre: | ModificarimpuestoAction() |
| Descripción: | Modifica un tipo de impuesto. |
| Nombre: | EliminarimpuestoAction() |
| Descripción: | Elimina un tipo de impuesto. |
| Nombre: | ElemgastoAction() |
| Descripción: | Cargar datos de los elementos de gasto. |
| Nombre: | TieneasociacionAction() |
| Descripción: | Verifica si el impuesto afecta algún tipo de incidencia |

Tabla 18 Descripción de la clase NomimpuestoModel.

| | |
|---------------------------------|---|
| Nombre: NomimpuestoModel | |
| Tipo de clase: Modelo (negocio) | |
| Para cada responsabilidad: | |
| Nombre: | NomimpuestoModel () |
| Descripción: | Constructor de la clase. |
| Nombre: | Insertar(Nomimpuesto: Nomimpuesto) |
| Descripción: | Insertar un tipo de impuesto. |
| Nombre: | Actualizar(Nomimpuesto: Nomimpuesto) |
| Descripción: | Actualizar un tipo de impuesto. |
| Nombre: | Eliminar(Nomimpuesto) |
| Descripción: | Eliminar un tipo de impuesto. |
| Nombre: | ValidarCodigo(código) |
| Descripción: | Verificar que el código no se repita. |
| Nombre: | ValidarDenom(id, denom) |
| Descripción: | Verificar que la denominación no se repita. |
| Nombre: | Cantimpuestos() |
| Descripción: | Devuelve la cantidad de tipos impuestos. |

Tabla 19 Descripción de la clase Nomimpuesto.

| | |
|--|----------|
| Nombre: Nomimpuesto | |
| Tipo de clase: Model (dominio) tablas de la base de datos mapeadas por doctrine. | |
| Para cada responsabilidad: | |
| Nombre: | SetUp () |

| | |
|--------------|--|
| Descripción: | Se reconocen las relaciones con otras tablas. |
| Nombre: | Buscar(idimpuesto) |
| Descripción: | Buscar un tipo de impuesto. |
| Nombre: | GetTodos() |
| Descripción: | Devuelve todos los tipos de impuestos. |
| Nombre: | Cantimpuestos(estruc) |
| Descripción: | Devuelve los tipos de incidencias definidos para una estructura. |
| Nombre: | GetPorLimite(limite= 10, inicio =0) |
| Descripción: | Devuelve una cantidad de tipos de impuestos. |
| Nombre: | Obtenerimpuestos() |
| Descripción: | Devuelve los tipos de impuestos. |
| Nombre: | Obtenerid(id) |
| Descripción: | Devuelve el id de un impuesto que tenga una denominación dada |

Tabla 20 Descripción de la clase DatRegistroimpuestoModel.

| | |
|----------------------------------|---|
| Nombre: DatRegistroimpuestoModel | |
| Tipo de clase: Modelo (negocio) | |
| Para cada responsabilidad: | |
| Nombre: | DatRegistroimpuestoModel () |
| Descripción: | Constructor de la clase. |
| Nombre: | Insertar(DatRegistroimpuesto: DatRegistroimpuesto) |
| Descripción: | Inserta una asociación "tipo incidencia-tipo impuesto" (define que un tipo de incidencia va a ser afectada por un tipo de impuesto) |
| Nombre: | Actualizar(DatRegistroimpuesto: DatRegistroimpuesto) |

| | |
|--------------|--|
| Descripción: | Inserta un registroimpuesto (un registroimpuesto es la relación entre impuesto e incidencia) |
| Nombre: | Eliminar(DatRegistroimpuesto) |
| Descripción: | Elimina un registroimpuesto (un registroimpuesto es la relación entre impuesto e incidencia) |
| Nombre: | DenominacionImp(idimpuesto) |
| Descripción: | Devuelve la denominación de un tipo de impuesto. |
| Nombre: | DenominacionInc(idincidencia) |
| Descripción: | Devuelve la denominación de un tipo de incidencia |
| Nombre: | Cantasoc() |
| Descripción: | Devuelve la cantidad de asociaciones que existen |
| Nombre: | DenominacionesImp(idestructura=0) |
| Descripción: | Devuelve las denominaciones de los impuestos de una estructura. |
| Nombre: | DenominacionesInc(idestructura =0) |
| Descripción: | Devuelve las denominaciones de las incidencias de una estructura. |
| Nombre: | HayAsociacion(idincidencia) |
| Descripción: | Devuelve si un tipo de incidencia es afectada por algún tipo de impuesto. |
| Nombre: | HayAsociacionImp(idimpuesto) |
| Descripción: | Devuelve si un tipo de impuesto afecta por algún tipo de incidencia. |
| Nombre: | EliminarAsociacion(idincidencia) |
| Descripción: | Elimina todas las asociaciones de un tipo de incidencia. |
| Nombre: | EliminarAsociacion(impuesto) |
| Descripción: | Elimina todas las asociaciones de un tipo de impuesto. |
| Nombre: | Cantidadimpuesto(ident) |
| Descripción: | Devuelve el total de tipos de impuestos asociados. |

| | |
|--------------|---|
| Nombre: | Cantidadincidencias(ident) |
| Descripción: | Devuelve el total de tipos de incidencias asociadas |

Tabla 21 Descripción de la clase DatRegistroimpuesto.

| | |
|---|---|
| Nombre: DatRegistroimpuesto | |
| Tipo de clase: Modelo (dominio) tablas de la base de datos mapeadas por doctrine. | |
| Para cada responsabilidad: | |
| Nombre: | SetUp () |
| Descripción: | Se reconocen las relaciones con otras tablas. |
| Nombre: | Buscar(idregistroimpuesto) |
| Descripción: | Devuelve una asociación "tipo incidencia-tipo impuesto" |
| Nombre: | GetPorLimite(limite= 10, inicio =0) |
| Descripción: | Devuelve una cantidad de tipos de incidencias. |
| Nombre: | DenominacionImp(idimpuesto) |
| Descripción: | Devuelve la denominación del tipo de impuesto. |
| Nombre: | DenominacionInc(idincidencia) |
| Descripción: | Devuelve la denominación del tipo de incidencia. |
| Nombre: | Cantasoc() |
| Descripción: | Devuelve la cantidad de asociaciones. |
| Nombre: | DenominacionesImp() |
| Descripción: | Devuelve las denominaciones de los impuestos. |
| Nombre: | DenominacionesInc() |
| Descripción: | Devuelve las denominaciones de las incidencias. |

Tabla 22 Descripción de la clase GestIncidenciaTrabController.

| |
|--------------------------------------|
| Nombre: GestIncidenciaTrabController |
|--------------------------------------|

| | |
|-----------------------------|---|
| Tipo de clase: Controladora | |
| Para cada responsabilidad: | |
| Nombre: | BuscarAction() |
| Descripción: | Muestra el buscador para buscar un trabajador. |
| Nombre: | IndienciasdeltrabAction() |
| Descripción: | Devuelve las incidencias de un trabajador. |
| Nombre: | Cargar cuentaCAAction() |
| Descripción: | Cargar los datos de las cuentas. |
| Nombre: | Cargar combo Período Pago Action() |
| Descripción: | Cargar los datos del período de pago. |
| Nombre: | Cargar combo Elemento De Gasto Action() |
| Descripción: | Cargar los datos de los elementos de gasto. |
| Nombre: | Cargar combo Centro De Costo Action() |
| Descripción: | Cargar los datos de centro de costo. |
| Nombre: | Cargar combo Incidencias Action() |
| Descripción: | Cargar los datos de los tipos de incidencia. |
| Nombre: | Registrar incidencia trab Action() |
| Descripción: | Registra una incidencia a un trabajador. |
| Nombre: | Modificar incidencia trab Action() |
| Descripción: | Modifica una incidencia a un trabajador. |
| Nombre: | Eliminar incidencia trab Action() |
| Descripción: | Elimina una incidencia a un trabajador. |
| Nombre: | Buscar centro y cuenta Action() |
| Descripción: | Busca el elemento de gasto y el centro de costo del trabajador. |

Tabla 23 Descripción de la clase DatRegistroincidenciumModel.

| | |
|-------------------------------------|---|
| Nombre: DatRegistroincidenciumModel | |
| Tipo de clase: Modelo (negocio) | |
| Para cada responsabilidad: | |
| Nombre: | DatRegistroincidenciumModel() |
| Descripción: | Constructor de la clase. |
| Nombre: | Insertar(DatRegistroincidenciumModel: DatRegistroincidenciumModel) |
| Descripción: | Registra una incidencia a un trabajador. |
| Nombre: | Actualizar(DatRegistroincidenciumModel: DatRegistroincidenciumModel) |
| Descripción: | Actualiza la incidencia del trabajador. |
| Nombre: | Eliminar(DatRegistroincidenciumModel) |
| Descripción: | Elimina una incidencia a un trabajador. |
| Nombre: | IncidenciasDelTrabajador(idtrabajador,idperiodopago) |
| Descripción: | Devuelve las incidencias de un trabajador en un período de pago. |
| Nombre: | Procesa(idInc,idTrab,idperíodo) |
| Descripción: | Marca la incidencia del trabajador como que ya fue procesada en nómina. |

Tabla 24 Descripción de la clase DatRegistroincidencium.

| | |
|---|---|
| Nombre: DatRegistroincidencium | |
| Tipo de clase: Modelo (dominio) tablas de la base de datos mapeadas por doctrine. | |
| Para cada responsabilidad: | |
| Nombre: | SetUp() |
| Descripción: | Se reconocen las relaciones con otras tablas. |

| | |
|--------------|--|
| Nombre: | Buscar(idregistroincidencia) |
| Descripción: | Devuelve una incidencia de un trabajador. |
| Nombre: | IncidenciasdelTrabenPeríodo(idtrabajador,idperiodopago) |
| Descripción: | Devuelve las incidencias de un trabajador en un período de pago. |
| Nombre: | GetPorLimite(idtrabajador, limite= 10, inicio =0) |
| Descripción: | Devuelve una cantidad de incidencias del trabajador. |
| Nombre: | GetIncidenciasdelTrabajador(idtrabajador) |
| Descripción: | Devuelve las incidencias de un trabajador |
| Nombre: | CantIncdelTrab(idtrabajador) |
| Descripción: | Devuelve la cantidad de incidencias que han sido registradas a un trabajador |

Tabla 25 Descripción de la clase AsociarimpincController.

| | |
|---------------------------------|---|
| Nombre: AsociarimpincController | |
| Tipo de clase: Controladora | |
| Para cada responsabilidad: | |
| Nombre: | AsociarincidimpuestoAction() |
| Descripción: | Insertar una asociación. |
| Nombre: | EliminarincidimpuestoAction() |
| Descripción: | Eliminar una asociación. |
| Nombre: | CargardenominacionesimpuestoAction() |
| Descripción: | Devuelve las denominaciones de los tipos de impuesto. |
| Nombre: | CargardenominacionesincidenciaAction() |
| Descripción: | Devuelve las denominaciones de los tipos de incidencia. |
| Nombre: | ComprobarIncidenciasImpuestosAction() |
| Descripción: | Verificar que no se repita ninguna asociación |

Tabla 26 Descripción de otras clases.

| | |
|---------------------|---|
| Nombre de la clase: | ZenExt_Controller_Secure |
| Descripción | Controlador del framework: paquete de clases. |
| Nombre de la clase: | ZenExt_Model |
| Descripción | Modelo del framework. |
| Nombre de la clase: | Doctrine |
| Descripción | Framework de acceso a datos. |

3.4 Diagrama de componentes.

Lo que distingue a un diagrama de componentes de otros tipos de diagramas es su contenido. Normalmente contienen componentes, interfaces y relaciones entre ellos. Y como todos los diagramas, también puede contener paquetes utilizados para agrupar elementos del modelo.

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes.

A continuación el diagrama de componentes:

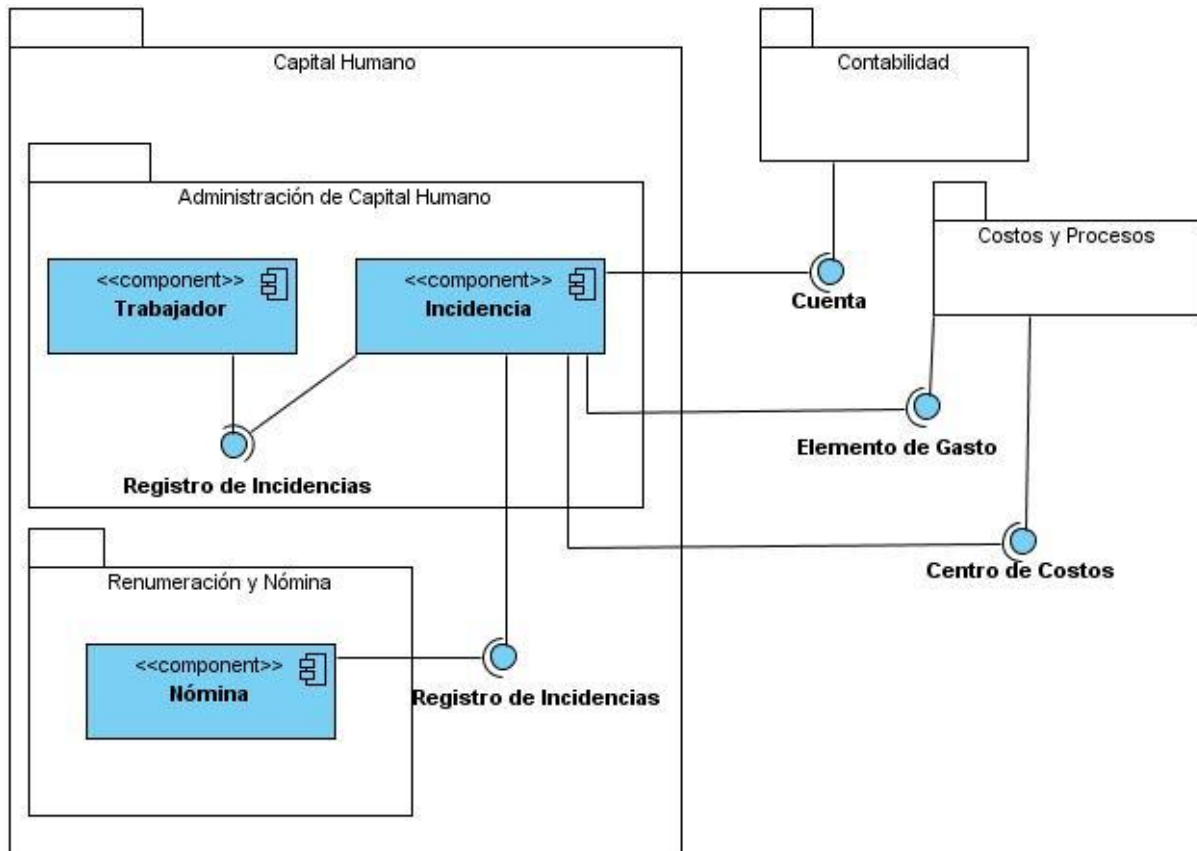


Figura 7. Diagrama de Componentes.

El diagrama representado en la figura anterior está compuesto por los componentes pertenecientes a los procesos de Administración de Capital Humano y Remuneración y Nómina que son: Trabajador, Incidencia y Nómina que se relacionan para realizar la gestión de las incidencias de los trabajadores mediante las interfaces Registro de Incidencia dentro del subsistema Capital Humano. Las interfaces que corresponden a otros subsistemas como Contabilidad y Costos y Procesos se realizan mediante Cuenta, Elemento de gasto y Centro de costos que son utilizados para gestionar las incidencias ya que las mismas deben ser contabilizadas.

3.5 Conclusiones.

Mediante el diseño realizado en este capítulo donde su principal objetivo era realizar los diagramas de clases del diseño, describir las clases y realizar el diagrama de componentes correspondiente, queda hecha la propuesta del sistema permitiendo dar paso a la realización posterior de la implementación.

Capítulo 4. Validación de los requisitos y del diseño.

4.1 Introducción.

En el presente capítulo se realizará la validación de los requisitos y el diseño, ya que de esa forma se muestra que los requisitos definen el sistema que realmente el usuario desea y que el diseño que aquí se realizó cumple con los patrones utilizados. Se utilizarán para la validación técnicas y métricas.

4.2 Pasos para la validación de requisitos.

Esta actividad tiene como propósito validar que se ha descrito lo correcto y verificar que se ha hecho correctamente.

La validación se realiza en tres pasos:

- ✓ Revisión Técnica por el equipo de Analistas Principales. El objetivo de este paso es verificar que se hayan construido correctamente los artefactos correspondientes a la Ingeniería de Requisitos.
- ✓ Revisión Funcional. Se realizará con el propósito de validar que las funcionalidades descritas satisfacen las expectativas de los interesados y que desde el punto de vista funcional se han descrito correctamente.
- ✓ Taller de Aprobación. El objetivo de este taller es presentar las soluciones a las observaciones detectadas en los pasos anteriores y aprobar la especificación de requisitos según los criterios para la evaluación y aceptación de los requisitos.

Si como resultado de este taller aún persisten observaciones y se decide aprobar los requisitos, se llegará a un acuerdo de cómo resolver cada observación. Los acuerdos se registran en la sección observaciones de la especificación de requisitos y se procede a su aprobación. Una vez aprobados los requisitos estos no pueden ser modificados, pero el equipo de desarrollo está obligado a resolver en el sistema las observaciones registradas y a actualizar la especificación de requisitos. Estos pasos se llevaron a cabo y los requisitos fueron firmados por el cliente y un conjunto de representantes de equipo de desarrollo del proyecto. (31)

4.3 Técnicas de validación de los requisitos.

Prototipos: Algunas propuestas se basan en obtener de la definición de requisitos prototipos que, sin tener la totalidad de la funcionalidad del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema.

- ✓ Un prototipo es una versión inicial de un sistema de software que se utiliza para demostrar los conceptos, probar las opciones de diseño y entender mejor el problema y su solución.
- ✓ Un prototipo puede revelar errores u omisiones en los requerimientos propuestos, favorece la comunicación entre clientes y desarrolladores, da una primera visión del producto.
- ✓ Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.

4.4 Métricas para la validación de los requisitos.

Criterios para la evaluación y aceptación de los requisitos mediante métricas.

Durante las validaciones hay que registrar las observaciones realizadas y clasificarlas según su tipo en: negocio, formato, ortografía y redacción, consistencia y otros.

De negocio: Son las observaciones realizadas porque el requisito omite o no refleja correctamente algún aspecto del negocio, por ejemplo: atributos de un concepto erróneos, omisión de validaciones y restricciones de negocio, omisión de un requisito.

Formato: Son las observaciones realizadas porque se han omitido secciones requeridas de la plantilla.

Ortografía y redacción: Son las observaciones realizadas por concepto de errores ortográficos o de redacción.

Consistencia: Son las observaciones realizadas por contradicciones existentes entre los requisitos.

Otros: Son las observaciones que no se ajustan a ninguna de las categorías anteriores.

Para aceptar una Especificación de requisitos debe cumplirse que:

La correctitud sea menor 0,10.

La completitud sea mayor que 0,90.

La consistencia sea menor que 0,20. (31) (32)

Correctitud

La métrica se propone determinar si la Especificación de requisitos contiene todos los requisitos necesarios para satisfacer las necesidades del negocio y los interesados.

$$X = D/T$$

X – Correctitud.

D – Total de observaciones de negocio realizadas en la validación.

T – Total de requisitos revisados.

Correctitud

$$X = D/T$$

$$X = 1/11$$

$$X = 0.09$$

Compleitud

La métrica pretende determinar si la Especificación de requisitos es completa.

$$X = 1 - O/S$$

X – Compleitud

O – Total de observaciones de formato.

S – Total de secciones del documento.

Compleitud

$$X = 1 - (O/S)$$

$$X = 1 - (0/6)$$

$$X = 1$$

Consistencia

La métrica se propone determinar si la Especificación de requisitos es consistente.

$$X = C/T$$

X – Consistencia

C – Total de observaciones de consistencia.

T – Total de requisitos revisados.

Consistencia

$$X = C/T$$

$$X = 2/11$$

$$X = 0.18$$

Después de obtener los resultados al aplicar las métricas a los requisitos funcionales definidos, se puede llegar a la conclusión de que como la Correctitud es menor que 0.10, la Completitud es mayor que 0.90 y la Consistencia es menor que 0.20, entonces la especificación de los requisitos puede ser aceptada satisfactoriamente, ya que tienen una buena Correctitud, Completitud y Consistencia.

4.5 Métricas para la validación del diseño.

Las métricas a utilizar en el diseño de clases para realizar la validación de que los patrones Alta Cohesión y Bajo Acoplamiento fueron bien utilizados son:

Tamaño operacional de clase (TOC): Numero de métodos. Responsabilidad, aumenta para TOC+, Complejidad de implementación, aumenta TOC+, Reutilización, disminuye TOC+.

Relaciones entre clases (bajo) (RC): Numero de relaciones de uso de una clase. Acoplamiento, aumenta RC+. Complejidad del mantenimiento, aumenta RC+. Reutilización, disminuye RC+. Cantidad de Pruebas, aumenta RC+.

Para aplicar la métrica TOC se tuvo en cuenta la cantidad de procedimientos que tenían las clases del componente Incidencias, a partir del promedio de los procedimientos y mediante un criterio se obtuvo la categoría (baja, media, alta) para la Responsabilidad, Complejidad y Reutilización.

Luego de efectuar los cálculos para un:

| | |
|-----------------------------------|--------------------|
| Total de clases | 14 |
| Promedio de procedimientos | 7.785714286 |

Tabla 27 Cantidad de procedimientos por clases.

| No | Nombre | Procedimientos |
|----|--------------------------|----------------|
| 1 | GestIncidenciaController | 9 |
| 2 | NomIncidenciumModel | 8 |
| 3 | Nomincidencium | 9 |

| | | |
|----|------------------------------|----|
| 4 | NomTipoincidencium | 3 |
| 5 | NomTipoincidenciumModel | 4 |
| 6 | GestimpuestoController | 8 |
| 7 | NomimpuestoModel | 7 |
| 8 | Nomimpuesto | 7 |
| 9 | DatRegistroimpuestoModel | 15 |
| 10 | DatRegistroimpuesto | 8 |
| 11 | GestIncidenciaTrabController | 11 |
| 12 | DatRegistroincidenciumModel | 6 |
| 13 | DatRegistroincidencium | 6 |
| 14 | AsociarimpincController | 8 |

Se obtuvieron las siguientes gráficas para la Responsabilidad, Complejidad y Reutilización, utilizando las siguientes tablas para obtener las categorías:

Tabla 28 Criterio para calcular la Responsabilidad.

| Responsabilidad | Categoría | Criterio |
|-----------------|-----------|---------------------------|
| | Baja | \leq Prom. |
| | Media | Entre Prom. y 2^* Prom. |
| | Alta | $> 2^*$ Prom. |

Tabla 29 Criterio para calcular la Complejidad de implementación.

| Complejidad implementación | Categoría | Criterio |
|----------------------------|-----------|---------------------------|
| | Baja | \leq Prom. |
| | Media | Entre Prom. y 2^* Prom. |
| | Alta | $> 2^*$ Prom. |

Tabla 30 Criterio para calcular la Reutilización.

| Reutilización | Categoría | Criterio |
|---------------|-----------|---------------------------|
| | Baja | \leq Prom. |
| | Media | Entre Prom. y 2^* Prom. |
| | Alta | \leq Prom. |

Tabla 31 Responsabilidad y Complejidad de las clases.

| Responsabilidad y Complejidad | Cantidad de clases |
|-------------------------------|--------------------|
| Baja | 10 |
| Media | 4 |
| Alta | 0 |

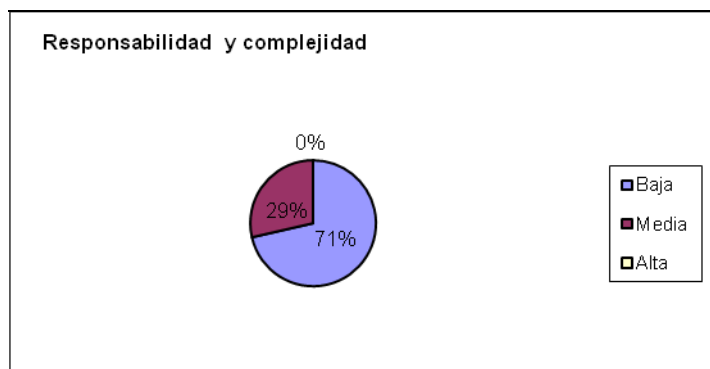


Figura 8 Representación de la responsabilidad y complejidad.

La responsabilidad y la complejidad de las clases son relativamente bajas.

Tabla 32 Reutilización de las clases.

| Reutilización | Cantidad de clases |
|---------------|--------------------|
| Baja | 0 |
| Media | 4 |
| Alta | 10 |

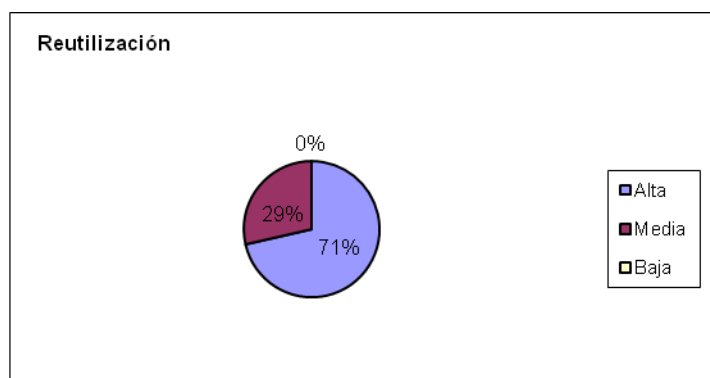


Figura 9 Representación de la reutilización.

Cuando la responsabilidad (Figura 8) de las clases entre media y baja es mayor que el 80% se puede decir que cumple con el patrón de alta cohesión, además de que presentan una reutilización alta.

Para aplicar la métrica RC se tuvo en cuenta la cantidad de relaciones que tenían las clases del componente Incidencias, a partir del promedio de las relaciones y mediante un criterio se obtuvo la categoría (baja, media, alta y ninguna en caso del Acoplamiento) y (baja, media, alta) para la Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas.

Luego de efectuar los cálculos para un:

| | |
|--|--------------------|
| Total de clases | 14 |
| Promedio de asociaciones de uso | 1.142857143 |

Tabla 33 Relaciones de uso por clases.

| No | Nombre | Procedimientos |
|-----------|------------------------------|-----------------------|
| 1 | GestIncidenciaController | 2 |
| 2 | NomIncidenciumModel | 1 |
| 3 | Nomincidencium | 0 |
| 4 | NomTipoincidencium | 0 |
| 5 | NomTipoincidenciumModel | 2 |
| 6 | GestimpuestoController | 1 |
| 7 | NomimpuestoModel | 0 |
| 8 | Nomimpuesto | 1 |
| 9 | DatRegistroimpuestoModel | 0 |
| 10 | DatRegistroimpuesto | 4 |
| 11 | GestIncidenciaTrabController | 1 |
| 12 | DatRegistroincidenciumModel | 0 |
| 13 | DatRegistroincidencium | 3 |
| 14 | AsociarimpincController | 1 |

Tabla 34 Acoplamiento de las clases.

| Acoplamiento | Cantidad de clases |
|--------------|--------------------|
| Ninguno | 5 |
| Bajo | 5 |
| Medio | 2 |
| Alto | 2 |

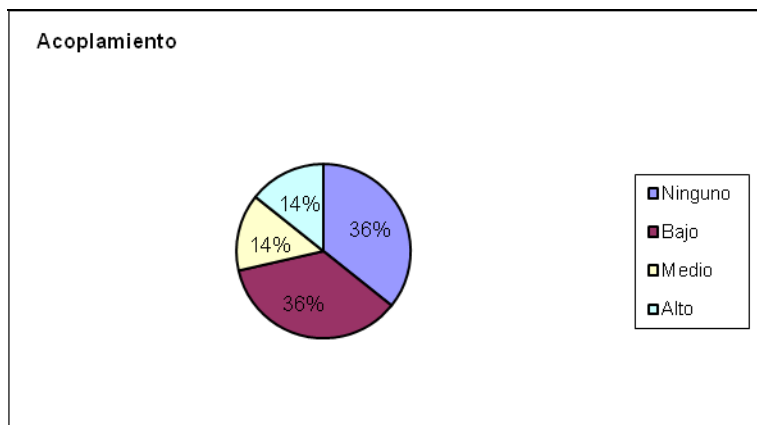


Figura 10 Representación del acoplamiento.

Para la reutilización.

Tabla 35 Reutilización de las clases.

| Reutilización | Cantidad de clases |
|---------------|--------------------|
| Baja | 1 |
| Medio | 3 |
| Alto | 10 |

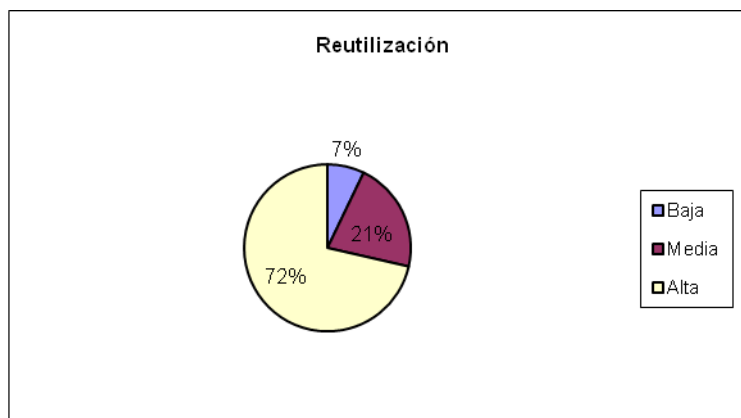


Figura 11 Representación de la reutilización.

De manera general los resultados de esta métrica son positivos. El acoplamiento existente entre las clases es bajo, el 39% no tiene ningún acoplamiento, el 39% es bajo, el 15% es medio, y solo el 15% tiene alto acoplamiento. El nivel de reutilización de las clases es bastante bueno, el 69% de las clases pueden ser reutilizadas.

4.6 Conclusiones.

La validación de los requerimientos es una actividad muy importante, pues un levantamiento de requerimientos con errores que no se detecten a tiempo, además de conducir a resultados inesperados provoca costos excesivos y gran pérdida de tiempo.

Mediante las métricas utilizadas para la validación de requisitos se pudo verificar que los requisitos tienen la calidad requerida para su aprobación ya que la correctitud, completitud y consistencia arrojaron los resultados esperados para su aceptación.

Mediante las métricas utilizadas para el diseño se pudo verificar que el diseño de las clases cumple con los patrones de bajo acoplamiento y alta cohesión, arrojando el TOC que las clases contienen baja responsabilidad y complejidad y alta reutilización, mientras que el RC demostró la existencia de un acoplamiento bajo y una alta reutilización de las clases.

CONCLUSIONES

Después de realizada la investigación se arribaron a las siguientes conclusiones:

- ✓ Se arrojó a la necesidad de crear un nuevo sistema para la gestión de incidencia luego de un minucioso estudio de los sistemas existentes.
- ✓ Mediante la identificación de los procesos de negocio y la definición de los requisitos funcionales se realizó la descripción de un sistema que gestiona las incidencias de los trabajadores.
- ✓ Se realizó el diseño del sistema utilizando patrones logrando un mejor funcionamiento del sistema propuesto.
- ✓ Se evaluó el análisis y diseño del sistema aplicando métricas que demostraron que los requisitos funcionales y el diseño de clases no presentan dificultades.
- ✓ Se le da cumplimiento a los objetivos propuestos.

RECOMENDACIONES

En este trabajo se realizó una primera versión que brindó solución a los problemas referentes a la gestión de incidencias de los trabajadores. Por lo que se recomienda:

- ✓ Realizar la implementación del componente propuesto.
- ✓ Continuar el estudio del tema de las incidencias y se le incluyan los indicadores necesarios para la realización del análisis del fondo de tiempo laboral y el ausentismo.
- ✓ Realizar el despliegue del componente propuesto como parte del subsistema de Capital Humano del sistema Cedrux.

BIBLIOGRAFÍA

1. **MINISTERIO DE FINANZAS Y PRECIOS.** *RESOLUCION No.13.* Habana : pág 9, 2007.
2. **Ministerio de Finanzas y Precios.** *RESOLUCIÓN No.13.* Habana : pág 6, 2007.
3. **Ministerio de Trabajo y Seguridad Social.** *RESOLUCIÓN No. 39.* Habana : s.n., 1989.
4. **Ministerio de Finanzas y Precios.** *Ley 73.* Habana : s.n., 1994.
5. **Aqua eSolutions.** Aqua eSolutions. [En línea] [Citado el: 3 de 12 de 2008.] http://www.aquaesolutions.com/sb_business_entities.htm.
6. **Assets.** assets. [En línea] 2004. [Citado el: 05 de 12 de 2008.] <http://assets.co.cu/assets.asp>.
7. **Assets NS.** *Manual de Usuario Módulo de Recursos Humanos Assets.* 2008. pág. 1.
8. *El VERSAT-Sarasola: Sistema cubano de Gestión Contable-Financiero.* **Cobo Morales, Lic.Pedro H. y Sosa Porteiro, MSc.Marisel.** Habana : s.n., 2008, La Revista del empresario cubano.
9. *Un efectivo sistema cubano de contabilidad.* **Luis Martín, Eduardo.** Habana : Edición Online el Economista de Cuba, 2005.
10. **Zito, Míriam.** radiorebelde. [En línea] 18 de Abril de 2008. [Citado el: 7 de Diciembre de 2008.] http://www.infosoc.cu/noticias_busca.php?fichero=561.
11. **Grau Merconchini, David.** juventudrebelde. [En línea] 14 de Febrero de 2008. [Citado el: 8 de Enero de 2009.] <http://www.juventudrebelde.cu/cuba/2008-02-14/software-libre-ii-una-estrategia-decisiva-de-desarrollo/>.
12. —. juventudrebelde. [En línea] 14 de Febrero de 2008. [Citado el: 10 de Enero de 2009.] <http://www.juventudrebelde.cu/cuba/2008-02-14/software-libre-ii-una-estrategia-decisiva-de-desarrollo/>.
13. **Angel Álvarez, Miguel.** DesarrolloWeb. [En línea] [Citado el: 11 de Enero de 2009.] <http://www.desarrolloweb.com/articulos/25.php>.
14. **Correa, Leonardo.** NovaWeb. [En línea] [Citado el: 11 de Enero de 2009.] <http://www.webnova.com.ar/articulo.php?recurso=357>.
15. **SlideShare Inc.** . SlideShare. [En línea] [Citado el: 12 de Enero de 2009.] <http://www.slideshare.net/almarag/ext-js-y-frameworks-javascript>.
16. **Alberto.** La Terminal. [En línea] 10 de Noviembre de 2007. [Citado el: 12 de Enero de 2009.] <http://laterminal.wordpress.com/2007/11/10/ext-js-otro-framework-para-javascript/>.
17. **Rodas Hinostroza, Raul.** LinuxCentro. [En línea] 2005. [Citado el: 13 de Enero de 2009.] <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.

18. **Gonzalez, Carlos.** UsabilidadWeb. [En línea] [Citado el: 16 de Enero de 2009.]
<http://www.usabilidadweb.com.ar/postgre.php>.
19. **Centro de desarrollo de soluciones empresariales (CEDRUX).** 2008.
20. **Jacobson, Ivar, Boch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* 2000. pág. 13.
21. **Ruiz, Francisco.** Facultad de Ciencias Económicas. [En línea] [Citado el: 14 de Enero de 2009.]
http://www.economicasunp.edu.ar/02-EGrado/materias/trelew/analisis_sistemas%20II/info/Tecnologia%20para%20la%20Gestion%20de%20Procesos%20de%20Negocio.pdf.
22. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *El Lenguaje Modificado de Modelado.* 2000. pág. XIX.
23. **García, Joaquín.** IngenieroSoftware. [En línea] 2005. [Citado el: 15 de Enero de 2009.]
<http://www.ingenierosoftware.com/analisisydiseno/uml.php>.
24. **Ferré Grau, Xavier y Sánchez Segura, María Isabel.** elquintero. [En línea] [Citado el: 15 de Enero de 2009.] <http://www.elquintero.net/Manuales/UML/umlTotal.pdf>.
25. **Murillo Alfaro, Félix.** [En línea] [Citado el: 19 de Enero de 2009.]
<http://www1.inei.gov.pe/biblioineipub/bancopub/Inf/Lib5103/Libro.pdf>.
26. **AnunciosGoogle.** Visual Paradigm for UML. [En línea] 2007.
[http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).
27. **Autores, Colectivo de.** Teleformación. [En línea] [Citado el: 19 de Enero de 2009.]
<http://teleformacion.uci.cu/mod/resource/view.php?id=11553>.
28. **Arias Chaves, Michael.** intersedes. [En línea] 26 de Junio de 2007. [Citado el: 20 de Enero de 2009.] http://www.intersedes.ucr.ac.cr/pdfs_10/10-art_11.pdf.
29. **Larman, Craig.** *UML y Patrones.* 1999.
30. *Arquitectura y Patrones de diseño.* **Colectivo de autores.** 2008-2009.
31. **Miniet, Yanet Vega.** *Procedimiento de gestión de requisitos.* 2008.
32. **IEEE.** *ESPECIFICACIONES DE LOS REQUISITOS .* 1998.

GLOSARIO DE TÉRMINOS

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

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

Nóminas: La nómina es un documento interno en el cual se registra en forma detallada la liquidación del salario de cada empleado de una empresa, de acuerdo con la legislación laboral vigente y los acuerdos pactados en las convenciones colectivas de trabajo.

Sistema gestor de base de datos: Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.

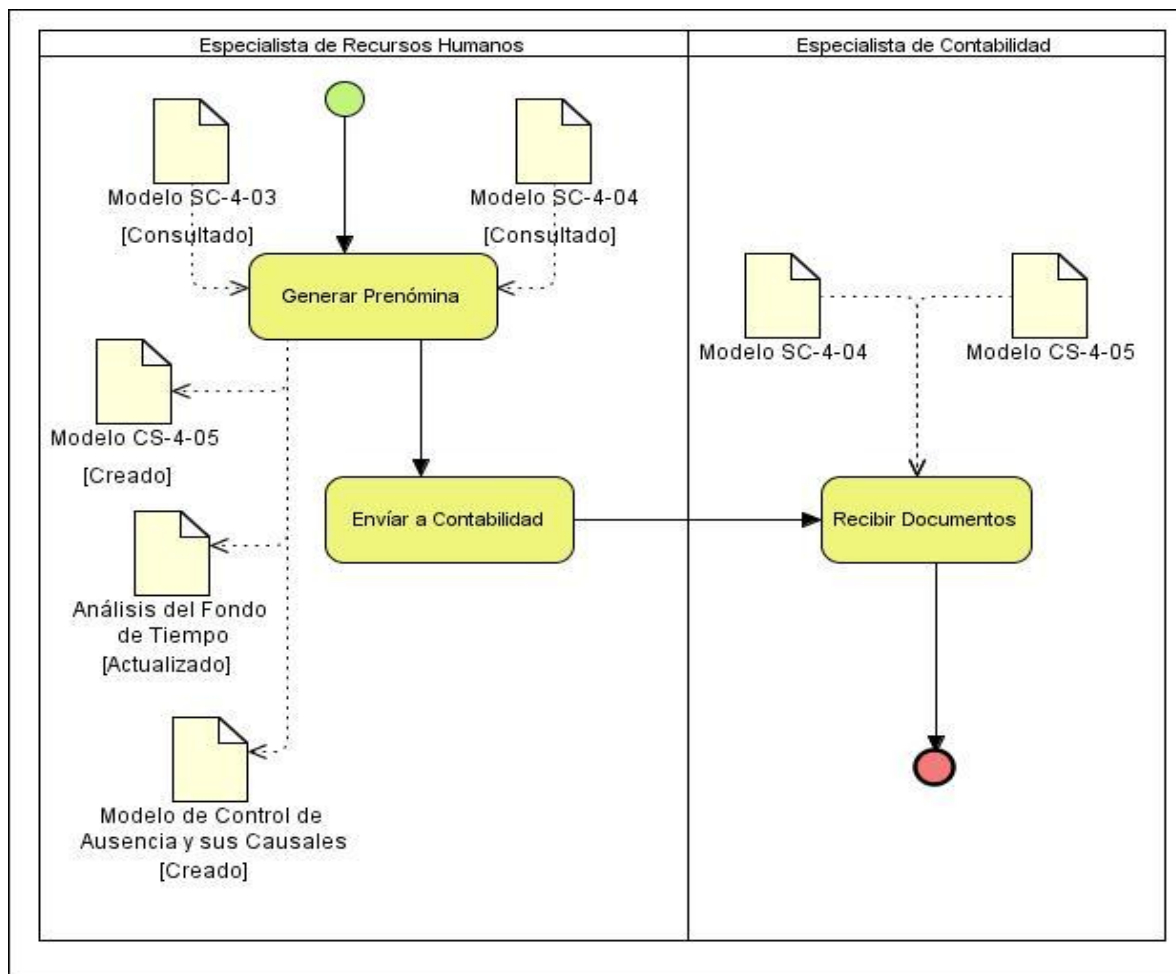
SQL (Structured Query Language): Conjunto estándar de comandos para gestionar bases de datos relacionales por sus mismas características relacionales.

UML (Unified Modeling Language): Lenguaje gráfico que brinda un vocabulario y reglas para especificar, construir, visualizar y documentar los artefactos de un sistema utilizando el enfoque orientado a objetos.

BPMN (Business Process Management Notation): Es una notación que modela los procesos de negocio, basada en diagramas de flujo fácil de entender.

ANEXOS

Anexo 1. Diagrama de procesos del negocio.



Anexo 2. Reporte de tipos de incidencias.

Listado de Incidencias de la entidad.

Nombre de la Entidad: _____ **Fecha:** _____

| Código | Denominación | Por ciento | Vacaciones | | Solapado | Comportamiento | Clasificación | Tipo de nómina |
|--------|--------------|------------|------------|---------|----------|----------------|---------------|----------------|
| | | | Tiempo | Importe | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

Anexo 3. Reporte de tipos de impuestos.

Listado de Impuestos de la entidad.

Nombre de la Entidad: _____ **Fecha:** _____

| Código | Denominación | Por Ciento | Impuesto de tipo Salarial. |
|--------|--------------|------------|----------------------------|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Anexo 4. Reporte Pre-nómina.

Pre-nómina.

Nombre de la Entidad: _____

Código de la entidad: _____

Área de Trabajo: _____

| Periodo de Pago | Expediente Laboral | Nombre y Apellidos | Tipo de Incidencia | Cantidad de Horas | Centro de Costo | Fecha Inicio | Fecha Fin |
|-----------------|--------------------|--------------------|--------------------|-------------------|-----------------|--------------|-----------|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

Elaborado por: _____ Aprobado por: _____

Firma

Firma

Fecha: __/__/__

Fecha: __/__/__

Anexo 5. Reporte tipos de subsidios.

Reporte de Subsidios.

Nombre de la Entidad: _____

Código de la entidad: _____

Área de Trabajo: _____

| Periodo de Pago | Expediente Laboral | Nombre y Apellidos | Tipo de Subsidio | Cantidad de Horas | Importe | Centro de Costo | Fecha Inicio | Fecha Fin |
|-----------------|--------------------|--------------------|------------------|-------------------|---------|-----------------|--------------|-----------|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

Elaborado por: _____ Aprobado por: _____

Firma

Firma

Fecha: __/__/__

Fecha: __/__/__

Anexo 6. Prototipo de interfaz tipo de incidencia.

Tipo de Incidencia

Incendencia

Código Denominación Por ciento

Tipo de Nómina Clasificación Comportamiento

Vacaciones

Acumula días de vacaciones Acumula importe vacaciones

Sub-elemento Cuenta crédito vacaciones

Débitos

Elemento de Gasto

Otra cuenta

Cuenta crédito de incidencia Solapa

Cancelar Aplicar Aceptar

Anexo 7. Prototipo de interfaz tipo de impuesto.

Tipo de Impuesto

Impuesto

Código Denominación Por ciento

Cuenta de crédito Impuesto salarial

Débitos

Elemento de gasto

Otra cuenta

Cancelar Aplicar Aceptar

Anexo 8. Prototipo de interfaz asociación incidencia-impuesto.

Asociar impuesto a incidencia

Asociar impuesto a incidencia

Incidencia: ▼

Impuesto: ▼

Cancelar Aplicar Aceptar

Anexo 9. Prototipo de interfaz registrar incidencia.

Registrar Incidencia

Registrar incidencias al trabajador

Buscar trabajador

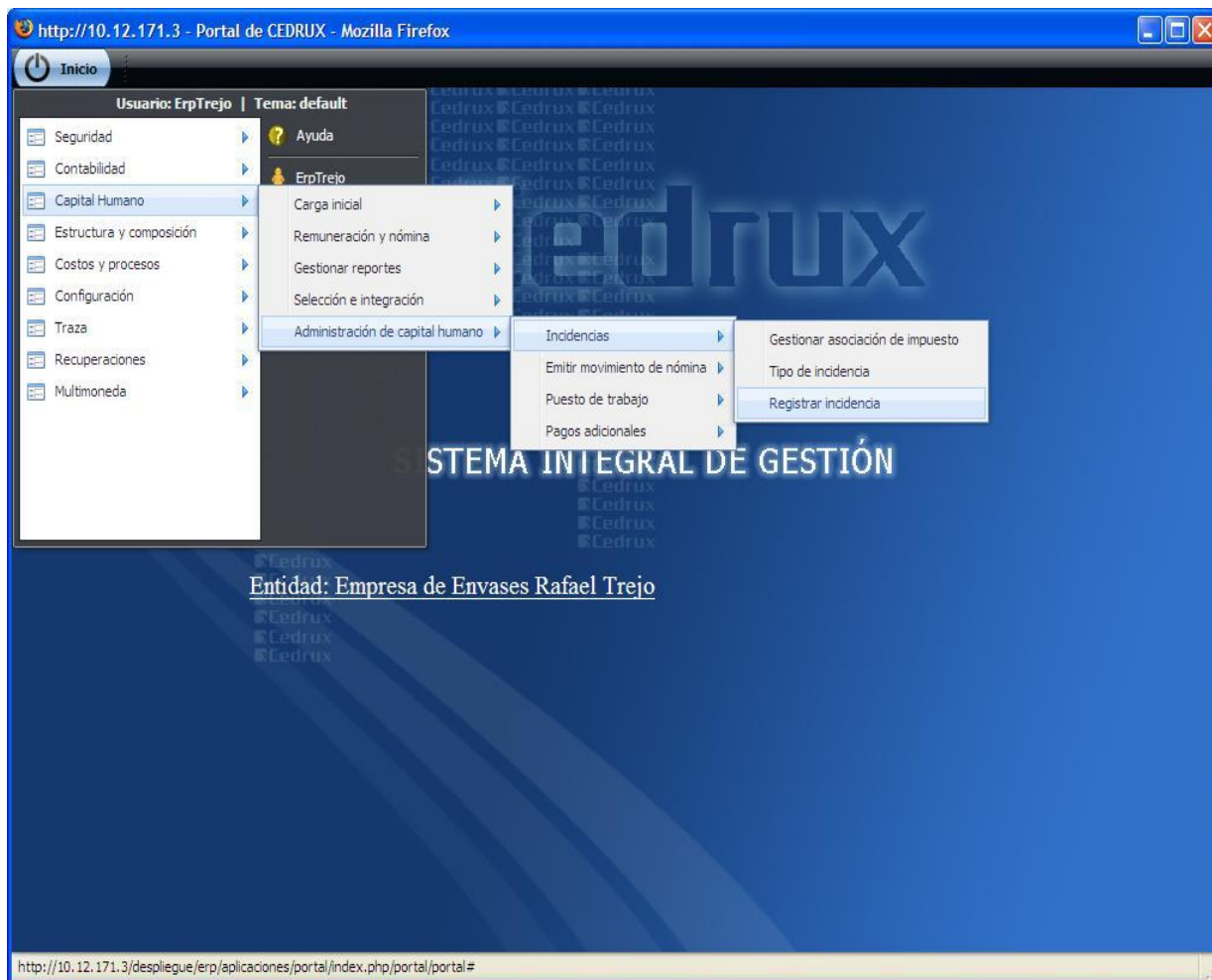
CI: Nombre (s): Primer Apellido: Segundo Apellido:

Incidencias del trabajador

| | | | |
|---------------------------------|---------------------------------|---------------------------------|----------------------|
| Tipo de incidencia | Fecha Inicio | Fecha Fin | Horas |
| <input type="text" value=""/> ▼ | <input type="text" value=""/> ▼ | <input type="text" value=""/> ▼ | <input type="text"/> |
| Elemento de gasto | Centro de Costo | Período de pago | Importe |
| <input type="text" value=""/> ▼ | <input type="text" value=""/> ▼ | <input type="text" value=""/> ▼ | <input type="text"/> |
| Otra cuenta | <input type="text" value=""/> ▼ | | |

Cancelar Aplicar Aceptar

Anexo 10. Navegabilidad en la aplicación para registrar las incidencias.



Anexo 11. Registrar las incidencias.

http://10.12.171.3 - Portal de CEDRUX - Mozilla Firefox

Inicio Registrar...

Registrar incidencia

Registrar incidencias al trabajador

Buscar trabajador

Incidencias del trabajador

No. carnet: 84100401053 Nombre: Ariagna P. apellido: Gonzalez S. apellido: landeiro + Adicionar - Modificar - Eliminar

| Tipo de incidencia | Período de pago | Fecha inicio | Fecha fin | Importe | Horas | Otra cuenta | Centro de costo |
|------------------------|------------------------|--------------|------------|---------|-------|-------------------------|------------------------|
| Llegada tarde | Mayo 2008 1ra quincena | 2009-05-08 | 2009-05-08 | 4.00 | 2.00 | Retenciones 1.5% Seguri | Gestion Capital Humano |
| Ausencia Injustificada | Mayo 2008 1ra quincena | 2009-05-09 | 2009-05-09 | 12.00 | 8.00 | | Gestion Capital Humano |
| Interrupción Laborales | Mayo 2008 1ra quincena | 2009-05-06 | 2009-05-08 | 18.00 | 24.00 | | Gestion Capital Humano |

Trabajador

No. carnet: 84100401053 Nombre: Ariagna Primer apellido: Gonzalez Segundo apellido: landeiro

Tipo de incidencia: 101 --> Interrupció Fecha inicio: 06/05/2009 Fecha fin: 08/05/2009 Horas: 24

Otra cuenta: [Seleccionar] Centro de costo: Gestion Capital Hur Período de pago: Mayo 2008 1ra qui Importe: 18

Elementos del Gasto: Gastos de sistemas

Cancelar Aceptar

Página 1 de 1 Mostrando 1 - 3 de 3

Done