

Universidad de las Ciencias Informáticas



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

TÍTULO

*Desarrollo de los componentes Puesto de Trabajo y Pagos Adicionales del
subsistema Capital Humano integrado al sistema integral de gestión
CEDRUX.*

Autores:

Héctor Prada Nicot

Kenner Sánchez González

Tutor: *Ing. Javier Heredia Ruiz*

La Habana. junio, 2009

“Año 50 de la Revolución”

“Si no existe la organización, las ideas, después del primer momento de impulso, van perdiendo eficacia.”

Ernesto Guevara de la Serna

DATOS DEL CONTACTO

Datos del Contacto

Nombre y Apellidos: Ing. Javier Heredia Ruiz.

Fecha de nacimiento: 5 de noviembre de 1983.

País: Cuba.

Ciudadanía: cubana.

Carnet de Identidad: 83110524587.

Correo: jheredia@uci.cu

Situación laboral: Profesor Adiestrado, Dirección de Producción 4.

Institución: Universidad de las Ciencias Informáticas (UCI).

Dirección: Carretera San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba, Código postal 19370.

Currículo:

Ingeniero en Ciencias Informáticas de la Universidad de las Ciencias Informáticas desde julio del 2007. Al graduarse pasa a formar parte del claustro de profesores de la UCI, en la Disciplina de Práctica Profesional impartiendo en la Facultad 8 el Segundo Perfil de Calidad del Software durante el primer semestre, en el segundo semestre pasa a formar parte del claustro de la facultad 10 en el departamento de Sistemas Digitales en la asignatura de Seguridad Informática, facultad en la cual se integró al proyecto con el Consejo de Estado, específicamente en el de las Casas de Protocolo, en el segundo año de adiestramiento se incorpora al CESGE específicamente en el Proyecto ERP-Cuba, en la Línea de Capital Humano como especialista de Calidad del Software y luego a la Subdirección de Calidad del centro, durante el período que ha formado parte del proyecto impartió las asignaturas de Ingeniería de Software I y II para los estudiantes de 5to año y Comercio Electrónico para los de 4to año . En el tiempo que lleva como graduado ha ejercido como miembro de tribunales de diferentes eventos desarrollados en la universidad, de tesis. Se encuentra cursando el Diplomado de Software Libre el cual le tributará para la maestría.

AGRADECIMIENTOS

Agradecemos a nuestro tutor por el apoyo incondicional que siempre nos brindó, a nuestros compañeros y compañeras del proyecto, a Chenchó y a todas aquellas personas que de una u otra forma hicieron posible la realización de este trabajo.

A nuestra familia que es lo más grande que tenemos, por la confianza que siempre han depositado en nosotros y por formar parte de este sueño que ya podemos compartir con ellos.

DEDICATORIA

A todas aquellas personas maravillosas que de una forma u otra contribuyeron para que hoy este aquí, aquellas personas de las que aprendí mucho, y me hicieron reflexionar y aprender que el sentido de la vida es sentirse responsable de aquellos a quien uno ama. A todos mis compañeros de aulas, proyecto y amigos que siempre estuvieron presente en cada momento de alegría y tristeza. A toda mi familia, especialmente a mi mamá quien siempre ha sido mi principal inspiración y guía, ella supo darme la confianza y ser la mejor de las madres, me enseñó a no detenerme ante las adversidades de la vida, a salir adelante, a confiar en mi persona y demostrarme que todo es posible cuando se desea y quiere.

Kenner Sánchez González.

Dedico este trabajo a toda mi familia y especialmente a mi madre, mi padre y mi hermano. A todas aquellas personas que de una u otra forma han contribuido a que este aquí en este momento, que me han soportado y me han sabido valorar en la buenas y en las malas, a todos mi amigos y a mi tutor; y de una forma súper especial a mi chiqui que no sabe cuan imprescindible se ha vuelto para mí.

Héctor Prada Nicot.

RESUMEN

Los procesos de gestión de Puestos de Trabajo y Pagos Adicionales en las entidades presupuestadas y empresariales cubanas no engloban una solución integral. Esto provoca demora en los servicios, pérdida de la información y que los reportes estadísticos no sean los más confiables en las entidades donde la gestión de estos procesos se lleva a cabo de forma manual o semiautomática. Para darle solución al problema anterior, se desarrolló el presente trabajo teniendo como objetivo fundamental desarrollar de los componentes Pagos Adicionales y Puesto de Trabajo del subsistema Capital Humano del sistema integral de gestión CEDRUX para lograr una gestión integral de los Pagos Adicionales y los Puestos de Trabajo en las entidades presupuestadas y empresariales cubanas.

Para el desarrollo de los componentes se utilizó como lenguaje de programación del lado del servidor PHP en su versión 5.2 y del lado del cliente Java Script aplicándose ambos mediante el IDE de desarrollo Zend Studio para Eclipse en su versión 6.0. La especificación, construcción y documentación de la solución se realizó basándose en el Modelo de Desarrollo Orientado a Componentes (MDOC) haciéndose uso del Lenguaje Unificado de Modelado (UML) para el diseño y la Notación para el Modelado de Procesos de Negocio (BPMN) aplicándose ambos a través la herramienta de modelado Visual Paradigm for UML 6.1 Enterprise Edition.

Como resultado se obtuvo una solución capaz de gestionar la información de forma correcta que a diferencia de otros sistemas permite una gestión integral con funcionalidades generales de los procesos y las particularidades de la economía cubana.

Palabras Claves: Pagos Adicionales, Puestos de Trabajo, procesos de gestión, entidades presupuestadas y empresariales, componentes.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	4
Introducción	4
1.1 Gestión de Recursos Humanos.....	4
1.2 Funciones de la Gestión de Recursos Humanos.....	4
1.3 Puestos de Trabajo	5
1.4 Pagos Adicionales.....	9
1.5 Análisis de soluciones existentes	10
1.6 Modelo de desarrollo adoptado	12
1.7 Herramientas y tecnologías.....	13
1.7.1 Herramientas CASE	14
1.7.1.1 Visual Paradigm	14
1.7.2 Lenguajes de Modelado	14
1.7.2.1 UML	15
1.7.2.2 BPMN.....	15
1.7.3 Servidor Web	16
1.7.3.1 WAMP 5.....	16
1.7.4 Lenguajes de programación	17
1.7.4.1 JavaScript (tecnología del lado del cliente)	17
1.7.4.2 AJAX (tecnología del lado del cliente)	17
1.7.4.3 PHP (tecnología del lado del servidor)	19
1.7.5 Ambiente de desarrollo integrado (IDE).....	19
1.7.5.1 Zend Studio para Eclipse 6.0	20
1.7.6 Frameworks	20
1.7.6.1 Zend Framework	21
1.7.6.2 Zend_Ext Framework	21
1.7.6.3 Doctrine Framework.....	22
1.7.7 Sistema Gestor y herramienta de Base de Datos.....	22
1.7.7.1 PostgreSQL.....	22
1.7.7.2 EMS PostgreSQL Manager 3	23

1.7.8 Control de versiones	23
1.7.8.1 TortoiseSVN.....	24
1.7.9 Herramienta de apoyo.....	24
1.7.9.1 Navegador	24
Conclusiones	25
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN.....	26
Introducción	26
2.1 Caso de estudio	26
2.2 Propuesta de solución.....	26
2.3 Descripción de los procesos de negocio	26
2.3.1 Gestión de Puestos de Trabajo	27
2.3.2 Gestión de Pagos Adicionales.....	27
2.4 Mapa de procesos del negocio.....	29
2.5 Requisitos Funcionales	29
2.5.1 Especificación de los requisitos funcionales	30
2.6 Requisitos no Funcionales	47
2.7 Modelo conceptual	50
2.8 Modelo de diseño.....	51
2.8.1 Patrones de Diseño.....	51
2.8.1.1 Patrón Arquitectónico Modelo-Vista-Controlador (MVC).....	52
2.8.1.2 Patrones GRASP	54
2.8.1.3 Patrones de comportamiento	55
2.8.1.4 Patrones estructurales	56
2.8.2 Diagramas de clases del diseño.....	57
2.9 Diseño de la Base de Datos	61
2.9.1 Modelo de datos.....	61
2.10 Evaluación del modelo de diseño propuesto	61
Conclusiones	70
CAPÍTULO 3 IMPLEMENTACIÓN Y PRUEBA.....	71
Introducción	71
3.1 Modelo de implementación	71
3.2 Diagrama de componentes	71

3.3 Modelo de prueba	72
3.3.1 Descripción de las pruebas	73
3.3.2 Aplicación de pruebas de caja blanca	73
Conclusiones	82
CONCLUSIONES	83
RECOMENDACIONES	84
REFERENCIAS BIBLIOGRÁFICAS	85
BIBLIOGRAFÍA	87
GLOSARIO DE TÉRMINOS	89
ANEXOS	93
Anexo 1 Modelo de datos	93
Anexo 2 Descripción de las tablas de la Base de Datos.....	93
Anexo 3 Instrumento de medición de la métrica Tamaño operacional de clase (TOC).....	99
Anexo 4 Instrumento de medición de la métrica Relaciones entre clases (RC).....	101

INTRODUCCIÓN

Hoy en día los sistemas empresariales están en constante evolución y desarrollo, a raíz de esto la competencia entre ellos se hace cada vez más fuerte. Para poder triunfar en el mercado las empresas necesitan gestionar con mayor eficiencia los procesos que se llevan a cabo en ellas, entre estos procesos se encuentran la gestión de la cadena de aprovisionamiento y fabricación, proyectos, finanzas, gestión de Recursos Humanos, marketing, ventas y procesos de servicio.

“Si se quiere conocer el grado de modernización de la administración de un país, el mejor indicador de que se dispone es el de los Recursos Humanos y, más concretamente, el tránsito de la tradicional administración de personal a la gestión estratégica del capital humano” (1)

El área de Recursos Humanos (RH) es considerada dentro de las más importantes en una empresa, es en ella donde se tiene como propósito garantizar la eficiencia y la eficacia de la participación del personal en el éxito empresarial. A los Recursos Humanos se les conoce también como Capital Humano (CH) indicando con ello el gran aporte humano y su aprovechamiento, al que como activo intangible se le asigna un valor y se considera como capital. Como parte de la administración de RH se tienen en cuenta por lo menos los siguientes elementos: reclutamiento del personal, la selección de personal, la descripción y el análisis de Puestos de Trabajo, la evaluación del desempeño de personal, los sistemas de compensación, el plan de beneficio social, la higiene y seguridad en el trabajo y la capacitación y desarrollo de personal.

Formando parte de los procesos que se llevan a cabo dentro de la compleja y amplia área de RH se encuentran la gestión de los Pagos Adicionales y Puestos de Trabajo. Los Pagos Adicionales no son más que los pagos que se les ingresan a los trabajadores por laborar en determinadas condiciones o Puestos de Trabajo y otros factores extracalificatorios, siendo así los Puestos de Trabajo los cargos u ocupaciones referentes a las actividades que realizan los individuos en las entidades. Los Pagos Adicionales son conocidos en el mundo como complementos salariales, bonificaciones o pluses salariales. Además de ser asociados por características propias del trabajador estos pagos son asociados por características propias del Puesto de Trabajo en función de circunstancias relativas al trabajo realizado.

INTRODUCCIÓN

Como parte del amplio proceso de automatización de la economía cubana que se viene desarrollando en todo el país y la importancia de contribuir con la independencia tecnológica, ante los enormes retos que impone el mundo actual y las dificultades económicas en que se encuentra Cuba a raíz del bloqueo de los EU, la necesidad de identificar y explotar las reservas de productividad del trabajo para poder satisfacer las exigencias requeridas en la legislación laboral vigente se hace inminente. Para contribuir con este proceso de automatización específicamente en las entidades presupuestadas y empresariales cubanas se impone la necesidad de gestionar los Puestos de Trabajo y los Pagos Adicionales de una forma integral como un apoyo a la gestión integral de recursos Humanos que permita la adaptabilidad al cambiante entorno económico en el que vivimos. Los procesos de gestión de los Pagos Adicionales y Puestos de Trabajo en gran parte de las entidades presupuestadas y empresariales del gobierno cubano se lleva a cabo de forma manual y semiautomática, archivando la información en formato duro (papel), esto provoca pérdida de información, duplicación de la misma y que los reportes estadísticos no sean los más confiables; de esta forma el desarrollo de estos procesos se torna engorroso y complejo, trayendo consigo el mal funcionamiento de las entidades en general, sin dejar de mencionar que los RH han alcanzado un gran desarrollo y se hace cada vez más compleja su administración. Existen en Cuba sistemas certificados que de una u otra forma alcanzan a gestionar parcialmente los Puestos de Trabajo y los Pagos Adicionales, los mismos están enfocados hacia objetivos muy específicos y ninguno de ellos logra una gestión integral que permita realizar un análisis detallado de la amplia gama de aspectos que caracterizan estos procesos en un entorno laboral.

De lo antes expuesto se define el siguiente **problema a resolver**: La automatización de los procesos de gestión de Puestos de Trabajo y Pagos Adicionales no engloba una solución integral para las entidades presupuestadas y empresariales cubanas.

Para darle solución al presente problema, se define como **objeto de estudio**: Procesos de gestión de Recursos Humanos. Por lo que se especifica como **campo de acción**: Los procesos de gestión de Puestos de Trabajo y Pagos Adicionales en las entidades presupuestadas y empresariales de Cuba, definiéndose como **objetivo general**: Desarrollar los componentes Puesto de Trabajo y Pagos Adicionales que permitan la gestión de los Puestos de Trabajo y los Pagos Adicionales en las entidades

INTRODUCCIÓN

presupuestadas y empresariales cubanas basada en los principios de independencia tecnológica.

Para dar cumplimiento a lo anteriormente planteado se definen los siguientes **objetivos específicos**:

- Valorar las bases teóricas metodológicas en torno a los Pagos Adicionales y Puestos de Trabajo.
- Identificar los procesos de gestión de Puestos de Trabajo y Pagos Adicionales en Cuba.
- Desarrollar una solución informática que permita la gestión de Puestos de Trabajo y Pagos Adicionales en las entidades presupuestadas y empresariales cubanas.
- Realizar pruebas a la solución.

De acuerdo a lo anterior se plantea la siguiente **idea a defender**: Con el desarrollo de los componentes Puesto de Trabajo y Pagos Adicionales se podría brindar una solución integral para la gestión de Puestos de Trabajo y Pagos Adicionales en las entidades presupuestadas y empresariales cubanas.

Posibles resultados: Se espera obtener una adecuada solución integral que gestione los procesos de Puestos de Trabajo y Pagos Adicionales en las entidades presupuestadas y empresariales de Cuba.

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Introducción

En este capítulo, se abordan distintos aspectos que serán utilizados como soporte técnico de los procesos que se pretenden analizar. Se exponen, mediante una breve descripción, los conceptos asociados a los procesos de gestión de Puestos de Trabajo y Pagos Adicionales. Se realiza un breve estudio acerca de las soluciones existentes para llevar a cabo dichos procesos, además de realizar una breve descripción de las tecnologías y herramientas seleccionadas para el desarrollo de la solución propuesta.

1.1 Gestión de Recursos Humanos

En la administración de empresas, se denomina Recursos Humanos al trabajo que aporta el conjunto de los empleados o colaboradores de una organización. La gestión de los Recursos Humanos se encarga de obtener y coordinar a las personas de una organización, de forma que consigan las metas establecidas. Para ello es muy importante cuidar las relaciones humanas. Las fases por las que ha pasado la gestión de los RH son cuatro:

- Administrativa: Es puramente burocrático y actúa sobre la disciplina y las remuneraciones. Las medidas a adoptar son de tipo reactivo.
- Gestión: Se empieza a considerar las necesidades de tipos sociales y sociológicas de las personas. Las medidas a adoptar son de tipo proactivo.
- Desarrollo: Se busca la conciliación entre las necesidades de los trabajadores y las necesidades económicas de la entidad. Se considera que las personas son elementos importantes para la entidad y se busca su motivación y eficiencia.
- Gestión estratégica de los RH: La gestión de los RH esta ligada a la estrategia de la entidad. Los trabajadores son la fuente principal de la ventaja competitiva de la entidad. (2)

1.2 Funciones de la Gestión de Recursos Humanos

La gestión de Recursos Humanos proporciona a las entidades en todo momento el personal capacitado necesario y en la cantidad requerida para desarrollar prósperamente los procesos productivos con un criterio de rentabilidad económica, como parte de la gestión de Recursos Humanos se puede adicionar o disminuir personal de la

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

organización así como el manejo burocrático desde que una persona ingresa a la entidad hasta que la abandona. Como parte de la función de retribución RH se encarga de definir una estructura de salarios que sea: motivadora, internamente equitativa y externamente competitiva; por otro lado también se encarga de tratar los conflictos de los trabajadores dentro y fuera de la entidad estableciendo medidas para la mejora del clima laboral, todo esto en función de que los individuos crezcan dentro de la organización. (2)

1.3 Puestos de Trabajo

Los estudios anteriores del Análisis y Descripción de Puestos de Trabajo (ADPT) coinciden con la revolución industrial en Europa y en Estados Unidos, Charles Babbage, en Europa, y Frederick Taylor en la Unión Americana fueron los primeros autores que plantearon que el trabajo podría y debía estudiarse de manera sistemática y en relación con algún principio científico.

En la medida en que fueron cambiando las relaciones con los empleados y las personas fueron más valiosas, los métodos y funciones de los Recursos Humanos se convirtieron en aspectos claves de las organizaciones.

Las tendencias actuales de la Gestión de los Recursos Humanos se dirigen hacia enfoques sistemáticos prácticos, multidisciplinarios y participativos que consideran el Análisis y Descripción de los Puestos de Trabajo (ADPT) como una herramienta básica para el establecimiento de toda política de Recursos Humanos pues casi todas las actividades desarrolladas en el área de Recursos Humanos se basan de uno u otro modo en la información que proporciona este procedimiento.

Chiavenato (1992) define un cargo como: “la reunión de todas aquellas actividades realizadas por una sola persona que pueden unificarse en un solo concepto y ocupan un lugar formal en el organigrama”. (3)

Chiavenato (1999), utilizando el término cargo como equivalente de puesto lo define: “un cargo constituye una unidad de la organización y consiste en un conjunto de deberes que lo separan y distinguen de los demás cargos. La posición del cargo en el organigrama define su nivel jerárquico, la subordinación, los subordinados y el departamento o división donde está situado”. (4)

De todas estas definiciones, si bien Chiavenato sintetiza en uno el concepto de cargo citado por los otros autores, se evidencia un enfoque todavía tradicional jerárquico, sin

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

embargo otros autores como Mondy y Noe (1997), utilizan un enfoque más transversal y lo presentan como parte de un proceso.

Así, Mondy y Noe (1997) lo definen: “un Puesto de Trabajo es la esencia misma del grado de productividad de una organización, por tanto, consiste en un grupo de tareas que se deben desarrollar para que una organización pueda alcanzar sus objetivos”. **(5)** Como base conceptual para el desarrollo de este trabajo ya que se refiere al Puesto de Trabajo de una forma más concreta y específica se toma la definición de Puesto de Trabajo concretada en la NORMA CUBANA 3000:2007 (SISTEMA DE GESTIÓN INTEGRADA DE CAPITAL HUMANO—VOCABULARIO) emitida por la Oficina Nacional de Normalización(NC), donde refiriéndose al cargo como equivalente al puesto lo define como: “los que aparecen recogidos en los calificadores comunes, de rama o actividad y propios de organismos, pertenecientes a las diferentes categorías ocupacionales, así como en resoluciones. En ellos se definen la denominación, contenido de trabajo y los requisitos para ocuparlos y expresan el empleo u oficio que desempeñan, los trabajadores en la organización”. **(6)**

Categoría ocupacional:

“Clasificación de los diferentes cargos, de acuerdo con las funciones y tareas que desarrollan los trabajadores, tomando en cuenta sus características disímiles, la complejidad y su vinculación con la producción y la prestación de servicios. Se agrupan en las categorías de operarios, trabajadores de servicios, trabajadores administrativos, técnicos y dirigentes”. (6)

La gestión de Puestos de Trabajo comprende:

➤ Diseño de Puestos de Trabajo:

Muy pocas empresas tienen presente que también el puesto de trabajo requiere diseño y una perfilación viable para la persona que lo detenta. Un diseño del puesto de trabajo erróneo es fuente principal de desmotivación, insatisfacción y baja productividad de los recursos humanos. (7)

Es así que Mondy, (1997) señala: “el diseño de puestos consiste en determinar las actividades específicas que se deben desarrollar, los métodos utilizados para desarrollarlas, y cómo se relaciona el puesto con los demás trabajos en la organización”. (5)

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

Chiavenato, (1999): “el diseño del cargo es la especificación del contenido, de los métodos de trabajo y de las relaciones con los demás cargos para cumplir requisitos tecnológicos, empresariales, sociales y personales del ocupante del cargo”. (4)

Estas definiciones antes citadas se relacionan en contenido pero se hallan un poco pobres, a continuación otra definiciones de otros autores que tratan el tema desde un punto de vista más amplio.

Así mismo, Chiavenato en su libro Gestión del Talento Humano agrega: “el diseño de los cargos es el proceso de organización del trabajo a través de las tareas necesarias para desempeñar un cargo específico. Incluye el contenido del cargo, las calificaciones del ocupante y las recompensas de cada cargo para atender las necesidades de los empleados y de la organización”. (8)

Fernández (2001) opina:“el diseño de puestos de trabajo es el procedimiento metodológico que nos permite obtener toda la información relativa a un puesto de trabajo”. (9)

Analizando el conjunto de concepciones citadas se puede definir que el diseño de Puestos de Trabajo es un proceso de organización del trabajo que tiene como objetivo estructurar los elementos, deberes y tareas de los puestos de una entidad, atendiendo el contenido, requisitos, responsabilidades, condiciones y calificaciones de los ocupantes, para lograr que el desempeño de los puestos ocupe un lugar en la oferta de valor que se desea brindar al cliente.

➤ El análisis y descripción de Puestos de Trabajo:

Se debe contar con la persona ideal para cada puesto de trabajo, que cumpla con el perfil y los requerimientos necesarios tal como lo indica el análisis de puestos.

Ya lo decía Fuertes y Pereda citado por García (2001): “Un buen análisis y descripción de puestos ha de recoger toda la información relativa a los puestos de la organización: el espacio físico, ambiente o entorno de trabajo, herramientas a utilizar, funciones y tareas del puesto, responsabilidades, conocimientos etc.; es decir, todo lo que directamente o indirectamente influye o puede influir en el correcto desempeño de un puesto de trabajo”. (10)

Por lo tanto, Harper y Lynch (1992) definen el análisis y descripción de puestos de trabajo como: “un proceso que consiste en determinar mediante un riguroso estudio los elementos o características inherentes a cada puesto”. (11)

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

De tal forma, el análisis de puestos sería un proceso de determinación de las funciones o actividades, responsabilidades y deberes, objetivos y condiciones que debe poseer un puesto y el tipo de persona que contenga las cualidades, cualificación, conocimientos, habilidades y capacidades que le permita desempeñarlo adecuadamente.

La descripción del cargo se refiere a las tareas, los deberes y responsabilidades del cargo, en tanto que las especificaciones del cargo se ocupan de los requisitos que el ocupante necesita cumplir. Por tanto, los cargos se proveen de acuerdo con esas descripciones y esas especificaciones. (4)

Mondy y Noe, (1997): “la descripción de puestos es un documento que proporciona información acerca de las tareas, deberes y responsabilidades del puesto. Las cualidades mínimas aceptables que debe poseer una persona con el fin de desarrollar un puesto específico se contienen en la especificación del puesto”. (5)

En general, la descripción del cargo presenta el contenido de éste de manera impersonal, y las especificaciones suministran la percepción que tiene la organización respecto de las características humanas que se requieren para ejecutar el trabajo, expresadas en términos de educación, experiencia, iniciativa, etc. Es preciso señalar que en la descripción de puestos debe plasmarse no lo que se hace; ya que lo que se esté haciendo puede que sea erróneo y requiera una revisión; por eso, la descripción del puesto debe estar enfocada al puesto ideal y no a lo que realiza el ocupante.

➤ Identificación y naturaleza del puesto:

El seleccionador debe identificar el puesto y localizar su ubicación departamental y geográfica. Deberá definir el tipo de trabajo, si corresponde al trabajo cualificado en el área de profesional universitario o del nivel técnico, o es trabajo manual o de administración general.

Con todo lo expuesto hasta aquí, se puede argumentar que es indispensable contar con el análisis de cada uno de los puestos, esto permitiría la posibilidad de obtener todas las características e información relativa a cada uno de los cargos. Además el uso de esta información permitirá establecer la descripción y especificación de cada puesto, y a su vez proporcionará la base para unificar los subsistemas que conforman la gestión de los RH.

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

1.4 Pagos Adicionales

En las relaciones contractuales que se establecen entre los empleadores y los trabajadores, mediante los contratos de trabajo y/o convenios colectivos que rigen en las empresas, se incorporan una serie de partidas económicas al margen del salario base y formando parte del salario total, que tienen varias denominaciones y cuyo objetivo es aumentar la remuneración a cambio de más producción, o mitigar lo negativo de trabajar en determinadas condiciones. En el ámbito de la economía cubana estas partidas económicas son conocidas como Pagos Adicionales y en el ámbito mundial como pluses salariales y complementos del salario, estas denominaciones incluyen todas aquellas retribuciones monetarias que sin formar parte del salario base o salario escala son atribuidas al trabajador a causa de determinadas condiciones que tienen que ver con su persona, Puesto de Trabajo o resultados de la entidad.

Salario base: "Es el pago por complejidad y responsabilidad aprobado para cada ocupación o cargo, según el grupo de la escala salarial vigente". (6)

Los Pagos Adicionales varían de acuerdo al marco económico vigente, pero sin importar el mismo se destacan algunos como:

- Pago de nocturnidad: cuando se trabaja en horario nocturno.
- Pago de productividad o de beneficios: Se otorga generalmente en función de los resultados económicos de la entidad.
- Pago de transporte: cuando la empresa está lejos de la residencia de los trabajadores.
- Los PA del Puesto de Trabajo (complemento objetivo) son percepciones que el trabajador recibe por razón de las características del Puesto de Trabajo. Son: complementos por periodicidad, toxicidad, peligrosidad, suciedad, máquinas, turnos, trabajo nocturno o cualquier otro que deba percibir el trabajador por razón de las características del Puesto de Trabajo y de la forma de realizar su actividad profesional que comporte concepción distinta del trabajo corriente.

En la antes mencionada NORMA CUBANA 3000:2007 se definen los Pagos Adicionales como: "Pagos por trabajar en determinadas condiciones, cargos, actividades, ramas o sectores debido a su importancia económica, o del servicio que se presta, o como reconocimiento a trabajadores por sus aportes excepcionales y significativos al desarrollo económico y social del país y otros factores extracalificatorios". (6)

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

Formando parte de los Pagos Adicionales se encuentra el **Pago por Condiciones Laborales Anormales** que según esta norma es: “Pago que aprueba el Ministerio de Trabajo y Seguridad Social, a partir de la determinación de la existencia de riesgos que afectan la integridad de los trabajadores y que puedan provocar enfermedades profesionales o accidentes de trabajo y que no son posibles minimizar, mediante la aplicación de medidas técnicas u organizativas, o ambas, lo cual debe ser rigurosamente avalado por el organismo o entidad nacional que lo solicite”. (6)

1.5 Análisis de soluciones existentes

Existen en la actualidad infinidad de sistemas que resuelven de alguna u otra forma la gestión de los procesos de PT y PA, desde sistemas ERP (que sus siglas en inglés significan Planeación de Recursos Empresariales), software de gestión de RH y nóminas hasta sistemas para el control de pagos y retenciones. Dentro de estos grupos de sistemas se pueden mencionar los siguientes:

Nivel internacional

COLSIN-RH

- Sistema de Administración de Talento Humano desarrollado por COLSIN-LTDA, empresa colombiana fundada en 1987 de Informática y Sistemas.
- El módulo de planeación estratégica ofrece herramientas para la selección de personal y estructuración de cargos.
- Permite definir la información requerida para especificar los puestos y la estructura organizacional de la empresa. Las funciones, habilidades requeridas, competencias y topes salariales.
- Posee gran flexibilidad en la definición de conceptos de pago.

Otras características:

Desarrollado: Power Builder y Java tecnología Web 2.0 • Idioma: español/inglés

Bases de datos ORACLE, SQL Server, DB/2, SYBASE, INFORMIX, PostgreSQL y cualquier otro motor que permita conectividad vía ODBC. Navegador Mozilla Firefox 2.0, Microsoft Explorer 6.0 para la WEB. (12)

Sofos

Sistema para el control de los pagos y las retenciones del personal, creado por MICROIDEAS DE SOFTWARE, C.A. Empresa Venezolana fundada en el año 1997, es

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

muy completo y 100% programable, permitiendo una adaptación total a las exigencias del cliente. Sofos permite definir cualquier tipo de cálculo individual o colectivo, por ejemplo: Nómina Semanal, Quincenal, Confidencial, Mensual, Bonificaciones Especiales y Utilidades.

Sofos para Windows o DOS versión 5.0 permite procedimientos como:

- Cálculo del Bono Nocturno.
- Cálculo de las horas extras diurnas y nocturnas.
- Trabajos realizados en días feriados o de Descanso.
- Indemnización por antigüedad.

Otras características:

Basado en el standard Intel[®], desarrollado totalmente en el Leguaje de 4ta.Generación Foxpro[®] 2.6 para Windows, versión 2.6 para DOS y Visual Foxpro 6.0 en el caso de la edición Cliente-Servidor de Microsoft Corporation. (13)

Nivel nacional

Assets

Sistema de gestión integral comercializado por la firma panameña D'MARCO S.A. y distribuido en Cuba por INFOMASTER, entidad informática perteneciente a la Empresa Nacional de Producción y Servicios a la Educación Superior del MES.

Dentro de su modulo de Capital humano Assets permite:

- Modificar plantillas, introducir cambios en la estructura organizativa, crear nuevos cargos y realizar conversiones de plazas.
- Permite efectuar el pago de empresas bajo el régimen de perfeccionamiento empresarial, incluyendo el uso del coeficiente de participación laboral y la aplicación del aporte y descuento en caso que proceda.

Otras características:

Aplicación cliente-servidor programada en Visual Basic 6.0 y Microsoft SQL Server 2000, utilizando adicionalmente Crystal Reports 7.0 para la generación de reportes de salidas. (14)

RODAS XXI: Sistema integral económico desarrollado por la empresa cubana CITMATEL, su modulo Nóminas permite calcular y contabilizar nóminas de salario para cualquier tipo

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

de pago (sueldo fijo, jornal, pago por rendimiento), incluyendo el pago de horas extras, interrupciones y condiciones laborales anormales.

Su modulo Capital Humano permite crear plantilla de cargos, variaciones de plantilla de cargos, conversión de plazas, pagos de horas extras y pagos por condiciones laborales anormales. (15)

Una vez realizado el estudio de los sistemas existentes que tratan de dar solución al problema planteado entre los que se encuentran los mencionados anteriormente se pudo constatar el por qué de la necesidad de desarrollar una nueva solución para la gestión de los Puestos de Trabajo y Pagos Adicionales en Cuba. De los sistemas cubanos se puede decir que cumplen solamente con las necesidades de un pequeño grupo de entidades, por otra parte con los sistemas desarrollados por otros países implantados en Cuba sucede lo mismo incluyendo que muchos están enfocados a un sector específico o diseñados en el marco de otras economías y existe la dependencia que se crea a un suministrador externo, en quienes se seguiría invirtiendo hasta lograr adaptar el sistema a las necesidades de la economía cubana, aún cuando estos sean productos de reconocido prestigio. De los sistemas existentes en el área internacional decir que habría que pagarlos y por consiguiente es un gasto más que afecta la economía del país, la mayoría de las herramientas con que fueron desarrollados no son software libre.

Por tanto para intentar cumplir con el objetivo general de esta investigación se optó por el desarrollo de los Componentes Puesto de Trabajo y Pagos Adicionales del Subsistema Capital Humano integrado al Sistema CEDRUX como una solución integral para las entidades presupuestadas y empresariales cubanas, basada en los principios de independencia tecnológica y con funcionalidades generales de los procesos y las particularidades de la economía cubana.

1.6 Modelo de desarrollo adoptado

El Modelo de Desarrollo Orientado a Componentes (MDOC) es la metodología definida por el proyecto ERP-Cuba, para la realización del sistema CEDRUX y por consiguiente para este trabajo. Este es un modelo de desarrollo orientado a las necesidades y artefactos generados durante el proceso de desarrollo de CEDRUX. Es una combinación de diferentes metodologías de las cuales se ha tomado lo que seria más conveniente para

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

llevar a término el proyecto. Entre las características que posee se encuentran que se modela el negocio mediante procesos; la ingeniería de requisitos es mucho más clara que en otras metodologías; es orientada a componentes, posibilitando la independencia de funciones del sistema a la hora de mantener o modificar el sistema funcional; como lenguaje de modelado para el negocio se utiliza BPMN y para el diseño UML. Este modelo de desarrollo permitirá la generación de artefactos de vital importancia en el análisis y el diseño como son: Modelo de proceso de negocio, Descripción de procesos de negocio, Modelo conceptual, Prototipo de interfaz de usuario, Especificación de requisitos, Casos de prueba, Diagrama de clases y Descripción del diseño de clases.

El uso de este modelo posee los siguientes beneficios: (16) (17)

- Reutilización del software:
Lleva a alcanzar un mayor nivel de reutilización de software.
- Simplifica las pruebas:
Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados.
- Simplifica el mantenimiento del sistema:
Cuando existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema.
- Mayor calidad:
Dado que un componente puede ser construido y luego mejorado continuamente por un experto u organización, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo.

1.7 Herramientas y tecnologías

En la actualidad, gracias al avance que ha alcanzado la informática, han surgido diversidad de herramientas y tecnologías, estas herramientas y tecnologías tienen gran importancia y que permiten que el hombre pueda usar las computadoras de una forma más sencilla para la creación de aplicaciones de todo tipo que han revolucionado el mundo con todas sus facilidades y mejoras. Dentro de estas aplicaciones se incluyen las Web, las cuales ofrecen grandes facilidades para establecer comunicación con el usuario

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

por su capacidad para ser visualizadas desde cualquier parte del mundo haciendo uso de un navegador.

Por ello a continuación se exponen el modelo, las tecnologías y herramientas que por sus características y ventajas fueron definidas en el marco del proyecto ERP-Cuba y aplicadas al sistema Cedrux y por consiguiente al desarrollo de la solución propuesta.

1.7.1 Herramientas CASE

Se puede definir a las Herramientas CASE como 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. (18)

1.7.1.1 Visual Paradigm

Visual Paradigm para UML (Lenguaje de Modelado Unificado) es una herramienta CASE que utiliza UML como lenguaje de modelado y también soporta BPMN, soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Visual Paradigm ofrece:

Entorno de creación de diagramas para UML 2.0

Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.

Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.

Capacidades de ingeniería directa (versión profesional) e inversa.

Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.

Disponibilidad de múltiples versiones, para cada necesidad.

Disponibilidad de integrarse en los principales IDE.

Disponibilidad en múltiples plataformas. (19)

1.7.2 Lenguajes de Modelado

Un lenguaje de modelado es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar (parte de) un diseño de software. Algunas organizaciones los

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

usan extensivamente en combinación con una metodología de desarrollo de software para avanzar de una especificación inicial a un plan de implementación y para comunicar dicho plan a todo un equipo de desarrolladores. El uso de un lenguaje de modelado es más sencillo que la auténtica programación, pues existen menos medios para verificar efectivamente el funcionamiento adecuado del modelo. (20)

1.7.2.1 UML

UML se dice que es un lenguaje estándar para escribir planos de software, que se utiliza para visualizar, especificar, construir y documentar los artefactos de un mismo sistema que involucra una gran cantidad de software. Su alfabeto está constituido por elementos y relaciones, los cuales al combinarse cobran sentido al armar una colección de palabras formando diferentes tipos de diagramas. Es de vital importancia tener en cuenta que UML es sólo un lenguaje y por lo tanto es tan sólo una parte de un método de desarrollo de software, que se utiliza para visualizar, especificar, construir y documentar los artefactos que se obtienen como resultado de un proceso de construcción de software, lo cual lo convierte en una herramienta estándar para escribir planos de software, mas no en un proceso o metodología de desarrollo de software. (21)

UML contiene:

- Diagrama de casos de uso
- Diagrama de clases
- Diagrama de estados
- Diagrama de secuencias
- Diagrama de actividades
- Diagrama de colaboraciones
- Diagrama de componentes
- Diagrama de distribución

1.7.2.2 BPMN

BPMN (Business Process Modeling Notation) es un estándar de modelado de procesos de negocio, en donde se presentan gráficamente las diferentes etapas del proceso del mismo. Ha sido diseñado específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre estos.

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

BPMN y las extensiones de UML ayudan a modelar la situación actual y deseada en los procesos de negocio. BPMN ha sido desarrollado para proveer a los usuarios de una notación de uso libre. (22)

1.7.3 Servidor Web

Un servidor web es un programa que implementa el protocolo HTTP (Protocolo de Transferencia de Hipertexto). Este protocolo está diseñado para transferir lo que se llama hipertextos, páginas web o páginas HTML (Lenguaje de Marcado de Hipertexto): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

Es un ordenador que usa el protocolo http para enviar páginas web al ordenador de un usuario cuando el usuario las solicita a través de un navegador. Los servidores web, servidores de correo y servidores de bases de datos son a lo que tiene acceso la mayoría de las personas al usar Internet. (23)

1.7.3.1 WAMP 5

WAMP5 es un instalador para Windows; que compone una completa solución WAMP como sus siglas lo indican sirve para:

Windows - Apache - MySQL- PHP5.

Este software instala:

Apache 2.x.x.

PHP 5.x.x

MySQL 5.x.x

PHPmyadmin

SQLitemanager

Wampserver service manager

WAMP5 viene con un administrador de servicios como icono en la bandeja del sistema.

Este administrador permite controlar completamente el servidor y proyectos locales.

WAMP5 no está diseñado para servidor de producción, pero si como un entorno de desarrollo. (24)

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

1.7.4 Lenguajes de programación

Un lenguaje de programación es una construcción mental del ser humano para expresar programas. Está constituido por un grupo de reglas gramaticales, un grupo de símbolos utilizables, un grupo de términos con sentido único y una regla principal que resume las demás. Para que ésta construcción mental sea operable en un computador debe existir otro programa que controle la validez o no de lo escrito. A éste se le llama traductor. (28)

1.7.4.1 JavaScript (tecnología del lado del cliente)

Para la programación de las vistas se escogió JavaScript que es un lenguaje basado en objetos y guiado por eventos, diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet. Es un lenguaje interpretado ya que los programas escritos con él se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios solo basta con que el navegador soporte JavaScript.

Ventajas de JavaScript:

- Los programas escritos en este lenguaje no requieren de mucha memoria ni tiempo adicional de transmisión, por ser pequeños y compactos.
- JavaScript no requiere un tiempo de compilación; ya que los scripts se pueden desarrollar en un período de tiempo relativamente corto.
- Es independiente de la plataforma hardware o sistema operativo. (25)

1.7.4.2 AJAX (tecnología del lado del cliente) (26) (27)

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript Asíncrono y XML) facilita la creación de aplicaciones interactivas en la Web que se ejecutan en el navegador de los usuarios y mantienen comunicación asíncrona con el servidor; posibilitando realizar cambios sobre una página sin necesidad de recargarla, aumentando de esta forma la interactividad, velocidad y usabilidad de la misma, fundamentalmente por esta característica fue seleccionada la tecnología AJAX para la solución.

AJAX no es una tecnología en sí mismo. En realidad, se trata de la unión de varias tecnologías que se desarrollan de forma autónoma y que se unen de formas nuevas y sorprendentes.

AJAX está conformado por:

- XHTML y CSS, para crear una presentación basada en estándares.

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XML, Http Request, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías (explicada anteriormente).
- Provee un mecanismo para mezclar y hacer coincidir XML con XHTML.
- Las aplicaciones son más rápidas e interactivas, al estilo aplicaciones de escritorio.
- Reduce de manera significativa tener que cargar información continuamente del servidor, actualizando solamente porciones de la página.
- Cuando se utiliza AJAX adecuadamente en el desarrollo de una aplicación, se reduce de manera significativa los tiempos de carga inicial.

A continuación se explican las características más importantes de algunas de las otras tecnologías que componen AJAX que serán utilizadas en la solución:

XML: Es el estándar de Extensible Markup Language (Lenguaje de Etiquetado Extensible), conformado por un conjunto de reglas para definir etiquetas semánticas orientadas a organizar un documento en diferentes partes. Permite al usuario definir sus propios lenguajes de anotación adaptados a sus necesidades y contiene tres características muy importantes que son: extensibilidad, estructura y validación.

Ventajas de XML:

- Las aplicaciones se pueden generar rápidamente y su mantenimiento es más sencillo.
- Separa los datos de la presentación y del proceso, lo que permite mostrar y procesar los datos al gusto deseado con sólo aplicar distintas hojas de estilo y aplicaciones.
- La información es más accesible y reutilizable, por la flexibilidad de las etiquetas de XML sin tener que amoldarse a reglas específicas de un fabricante.

JSON: acrónimo de JavaScript Object Notation (Notación de Objetos de JavaScript), es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX. Una de las ventajas de JSON sobre XML como formato de intercambio de datos es que es mucho más sencillo escribir un analizador semántico de JSON.

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

1.7.4.3 PHP (tecnología del lado del servidor)

PHP fue el lenguaje seleccionado para la programación de la lógica de negocio el mismo es un lenguaje sencillo, de sintaxis cómoda y dispone de muchas librerías que facilitan en gran medida el desarrollo de las aplicaciones; convirtiéndolo en el favorito de millones de programadores en todo el mundo.

Características de PHP:

- Dispone de una conexión propia a varios sistemas de base de datos como: MySQL, PostgreSQL y Oracle.
- Incorpora bibliotecas que contienen funciones integradas para realizar útiles tareas relacionadas con la Web.
- Puede generar imágenes GIF al instante, establecer conexiones a otros servicios de red, enviar correos electrónicos, trabajar con cookies y generar documentos PDF, todo con unas pocas líneas de código.
- Es un producto de código abierto, soportado por una gran comunidad de desarrolladores que se encargan de encontrar y reparar los fallos de funcionamiento.
- Es un lenguaje multiplataforma.
- Permite las técnicas de Programación Orientada a Objetos.
- No requiere definición de tipos de variables.
- Posee tratamiento de errores.
- PHP 5.2 es una versión de PHP que además incluye:
- Soporte sólido para Programación Orientada a Objetos (OOP) con PHP Data Objects.
- Mejoras de rendimiento.
- Mejor soporte a XML.

1.7.5 Ambiente de desarrollo integrado (IDE)

Un entorno de desarrollo integrado o Integrated Development Environment (IDE) en inglés, es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien, poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDE pueden ser aplicaciones por si solos o pueden ser parte de aplicaciones existentes. (28)

1.7.5.1 Zend Studio para Eclipse 6.0 (29)

Zend Studio para Eclipse IDE es la herramienta seleccionada para la programación a través de los lenguajes del lado del cliente y del servidor, es la última versión del popular entorno de programación integrado (IDE), diseñado para desarrolladores profesionales de PHP, esta nueva versión combina un IDE versátil y potente con las capacidades de expansión del ecosistema del proyecto Eclipse. Dispone de un entorno mucho más flexible y profesional para controlar todo el ciclo de vida de un desarrollo. Entre sus funcionalidades, se destacan las capacidades de refactorización del código fuente, funcionalidad que permite adecuar el comportamiento externo de una función/clase sin cambiar el funcionamiento interno, que junto a los nuevos wizards y capacidades de generación de código facilitarán el trabajo a los desarrolladores.

Características nuevas que incluye:

- Código refactoring.
- Generación de código y magos Archivo.
- Código de Cobertura.
- PHP Unit pruebas de apoyo.
- El acceso al ecosistema de plugins de Eclipse.
- Apoyar el desarrollo de múltiples idiomas.
- Mejora el Editor de PHP con el formato avanzado, para listas de tareas y problemas de vista.
- Mejora del soporte de JavaScript.
- Mejora de apoyo, incluyendo HTML, HTML WYSIWYG, Código plegables, arrastrar y soltar los componentes y más.
- Mejora de Zend Framework con el apoyo del Proyecto Framework, plantillas y código MVC.

1.7.6 Frameworks

En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros software

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Los frameworks son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional. (30)

1.7.6.1 Zend Framework

Se trata de un framework para desarrollo de aplicaciones Web y servicios Web con PHP, brinda soluciones para construir sitios web modernos, robustos y seguros. Además es Open Source y trabaja con PHP 5.

Zend Framework es desarrollado por Zend que es la empresa que respalda comercialmente a PHP, sus principales características:

- Trabaja con MVC (Modelo Vista Controlador).
- Cuenta con módulos para manejar archivos PDF, canales RSS, Servicios Web (Amazon, Flickr, Yahoo), etc.
- El Marco de Zend también incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar su base de datos, sin tener que escribir ninguna consulta SQL.
- Una solución para el acceso a base de datos que balancea el ORM con eficiencia y simplicidad. Se puede ver esta función en el futuro.
- Completa documentación y test de alta calidad.
- Soporte avanzado para internacionalización.
- Robustas clases para autenticación y filtrado de entrada.
- Clientes para servicios web, incluidos Google Data APIs y Strikelron. (31)

1.7.6.2 Zend_Ext Framework

Es un framework open Source, que esta diseñado para php 5 y buenas capacidades de ampliación. Es elaborado a partir de Zend Framework cumpliendo con todas sus características. Este trae de novedoso un controlador vertical para el control de las acciones realizada por las vistas hacia el controlador, un motor de reglas para las validaciones en el servidor, se le incluyo el IOC para la comunicación en entre los módulos o componentes. Se le incorporo la integración con el ORM Doctrine Framework para trabajo en la capa de abstracción a base de datos y el ExtJs Framework para el desarrollo de las vistas. (32)

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

1.7.6.3 Doctrine Framework

Es un potente y completo sistema ORM (Object Relational Mapper) para PHP 5.2+ que incorpora una DBL (capa de abstracción a base de datos). Uno de sus rasgos importantes es la habilidad de escribir opcionalmente las preguntas de la base de datos orientado a objeto. Esto les proporciona una alternativa poderosa a diseñadores de SQL que mantiene un máximo de flexibilidad sin requerir la duplicación del código innecesario. También permite exportar una base de datos existente a sus clases correspondientes y también convierte clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos. (33)

1.7.7 Sistema Gestor y herramienta de Base de Datos

Un Sistema Gestión de Bases de Datos (SGBD (Data Base Management System DBMS)) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. El objetivo primordial de un SGBD es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos. El sistema de gestión de la base de datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, y proporciona acceso controlado a la misma, todo esto mediante una herramienta de administración de base de datos que en este caso se será EMS PostgreSQL Manager 3. (34)

1.7.7.1 PostgreSQL

Es un gestor de bases de datos relacional orientada a objetos, libre y gratuito. Está liberado bajo la licencia BSD, lo que significa que se puede disponer de su código fuente, modificarlo a voluntad y redistribuirlo libremente.

Presenta las siguientes propiedades:

Atomicidad: Asegura la realización de una operación, por lo que ante un fallo del sistema esta no queda a medias.

Consistencia: Posibilita la ejecución de aquellas operaciones que no van a romper las reglas y directrices de integridad de la base de datos.

Aislamiento: Mediante un sistema denominado MVCC (Acceso concurrente multiversión) asegura que una operación no pueda afectar a otras, de esta manera dos transacciones sobre la misma información no genera error.

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

Durabilidad: Asegura la permanencia de una operación realizada, y aunque falle el sistema esta no podrá deshacerse.

PostgreSQL presenta las siguientes características:

Se puede instalar un número ilimitado de veces sin temor de sobrepasar la cantidad de licencias.

Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos.

Posee confiabilidad, seguridad y flexibilidad. (35)

1.7.7.2 EMS PostgreSQL Manager 3

EMS SQL Manager para PostgreSQL es una herramienta mediante la cual se puede administrar todo tipo de bases de datos PostgreSQL, ya que cuenta con un paquete de herramientas optimizado para dicha función.

Mediante esta aplicación es posible importar y exportar datos y realizar tareas de mantenimiento, crear índices, permisos de acceso y usos.

Además, provee una navegación segura por las bases de datos, ampliando las posibilidades de conexión por diferentes métodos, inclusive mediante protocolo HTTP.

Últimas Modificaciones:

Soporta las últimas versiones de PostgreSQL.

Mejoradas sus herramientas de edición. (36)

1.7.8 Control de versiones

Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas (por ejemplo, para algún cliente específico).

El control de versiones se realiza principalmente en la industria informática para controlar las distintas versiones del código fuente. Sin embargo, los mismos conceptos son aplicables a otros ámbitos como documentos, imágenes, sitios web, etcétera. Aunque un sistema de control de versiones puede realizarse de forma manual, es muy aconsejable disponer de herramientas que faciliten esta gestión (CVS, Subversion, SourceSafe, ClearCase, Darcs, Plastic SCM, Git, Mercurial, etc. (37)

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

1.7.8.1 TortoiseSVN

Es un software de sistema de control de versiones, libre, bajo una licencia de tipo Apache/BSD y de fácil uso, también se le conoce como “svn” por ser ese el nombre de la herramienta de línea de comandos.

Ventajas:

- Se sigue la historia de los archivos y directorios a través de copias y renombrados.
- Las modificaciones (incluyendo cambios a varios archivos) son atómicas.
- Se envían sólo las diferencias en ambas direcciones.
- Permite selectivamente el bloqueo de archivos.
- Cuando se usa integrado a Apache permite utilizar todas las opciones que este servidor provee a la hora de autenticar archivos (SQL, LDAP, PAM, etc.). (38)

1.7.9 Herramienta de apoyo

1.7.9.1 Navegador

El navegador sobre el cual correrá la aplicación es Mozilla Firefox el cual permite incorporar medidas contra fraudes (phishing) y malware, además de contar con un sistema de avisos contra Hacker "one-click site ID". Teniendo en cuenta los procesos de código abierto con que cuenta esta herramienta existen miles de ingenieros en seguridad trabajando para mantener a salvo a la información personal de los usuarios.

El Firefox dispone de más de 5.000 complementos que ayudarán a los usuarios a personalizarlo y ajustarlo de acuerdo a sus necesidades. (39)

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

Conclusiones

Para lograr los objetivos principales de cualquier organización los Puestos de Trabajo constituyen un elemento imprescindible, y nada mejor que los Pagos Adicionales como mecanismo de distinción y mitigador de riesgos en determinados Puestos de Trabajo. En este capítulo se explico brevemente que son los RH, en que consiste su gestión y sus funciones; se analizaron conceptos claves asociados con la gestión de Puestos de Trabajo y Pagos Adicionales; además se justificaron las herramientas seleccionadas para el desarrollo de la solución propuesta a la cual se arribó luego de una valoración de otras soluciones existentes.

Habiendo cumplido los primeros objetivos trazados se crean las condiciones para comenzar y conocer el desarrollo de los componentes.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

Introducción

Este capítulo recoge los resultados obtenidos en los modelos de negocio, análisis y diseño de la solución propuesta. Primeramente se realizará la descripción de la propuesta de solución, para lo que se esquematizan los procesos de negocio que tienen que ver con el campo de acción. Además se plasmarán los requisitos funcionales y no funcionales del sistema a realizar, así como los prototipos de los usuarios pertenecientes a los requisitos previamente identificados conjuntamente con el modelo conceptual.

2.1 Caso de estudio

Se propone desarrollar dos componentes que contribuyan con la gestión integral de los Recursos Humanos en Cuba, los mismos están basados en los procesos de gestión integral de Puestos de Trabajo y Pagos Adicionales en las entidades presupuestadas y empresariales cubanas.

Objeto de automatización

Los procesos que serán objetos de automatización son los procesos de gestión de Puestos de Trabajo y Pagos Adicionales en las entidades presupuestadas y empresariales cubanas.

2.2 Propuesta de solución

Después de haber analizado profunda y detalladamente los problemas que a la hora de gestionar los Puestos de Trabajo y los Pagos Adicionales existen en las entidades presupuestadas y empresariales cubanas, tomando como punto de partida la existencia de sistemas que resuelven estos problemas en algunas de dichas entidades pero no en todas, se propone desarrollar una solución que cubra estos procesos de forma eficiente y de manera integral.

2.3 Descripción de los procesos de negocio

Un proceso de negocio no es más que un conjunto de actividades que tomando una o varias clases de entrada generan una salida que tiene valor para el cliente. Los procesos de negocio representan el flujo de trabajo y de información a través del negocio. La descripción de los procesos de negocio hace más viable el paso al análisis ya que

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

posibilita una comprensión más clara de los procesos en cuestión y contribuye a que los requisitos que se definan satisfagan las necesidades del cliente.

2.3.1 Gestión de Puestos de Trabajo

Una vez definida la Plantilla de cargos y ocupaciones en el módulo de Estructura y composición es necesario definir los Puestos de Trabajo asociados a cada cargo dentro de la estructura de la entidad, lo cual facilita grandemente poder darle un tratamiento diferenciado a cada uno de ellos.

Por Puesto de Trabajo se entiende el lugar físico dentro de un proceso productivo donde los trabajadores interactúan con los medios de producción.

La descripción de las funciones y tareas de los cargos y ocupaciones no reflejan en toda su magnitud los requerimientos y especificidades de los Puestos de Trabajo asociados a ellos. Un mismo cargo dentro de un área de trabajo puede agrupar una amplia gama de Puestos de Trabajo que además de abarcar una serie de elementos similares poseen sus propias peculiaridades que lo hacen diferentes entre ellos, tener en cuenta estos detalles permite alcanzar una mayor calidad y eficiencia en las procesos de Selección e integración, Evaluación del desempeño y Capacitación de desarrollo de los Recursos Humanos.

Lograr esta caracterización de los Puestos de Trabajo permite además:

- Poder diferenciar los Pagos Adicionales que están asociados a la persona del resto de los pagos.
- Establecer los requerimientos para cada Puesto de Trabajo.
- Crear grupos de Puestos de Trabajo con características muy similares.
- Adicionar los Horarios de trabajo, Sistemas de pagos a tiempo y el fondo de tiempo laboral.

2.3.2 Gestión de Pagos Adicionales

La administración de las entidades, podrá aplicar o tramitar la aprobación con la instancia correspondiente, según corresponda, Pagos Adicionales a los trabajadores hasta donde se lo permitan sus posibilidades de autofinanciamiento, de acuerdo con la legislación vigente, para estimular la rotación de turnos, nocturnidad, pagos por condiciones laborales anormales de trabajo y otros Pagos Adicionales que son aprobados centralmente por el Organismo facultado para ello.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

Los Pagos Adicionales aprobados legalmente que caractericen las condiciones de trabajo, la complejidad o responsabilidad del Puesto de Trabajo, importancia de la actividad, rama o empresa de que se trate, así como el pago adicional por aplicar el Perfeccionamiento Empresarial, como regla, se vinculan al cumplimiento de los indicadores que se definan en los sistemas de pago por los resultados.

Cada entidad debe elaborar un procedimiento donde se incluyan todos los Pagos Adicionales que tienen aprobados la empresa, el cual debe contener para cada pago, los aspectos siguientes:

- Denominación del pago adicional.
- Norma jurídica que lo autoriza (especificar nivel de aprobación)
- Trabajadores abarcados en el mismo.
- Condiciones para su aplicación. (si depende del cumplimiento de determinados indicadores, en que período se aplica, quien certifica su aplicación, en que casos se pierde el derecho a su cobro).
- Procedimiento para el cálculo de la cuantía a devengar por el trabajador (que elementos de salario se incluyen en su base de cálculo).

Los Pagos Adicionales son aprobados por el Ministerio de Trabajo y Seguridad Social a propuesta de los jefes de los órganos, organismos y entidades nacionales, oído el parecer del sindicato nacional correspondiente y se expresan en cuantías fijas.

Los Pagos Adicionales de uso general aparecen relacionados en las resoluciones del Ministerio de trabajo donde se aprueban la escala salarial única y las escalas para los trabajadores de los Sistemas nacionales Salud, Educación y Educación superior, mientras los Pagos Adicionales de uso específico son establecidos en la legislación relacionada con la aprobación de los Sistemas salariales para cada actividad o sector de la economía nacional.

Actualmente la dirección del Ministerio de trabajo en coordinación con los Organismos de la administración del estado lleva a cabo un proceso en todo el país para la conversión de todos los Pagos Adicionales aprobados actualmente en porcentos a cuantías fijas.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

2.4 Mapa de procesos del negocio

El mapa o diagrama de procesos del negocio muestra de manera más abstracta pero más simplificada para desarrolladores de software como se llevan a cabo, en este caso, los procesos de gestión de Puestos de trabajo y Pagos Adicionales. De acuerdo a los procesos de negocio anteriormente descritos se desarrolla el mapa de procesos del negocio de este trabajo (ver Figura 1). De esta forma de color gris aparecen procesos llevados a cabo por subsistemas externos pero que se relacionan con el proceso de Administración de Capital Humano y específicamente con los procesos en cuestión, de color rojo se muestran las actividades a automatizar.

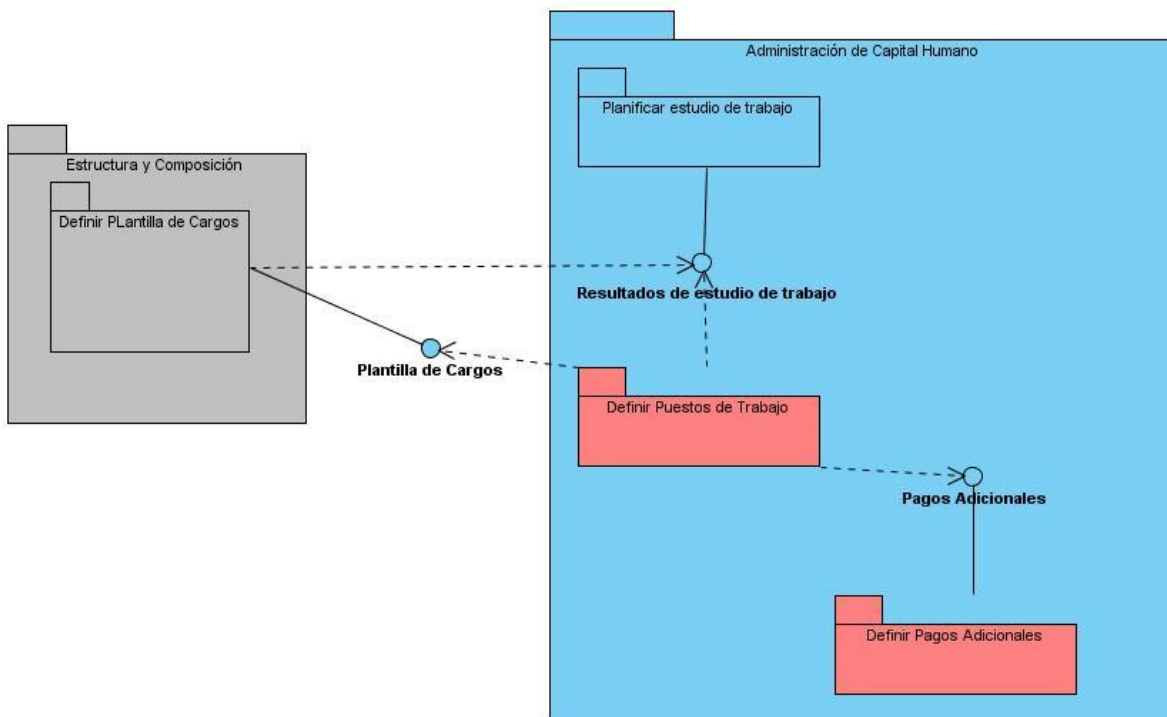


Figura 1. Mapa de procesos del negocio

2.5 Requisitos Funcionales

Uno de los aspectos más importantes para el éxito de un proyecto de software lo constituye el levantamiento correcto de los requisitos funcionales, lo que constituye el entendimiento y comprensión de los problemas que se necesitan solucionar y cómo resolverlos. Este entendimiento viene forzado por la cantidad y la calidad de información

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

suministrada por la persona que tenga ese problema. Información en la que debería de especificarse, de forma clara e inequívoca la realidad: un producto nuevo, una actualización de un producto antiguo, o la agregación de una nueva funcionalidad a un producto ya en funcionamiento. Todo ello expuesto en unos requisitos que en el propio lenguaje del propietario del sistema, actual o futuro, expresa su idea o visión de lo que necesita; permite lograr una comunicación efectiva entre los usuarios y el equipo de proyecto con el objetivo de llegar a un entendimiento de lo que hay que realizar, siendo así la clave del éxito en la producción de un software.

Como una vía de solución se determino crear Grupos de Puestos de Trabajo para así tener un mejor y más organizado control sobre los puestos. Estos grupos de puestos se compondrán por puestos con determinadas características comunes y a la hora de agregar los pagos adicionales se pueden agregar al grupo de puesto ahorrándose trabajo.

De acuerdo a las actividades realizadas hasta ahora la solución debe ser capaz de:

- Mostrar la información sobre los Pagos Adicionales existentes en la entidad.
 - Adicionar un nuevo Pago Adicional.
 - Modificar un Pago Adicional existente.
 - Eliminar un Pago Adicional.
- Mostrar los Puestos de Trabajo existentes en la entidad.
 - Adicionar un nuevo Puesto de Trabajo.
 - Modificar un Puesto de Trabajo.
 - Eliminar un Puesto de Trabajo.
- Mostrar los Grupos de Puestos de Trabajo existentes en la entidad.
 - Adicionar un nuevo Grupo de Puestos de Trabajo.
 - Modificar un Grupo de Puestos de Trabajo existente.
 - Eliminar un Grupo de Puesto de Trabajo.

2.5.1 Especificación de los requisitos funcionales

Para lograr una mejor comprensión de la funcionalidad asociada a cada proceso es necesario realizar una descripción textual de los mismos. La descripción puede ser elaborada de forma breve o extendida y debe ir acompañada del prototipo correspondiente. El prototipo del sistema que se construye en este punto muestra una visión de las pantallas diseñadas para cada requerimiento funcional, de forma estática

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

para la verificación de la realización de la interfaz. A continuación se muestran y describen los paquetes de requisitos funcionales y sus especificaciones que son necesarios para llevar a cabo la gestión de los Puestos de Trabajo y los Pagos Adicionales.

R1 Requisito funcional Gestionar Pago Adicional

El requisito funcional gestionar Pagos Adicionales tiene en cuenta la manera de gestionar los Pagos Adicionales en una entidad a través de adicionar Pagos Adicionales, modificar y eliminar los pagos.



Figura 2. Prototipo gestionar Pagos Adicionales.

Tabla 1. Especificación del requisito Adicionar Pago Adicional

Conceptos tratados	Conceptos	Atributos
	Pago adicional.	Código, denominación, descripción, elemento del gasto, comportamiento, asociado a, fecha efectiva, valor, ecuación.
Precondiciones	Precondiciones	Pre-requisito

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

	No procede.	No procede.
Descripción	<p>Al seleccionar la opción adicionar pago adicional el sistema permite agregar el código, la denominación, descripción, valor, seleccionar la fecha efectiva y procede a indicar el comportamiento y el elemento de gasto, además indicar si el pago esta asociado a un trabajador o a un Puesto de Trabajo.</p> <p>El comportamiento de los Pagos Adicionales puede ser fijo, asignado, proporcional, por tarifa y porciento, cada comportamiento tiene una ecuación para su cálculo, en el fijo es como su nombre lo indica un pago que no varía, en el asignado es un valor que se asigna y que al encontrarse un pago de este comportamiento se le pagará solo este valor al trabajador, en el proporcional se calcularía $\text{valor/horas laborables} \times \text{horas trabajadas}$ realmente, en el comportamiento por tarifa el valor será la tarifa de ese pago y la ecuación para el comportamiento de porciento sería $\text{valor}/100 \times (\text{salario escala} + \text{pago adicional1} + \text{pago adicional2} \dots)$.</p> <p>Luego se procede a indicar el elemento de gasto de acuerdo con los que están definidos en Costos y Procesos para los Pagos Adicionales.</p> <p>Al indicar el comportamiento si esta es por porciento se activara la ecuación que permite seleccionar los Pagos Adicionales, además del salario escala, necesarios para el cálculo del valor de ese pago.</p> <p>La fecha inicio es la fecha que en que se crea el pago adicional, el sistema la registra automáticamente.</p> <p>Validar la entrada de los datos.</p> <p>Si hubo errores en la entrada de los datos, notificar datos erróneos y permitir corregirlos.</p> <p>Si no registrar el pago adicional.</p>	
Validaciones	Validar los datos según el modelo conceptual: ERP- CH-Modelo Conceptual v1.8.doc.	

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

	Validar que la denominación no exista.
Post-condiciones	Se ha creado un nuevo pago adicional. Se ha asociado el elemento de gasto que le corresponda al pago adicional.
Post-requisito	No procede.

Adicionar Pago Adicional

Código:

Denominación:

Comportamiento:

Descripción:

Elementos del Gasto:

Valor:

Asociado A:

Fecha Efectiva:

Parámetros de la ecuación:

Salario Escala:

Código	Denominación	Comportamiento	Elementos del Gasto
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			

Cancelar Aplicar Aceptar

Figura 3. Prototipo adicionar Pagos Adicionales.

Tabla 2. Especificación del requisito Modificar Pago Adicional

Conceptos tratados	Conceptos	Atributos
--------------------	-----------	-----------

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

	Pago adicional.	Código, denominación, descripción, elemento del gasto, comportamiento, asociado a, fecha efectiva, valor, ecuación.
Precondiciones	Precondiciones	Pre-requisito
	Debe existir al menos un pago adicional en el sistema.	Adicionar pago adicional.
Descripción	<p>El sistema permite escoger el pago adicional que se desee modificar.</p> <p>Una vez seleccionada la opción modificar, se muestran los datos del pago adicional seleccionado, el sistema permitirá modificar los datos denominación, descripción, valor, fecha efectiva, asociado a, comportamiento, en caso de que el comportamiento sea por porcentaje se permitirá adicionar o modificar los datos de la ecuación y modificar la asociación del pago adicional con el elemento de gasto que lo proporciona Costo y Proceso.</p> <p>Una vez que se acepte la modificación de los datos, mostrar un mensaje de confirmación.</p> <p>Si se confirma se validan los datos.</p> <p>Si hubo errores en la entrada de los datos, notificarlos y permitir corregirlos.</p> <p>Si no se actualiza.</p>	
Validaciones	<p>Validar los datos según el modelo conceptual: ERP- CH-Modelo Conceptual v1.8.doc.</p> <p>Validar que la denominación no exista.</p> <p>Validar que los datos no se modifiquen mientras se esté procesando la nómina.</p>	
Post-condiciones	<p>Se han modificado los datos del pago adicional.</p> <p>Se ha modificado la relación del pago adicional si el usuario lo indica.</p>	

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

Post-requisito	No procede.
-----------------------	-------------

El prototipo muestra una ventana con el título "Modificar datos del pago". Dentro de la ventana, hay dos columnas de campos de entrada:

- Código:** un campo de texto con el valor "no se modifica".
- Denominación:** un campo de texto vacío.
- Comportamiento:** un menú desplegable.
- Descripción:** un campo de texto vacío.
- Elementos del gasto:** un menú desplegable.
- Valor:** un campo de texto vacío.
- Asociado a:** un menú desplegable.
- Fecha efectiva:** un menú desplegable.

En la parte inferior de la ventana, hay dos botones: "Cancelar" y "Aceptar".

Figura 4. Prototipo modificar Pagos Adicionales

Tabla 3. Especificación del requisito Eliminar Pago Adicional

Conceptos tratados	Conceptos	Atributos
	Pago adicional.	Fecha fin
Precondiciones	Precondiciones	Pre-requisito
	Debe existir al menos un pago adicional en el sistema.	Adicionar pago adicional.
Descripción	<p>El sistema permite seleccionar el pago adicional que desea eliminar.</p> <p>Al indicar la opción eliminar, el sistema muestra un mensaje de confirmación.</p> <p>Si se acepta se pone inactivo el pago adicional y se guarda la fecha en la cual paso a ser inactivo, siempre y cuando no haya pagos realizados por él, si no ponerlo inactivo y se guarda la fecha</p>	

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

	que paso a ser inactivo.
Validaciones	Validar que un pago adicional no se elimine si por este pago se han realizado pagos anteriormente.
Post-condiciones	Se registró la fecha de fin del pago adicional.
Post-requisito	No procede.

R2 Requisito Funcional Gestionar Puesto de Trabajo

El requisito funcional gestionar Puesto de Trabajo tiene en cuenta la manera de gestionar los Puestos de Trabajo en una entidad a través de adicionar, modificar y eliminar Puesto de Trabajo.

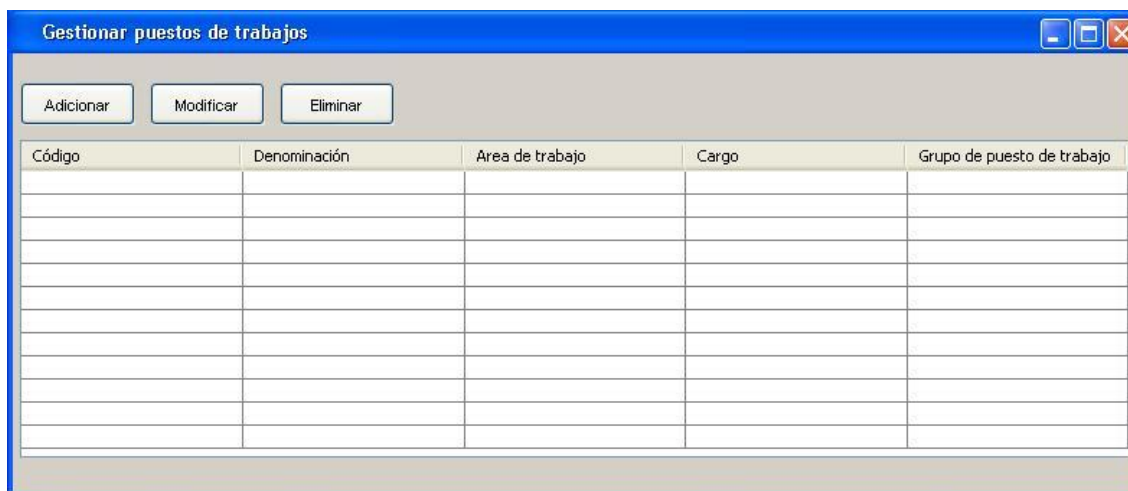


Figura 5. Prototipo gestionar Puesto de Trabajo

Tabla 4. Especificación del requisito Adicionar Puesto de Trabajo

Conceptos tratados	Conceptos	Atributos
	Puesto de Trabajo.	Área de trabajo, cargo, Grupo de Puestos de Trabajo, código, denominación, fecha efectiva, estado.
Precondiciones	Precondiciones	Pre-requisito
	Tiene que estar definido al menos un grupo de Puesto	Adicionar grupo de Puestos de Trabajo.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

	de Trabajo en el sistema.	
	Cargo	Denominación del cargo, categoría ocupacional, grupo de complejidad
	Grupo de complejidad.	Salario escala
Descripción	<p>Seleccionar el área de trabajo y el cargo que proporciona Estructura y Composición de acuerdo con la plantilla de la entidad, una vez que se seleccione el cargo, aparecerá automáticamente la cantidad de Puestos de Trabajo que tiene el cargo seleccionado en esa área, además la categoría ocupacional, el grupo de complejidad, el salario escala como información para el usuario.</p> <p>El sistema permite seleccionar el grupo de Puestos de Trabajo, agregar los datos código y denominación del Puesto de Trabajo, y seleccionar la fecha a partir de la cual será efectivo ese Puesto de Trabajo.</p> <p>El estado indica si el puesto está libre u ocupado, cuando se crea el Puesto de Trabajo este estará en estado libre y una vez que se haga el movimiento de nómina, entonces el puesto pasara a un estado ocupado.</p> <p>Al indicar la entrada de los datos el sistema los valida.</p> <p>Si hubo errores en la entrada, notificar datos erróneos y permitir corregirlos.</p> <p>registrar el Puesto de Trabajo.</p>	
Validaciones	<p>Validar los datos según el modelo conceptual: ERP- CH-Modelo Conceptual v1.8.doc.</p> <p>Validar que el código y la denominación no existan.</p> <p>Validar que la cantidad de Puestos de Trabajo que se inserten no exceda la cantidad de Puestos de Trabajo que tiene asignado el cargo escogido en esa área.</p>	
Post-condiciones	Se ha creado un nuevo Puesto de Trabajo.	
Post-requisito	No procede.	

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

Adicionar puesto de trabajo

Area de trabajo:

Grupo de puesto de trabajo:

Cargo:

Código:

Fecha efectiva:

Denominación:

Cancelar Aplicar Aceptar

Figura 6. Prototipo adicionar Puesto de Trabajo

Tabla 5. Especificación del requisito Modificar Puesto de Trabajo

Conceptos tratados	Conceptos	Atributos
	Puesto de Trabajo.	Grupo de Puestos de Trabajo, Denominación, fecha efectiva
	Movimiento de nómina	Numero consecutivo, tipo de movimiento, motivo de movimiento.
Precondiciones	Precondiciones	Pre-requisito
	Debe existir al menos un Puesto de Trabajo en el sistema.	Adicionar Puesto de Trabajo
Descripción	<p>Seleccionar el Puesto de Trabajo que se desea modificar.</p> <p>Al indicar la opción modificar, se muestran los datos del Puesto de Trabajo seleccionado, además se le adicionan los datos número</p>	

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

	<p>consecutivo, tipo de movimiento y el motivo del movimiento, estos últimos datos se mostrarán en caso de que el Puesto de Trabajo esté asociado a un trabajador y le cambiaran el grupo de Puestos de Trabajo que tiene asignado, si este pertenece a un cargo diferente.</p> <p>El sistema debe permitir modificar la denominación, la fecha efectivo y el grupo de Puestos de Trabajo al que pertenece.</p> <p>Si se confirma la modificación se validan los datos.</p> <p>Si hubo errores en los datos, notificarlos y permitir corregirlos.</p> <p>Si no se guardan los cambios y se actualiza.</p>
Validaciones	<p>Validar los datos según el modelo conceptual: ERP- CH-Modelo Conceptual v1.7.doc.</p> <p>Validar que los datos no se modifiquen mientras se este procesando la nómina.</p> <p>Validar que la denominación y el número consecutivo introducidos no existan.</p>
Post-condiciones	<p>Se han modificado los datos del Puesto de Trabajo.</p> <p>Se ha modificado la relación del Puesto de Trabajo si el usuario lo ha indicado.</p>
Post-requisito	No procede.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

El prototipo de la ventana 'Modificar puesto de trabajo' presenta los siguientes elementos:

- Área de trabajo:** Campo de selección con una flecha hacia abajo.
- Grupo de puesto de trabajo:** Campo de selección con una flecha hacia abajo.
- Cargo:** Campo de selección con una flecha hacia abajo.
- Denominación:** Campo de texto libre.
- Fecha efectiva:** Campo de selección con una flecha hacia abajo.
- Botones:** 'Cancelar' y 'Aceptar' situados en la parte inferior derecha.

Figura 7. Prototipo modificar Puesto de Trabajo

Tabla 6. Especificación del requisito Eliminar Puesto de Trabajo

Conceptos tratados	Conceptos	Atributos
	Puesto de Trabajo.	Área de trabajo, cargo, Grupo de Puestos de Trabajo, código, denominación, fecha, efectiva, estado, fecha fin.
Precondiciones	Precondiciones	Pre-requisito
	Debe existir al menos un Puesto de Trabajo en el sistema.	Adicionar Puesto de Trabajo.
Descripción	<p>Seleccionar el Puesto de Trabajo que se desea eliminar.</p> <p>Al indicar la opción eliminar, el sistema muestra un mensaje de confirmación.</p> <p>Si se acepta se elimina el Puesto de Trabajo y se guarda la fecha que fue eliminado.</p>	
Validaciones	Validar los datos según el modelo conceptual: ERP- CH-Modelo	

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

	Conceptual v1.8.doc. Validar que el puesto no se elimine mientras se este procesando la nómina. Validar que el puesto esté vacante para poder eliminarlo.
Post-condiciones	Se ha eliminado el Puesto de Trabajo.
Post-requisito	No procede.

R3 Requisito Funcional Gestionar Grupos de Puestos de Trabajo

Los grupos de Puestos de Trabajo constituyen un concepto para registrar los Puestos de Trabajo que tienen características semejantes, esto constituye un ventaja ya que los Puestos de Trabajo se organizan por grupos de puestos y de esta forma los pagos adicionales se pueden aplicar a los Grupos de Puestos de Trabajo ahorrándose el tener que adicionarlo puesto por puesto.

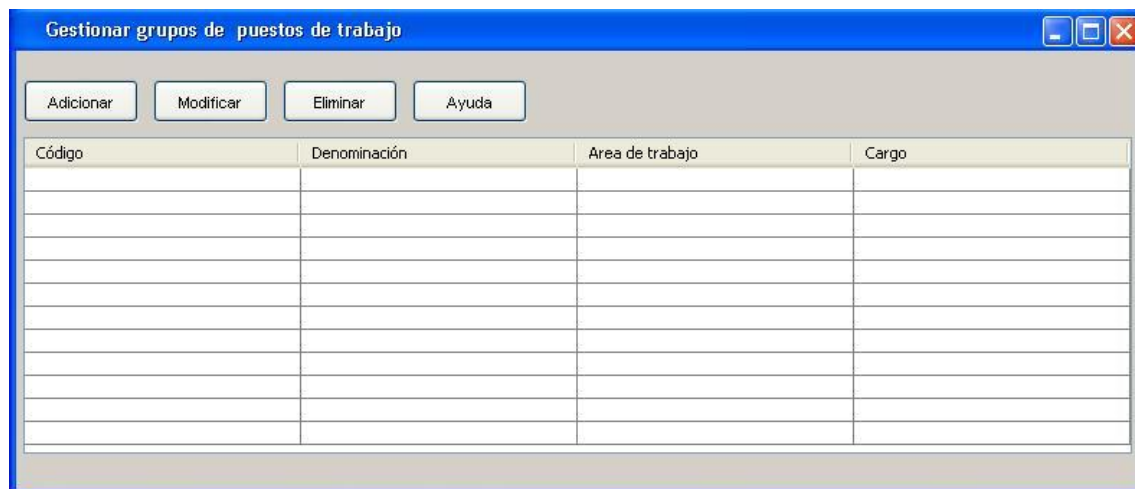


Figura 8. Prototipo gestionar grupo de Puesto de Trabajo

Tabla 7. Especificación del requisito Adicionar grupos de Puestos de Trabajo

Conceptos tratados	Conceptos	Atributos
	Grupos de Puestos de Trabajo.	Área de trabajo, cargo, código, denominación, jornada laboral, fondo de tiempo, fecha efectivo, fecha efectiva.
	Cargo	Denominación del cargo, categoría

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

		ocupacional, grupo de complejidad
	Grupo de complejidad.	Salario escala
Precondiciones	Precondiciones	Pre-requisito
	Tiene que estar definida la estructura de la entidad (áreas de trabajo y cargos).	Adicionar cargos. Adicionar área de trabajo.
Descripción	<p>Seleccionar el área de trabajo y el cargo que lo proporciona Estructura y Composición, al seleccionar el cargo debe mostrarse automáticamente la cantidad de Puestos de Trabajo que tiene ese cargo en el área escogida, la categoría ocupacional, el grupo de complejidad y el salario escala de este cargo.</p> <p>El sistema permite agregar los datos código, denominación, jornada laboral (horas), fondo de tiempo (horas) e indicar la fecha efectivo.</p> <p>La fecha inicio es la fecha en la cual se crea el grupo de Puestos de Trabajo.</p> <p>Al aceptar la entrada de los datos el sistema los valida.</p> <p>Si hubo errores en la entrada, notificar datos erróneos y permitir corregirlos.</p> <p>registrar el grupo de Puestos de Trabajo.</p> <p>Una vez que esté registrado el grupo de Puestos de Trabajo, se le asocian los Pagos Adicionales si los tiene (Ver requisito Asociar pago adicional a grupo de Puestos de Trabajo).</p>	
Validaciones	<p>Validar los datos según el modelo conceptual: ERP- CH-Modelo Conceptual v1.8.doc.</p> <p>Validar que la denominación y el código no existan.</p>	
Post-condiciones	<p>Se ha creado un nuevo grupo de Puestos de Trabajo.</p> <p>Se ha asociado al grupo de Puestos de Trabajo los Pagos Adicionales si los tiene.</p>	
Post-requisito	No procede.	

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

Adicionar grupo de puesto de trabajo

Area de trabajo: [Campo de selección con flecha hacia abajo]

Cargo: [Campo de selección con flecha hacia abajo]

Código: TextField3

Fondo de tiempo: TextField

Jornada laboral: TextField2

Fecha efectiva: [Campo de selección con flecha hacia abajo]

Denominación: [Campo de texto grande]

Seleccione pagos adicionales

Pago adicional	Valor
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	

Cancelar Aplicar Aceptar

Figura 9. Prototipo adicionar grupo de Puesto de Trabajo

Tabla 8. Especificación del requisito Modificar grupo de Puestos de Trabajo

Conceptos tratados	Conceptos	Atributos
	Grupo de Puestos de Trabajo.	Denominación, jornada laboral, fondo de tiempo, fecha efectiva
	Movimiento de nómina.	Número consecutivo, tipo de movimiento, motivo de movimiento

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

Precondiciones	Precondiciones	Pre-requisito
	Debe existir al menos un grupo de Puestos de Trabajo en el sistema.	Adicionar grupo de Puestos de Trabajo.
Descripción	<p>Seleccionar el grupo de Puestos de Trabajo que se desea modificar.</p> <p>Al indicar la opción modificar, se muestran los datos del grupo de Puestos de Trabajo seleccionado, además se le adicionan los datos número consecutivo, tipo de movimiento y el motivo del movimiento, estos últimos datos se mostrarán en caso de que el grupo de Puestos de Trabajo tenga al menos un Puesto de Trabajo asociado a un trabajador o se le modifiquen los datos área o cargo.</p> <p>El sistema debe permitir modificar el área, el cargo, la denominación, la jornada laboral, el fondo de tiempo, fecha efectivo y los Pagos Adicionales que tiene asociado ese grupo de Puestos de Trabajo (Ver requisito Asociar pago adicional a grupo de Puestos de Trabajo).</p> <p>Si se confirma la modificación, se validan los datos.</p> <p>Si hubo errores en los datos, notificarlos y permitir corregirlos.</p> <p>Si no, se guardan los cambios y se actualiza.</p>	
Validaciones	<p>Validar los datos según el modelo conceptual: ERP- CH-Modelo Conceptual v1.8.doc.</p> <p>Validar que los datos no se modifiquen mientras se esté procesando la nómina.</p> <p>Validar que la denominación y el código consecutivo no existan.</p>	
Post-condiciones	<p>Se han modificado los datos del grupo de Puestos de Trabajo.</p> <p>Se ha modificado la relación del grupo de Puestos de Trabajo con los Pagos Adicionales si el usuario lo ha indicado.</p>	
Post-requisito	No procede.	

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

El prototipo de la ventana 'Modificar grupo de puesto de trabajo' presenta los siguientes campos y controles:

- Area de trabajo:** Selector de lista desplegable.
- Cargo:** Selector de lista desplegable.
- Denominación:** Campo de texto libre.
- Fondo de tiempo:** Campo de texto libre.
- Jornada laboral:** Campo de texto libre.
- Fecha efectiva:** Selector de lista desplegable.
- Datos de movimiento de nómina:**
 - Id de movimiento:** Campo de texto libre.
 - Tipo de movimiento:** Selector de lista desplegable.
 - Motivo de movimie...:** Selector de lista desplegable.
- Seleccione pagos adicionales:** Tabla con columnas 'Pago adicional' y 'Valor', y tres filas con casillas de verificación.
- Botones:** 'Cancelar', 'Aplicar' y 'Aceptar'.

Figura 10. Prototipo modificar grupo de Puesto de Trabajo

Tabla 9. Especificación del requisito Eliminar grupo de Puestos de Trabajo

Conceptos tratados	Conceptos	Atributos
	Grupo de Puestos de Trabajo.	Fecha fin
Precondiciones	Precondiciones	Pre-requisito
	Debe existir al menos un grupo de Puestos de Trabajo en el sistema.	Adicionar grupo de Puestos de Trabajo.
Descripción	Seleccionar el grupo de Puestos de Trabajo que se desea	

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

	<p>eliminar.</p> <p>Al indicar la opción eliminar, el sistema muestra un mensaje de confirmación.</p> <p>Si se indica se elimina el grupo de Puestos de Trabajo y se guarda la fecha que fue eliminado.</p>
Validaciones	<p>Validar los datos según el modelo conceptual: ERP- CH-Modelo Conceptual v1.8.doc</p> <p>Validar que el grupo de Puestos de Trabajo no se elimine mientras se esté procesando la nómina.</p> <p>Validar que el grupo de Puestos de Trabajo esté vacío para poder eliminarlo.</p> <p>Validar que una vez que se elimine el grupo de Puestos de Trabajo se eliminen también las asociaciones que tiene este con los Pagos Adicionales.</p>
Post-condiciones	<p>Se ha eliminado el grupo de Puestos de Trabajo si no tiene Puestos de Trabajo asociados.</p> <p>Se ha registrado la fecha de fin del grupo de Puestos de Trabajo si tiene puestos asociados.</p>
Post-requisito	No procede.

Tabla 10. Requisitos Aprobados por:

Nombre	Cargo	Firma	Fecha
Dr. José Carlos Del Toro Ríos	Director Nacional del proyecto ERP-Cuba		
CP. Crescencio Campos Hung	Especialista Funcional		
Ing. Rolando Ramírez Concepción	Representante UCID		
Ing. Ileana Alicia Corrales Tabasco	Representante del		

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

	MAC		
Ing. Alain Fernández Deronceré	Jefe de Línea		
Ing. Yanet Vega Miniet	Analista Principal		

2.6 Requisitos no Funcionales

Los requerimientos no funcionales: Son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Existen múltiples categorías para clasificar a los requerimientos no funcionales, siendo las siguientes representativas de un conjunto de aspectos que se deben tener en cuenta, aunque no limitan a la definición de otros.

- **Requerimientos de Software**
- **Requerimientos de Hardware**
- **Restricciones en el diseño y la implementación**
- **Requerimientos de apariencia o interfaz externa**
- **Requerimientos de Usabilidad**
- **Requerimientos de Rendimiento**
- **Requerimientos Legales**
- **Requerimientos de Portabilidad**
- **Requerimientos Políticos culturales**
- **Requerimientos de Seguridad**

La seguridad puede ser tratada en tres aspectos diferentes:

Confidencialidad: La información manejada por el sistema esta protegida de acceso no autorizado y divulgación.

Integridad: La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos.

Disponibilidad: A los usuarios autorizados se les garantizarán el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

Requisitos no funcionales de la solución propuesta:

➤ **Apariencia o interfaz externa:**

El sistema debe tener una interfaz fácil de usar y amigable para que pueda ser utilizada sin mucho entrenamiento por el usuario.

Empleo de imágenes identificadas con el negocio donde se implantará el sistema y colores agradables a la vista, siendo éstos claros.

➤ **Usabilidad:**

El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora.

➤ **Rendimiento:**

Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos, no mayores de 5 segundos para las actualizaciones y 20 para las recuperaciones.

➤ **Portabilidad:**

El sistema debe ser multiplataforma, importante el correcto funcionamiento en Linux y Windows.

➤ **Seguridad:**

Autenticación (Contraseña de acceso).

Autorización (Atribución a los usuarios respecto a sus funciones de trabajo).

Implementación de auditoria (Registrar la confirmación de cada operación efectuada por el usuario que afecte los registros contables).

La atención al sistema incluyendo, el mantenimiento de las bases de datos así como la salva de la información se realizarán de forma centralizada por el administrador.

➤ **Políticos culturales:**

El sistema solo podrá ser utilizado en territorio cubano y por las entidades autorizadas por el Ministerio de las FAR.

El producto no debe contener palabras en otros idiomas.

El producto debe respetar los términos empleados normalmente por los especialistas en el tema de la esfera que se automatiza.

➤ **Legales:**

El sistema estará avalado por los tres documentos rectores emitidos en el país para la certificación y validación de los sistemas contables:

La Resolución Conjunta de los ministerios de Finanzas y Precios de fecha 8.04.04.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

La Resolución 340 del Ministerio de Finanzas y Precios de fecha 8.12.04.

La Resolución No. 12 del Ministerio de la Informática y las Comunicaciones de fecha 24.01.05.

➤ **Software:**

Para el cliente:

- Navegador Mozilla Firefox.
- Sistema operativo Linux, Windows 98 o superior.
- Para el servidor:
- Sistema operativo Windows Advancer Server (2000 o superior) o Linux en cualquiera de sus distribuciones.
- Un servidor Apache 2.0 o superior con módulo PHP 5.0 disponible, este debe estar configurado con la extensión "pgsql" incluida.
- Un servidor de base de datos PostgreSQL 8.0 o superior.

➤ **Hardware:**

Para el servidor:

- Requerimientos mínimos: Procesador Pentium III a 1GHz de velocidad de procesamiento y 1Gb de memoria RAM (Memoria de Acceso Aleatorio de sus siglas en ingles Random Access Memory).
- Al menos 40Gb de espacio libre en disco duro.
- Tarjeta de red.

Para el cliente:

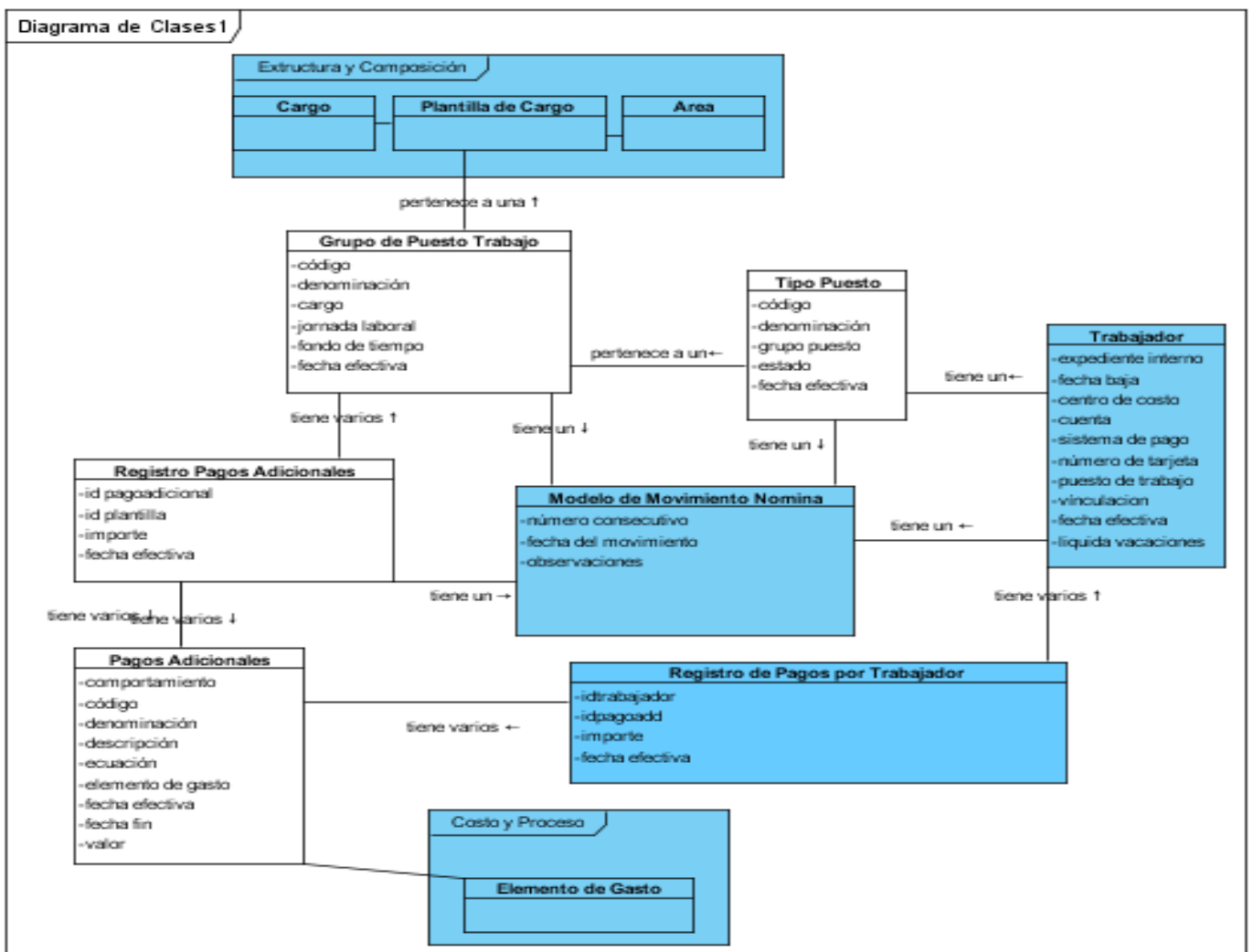
- Requerimientos mínimos: Procesador Pentium II a 133Mhz con 128 Mb de memoria RAM.
- Tarjeta de red.
- **Restricciones para el diseño e implementación:**
 - Emplear como servidores Web y de bases de datos Apache y PostgreSQL respectivamente.
 - Utilizar como lenguaje del lado del servidor al PHP 5.0 o superior y del lado del cliente el JavaScript.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

2.7 Modelo conceptual

El modelo conceptual es una representación de conceptos en un dominio del problema. Además este modelo muestra asociaciones entre conceptos y atributos de conceptos. Se puede ver como un modelo que comunica cuales son los términos importantes y como se relacionan entre sí.

En el siguiente diagrama correspondiente al modelo conceptual, los recuadros con fondo azul representan las clases externas (pertenecientes a otros subsistemas o a otros componentes del mismo subsistema) a los componentes Pagos Adicionales y Puesto de Trabajo y la integración de estas con los componentes tratados en la solución que son representados con los recuadros de fondo blanco con sus atributos.



CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

Figura 11. Modelo conceptual

2.8 Modelo de diseño

El Modelo de diseño es una abstracción del Modelo de Implementación y su código fuente, el cual fundamentalmente se emplea para representar y documentar su diseño. Es usado como entrada esencial en las actividades relacionadas a implementación. El Modelo de Diseño puede contener: los diagramas, las clases, paquetes, subsistemas, capsulas, protocolos, interfaces, relaciones, colaboraciones y atributos. (40)

Teniendo en cuenta la propuesta de solución realizada anteriormente, es necesario definir como se desarrollará el sistema. Se da paso así la exposición de los resultados del flujo de trabajo de diseño, que refleja cómo será implementado el sistema en términos de clases del diseño. Se presentan como artefactos los diagramas de clases del diseño para cada uno de los paquetes de requisitos identificados, donde se ubican las funcionalidades a desarrollar, todo esto basándose en el uso de patrones de diseño que fueron definidos y en los estándares que evalúan el diseño definidos por el proyecto ERP-Cuba para la realización del sistema CEDRUX, reflejándose el uso de herramientas y tecnologías explicadas en el CAPÍTULO 1; para concluir con el diseño se hace una evaluación del mismo.

2.8.1 Patrones de Diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias. (41)

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

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

- Soluciones concretas: proponen soluciones a problemas concretos, son teorías genéricas.
- Soluciones técnicas: indican soluciones basadas en programación orientada a objetos (OO).
- Se utilizan en situaciones frecuentes: se basan en la experiencia acumulada por los que resuelven problemas reiterativos.
- No se refleja en el código su uso.

Se utilizaron fundamentalmente tres tipos de patrones:

- Patrón Arquitectónico Modelo-Vista-Controlador (MVC).
- Patrones GRASP.
- Patrones de Comportamiento.

2.8.1.1 Patrón Arquitectónico Modelo-Vista-Controlador (MVC) (42)

MVC es un patrón de diseño que considera dividir una aplicación en tres módulos claramente identificables y con funcionalidad bien definida: El Modelo, las Vistas y el Controlador.

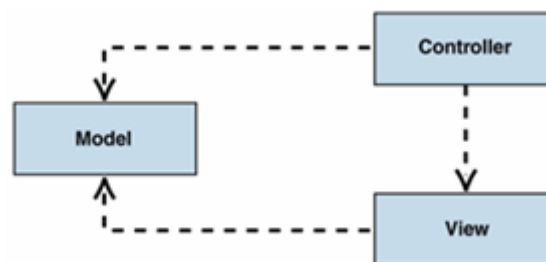


Figura 12. Patrón Arquitectónico Modelo-Vista-Controlador (MVC)

El patrón conocido como Modelo-Vista-Controlador (MVC) separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes:

Modelo: El modelo es un conjunto de clases que representan la información del mundo real que el sistema debe procesar, sin tomar en cuenta ni la forma en la que esa

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

información va a ser mostrada ni los mecanismos que hacen que esos datos estén dentro del modelo, es decir, sin tener relación con ninguna otra entidad dentro de la aplicación.

Controlador: El controlador es un objeto que se encarga de dirigir el flujo del control de la aplicación debido a mensajes externos, como datos introducidos por el usuario u opciones del menú seleccionadas por él. A partir de estos mensajes, el controlador se encarga de modificar el modelo o de abrir y cerrar vistas. El controlador tiene acceso al modelo y a las vistas, pero las vistas y el modelo no conocen de la existencia del controlador.

Vista: Las vistas son el conjunto de clases que se encargan de mostrar al usuario la información contenida en el modelo. Una vista está asociada a un modelo, pudiendo existir varias vistas asociadas al mismo modelo. Una vista obtiene del modelo solamente la información que necesita para desplegar y se actualiza cada vez que el modelo del dominio cambia por medio de notificaciones generadas por el modelo de la aplicación.

Desarrollar una aplicación siguiendo este patrón de diseño tiene muchas ventajas:

- La aplicación está implementada modularmente.
- Sus vistas muestran información actualizada siempre.
- El programador no debe preocuparse de solicitar que las vistas se actualicen, ya que este proceso es realizado automáticamente por el modelo de la aplicación.
- Si se desea hacer una modificación al modelo del dominio, como aumentar métodos o datos contenidos, solo debe modificarse el modelo y las interfaces del mismo con las vistas, no todo el mecanismo de comunicación y de actualización entre modelos.
- Las modificaciones a las vistas no afectan en absoluto a los otros módulos de la aplicación.
- MVC es bastante utilizado en la actualidad en marcos de aplicación orientados a objeto desarrollados para construir aplicaciones de gran tamaño; Java Swing, Apache Struts, Microsoft ASP.NET, las transformaciones XSL o incluso los documentos LATEX siguen este patrón de diseño.
- MVC está demostrando ser un patrón de diseño bien elaborado pues las aplicaciones que lo implementan presentan una extensibilidad y una mantenibilidad únicas comparadas con otras aplicaciones basadas en otros patrones.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

El Patrón Arquitectónico Modelo-Vista-Controlador (MVC) fue utilizado en la solución de la siguiente forma:

En las clases del modelo representadas en los diagramas de clases se representa la lógica de negocio. La vista transforma el modelo en una página Web que permite al usuario interactuar con ella. El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

Ejemplo: Un usuario realiza una petición a la clase controladora `GestionarpagosadicionalesController.php` la cual procesa la información y solicita los datos en la clase modelo, ejemplo `NomPagosadicionaleModel.php` que se encarga de realizar las operaciones pertinentes en la base de datos y enviar los datos pedidos a la clase controladora donde son capturados mediante una url en la vista `gestionarpagosadicionales.js` donde se le da la respuesta al usuario.

2.8.1.2 Patrones GRASP

Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. (43)

Alta Cohesión: La cohesión es una medida de la fuerza con la que se relacionan las clases y el grado de focalización de las responsabilidades de un elemento, Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable, una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo. (44)

La Alta Cohesión se pone de manifiesto por que se asignaron responsabilidades a las clases de manera que todos sus métodos tuvieran un comportamiento bien definido, por ejemplo: la responsabilidad de la clase `GestionarpagosadicionalesController` es la de acceder a los valores y métodos de las clases `NomPagosadicionale` y `NomPagosadicionaleModel`, de manera que en la clase `GestionarpagosadicionalesController` no existe ningún método que sea para acceder a la clase `gestionarpagosadicionales.js`, a no ser que estén estrechamente relacionadas.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

Bajo Acoplamiento: El acoplamiento es una medida de fuerza con que un elemento esta a, tiene conocimiento de, confía en, otros elementos. Este patrón es un principio que asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema. (44)

El uso de este patrón se pone de manifiesto en la solución al estar cada clase acoplada a las clases estrictamente necesarias, por ejemplo:

La clase GestionarpuestodetrabajoController realiza una integración con el componente Trabajador y no conoce la existencia de la clase NomTipomovnomina del componente Trabajador que es la que contiene los datos que ella pide, mediante el uso del IOC para obtener los datos o colección de datos que se necesiten en el componente Puesto de Trabajo logrando una escasa dependencia entre las clases y un aumento en la reutilización.

2.8.1.3 Patrones de comportamiento

Los patrones de comportamiento estudian las relaciones entre llamadas entre los diferentes objetos, normalmente ligados con la dimensión temporal. (45)

Los patrones de comportamiento estudian las relaciones entre llamadas entre los diferentes objetos, normalmente ligados con la dimensión temporal.

Debido a sus funciones se destacan para la solución los siguientes patrones de comportamiento:

Cadena de responsabilidad

Permite la separación entre objetos, pasando la tarea de un objeto al siguiente dentro de una cadena hasta que la tarea es reconocida. (46)

El patrón cadena de responsabilidad se reflejó mediante el tratamiento de errores puesto que a través del lanzamiento de excepciones de las clases y las capturas de las mismas por el aspecto gestor global de excepciones (managerexception) junto al identificador de las mismas (exception), se manifiesta que el manejador específico que satisfará la

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

petición, no se conoce “a priori” y se quiere trasladar dicha petición a un objeto de entre unos cuantos, sin especificar explícitamente el receptor final.

Estrategia

Encapsula un algoritmo dentro de una clase. (46) El uso de este patrón se evidencia en que todas definen numerosos comportamientos, los cuales se manifiestan como definiciones condicionales múltiples de sus operaciones, además de que los algoritmos usan datos que los clientes no tienen porque conocer.

2.8.1.4 Patrones estructurales

Fachada

Simplifica los accesos a un conjunto de objetos relacionados, funciona como intermediario entre uno y otro grupo objetos, se utiliza para comunicar estos dos grupos.

Tipo: Estructura, a nivel de objetos.

Propósito: Simplificar el acceso a un conjunto de clases o interfaces. Proporcionar una interfaz unificada de alto nivel que representa a todo un subsistema facilitando su uso. La “fachada” satisface a la mayoría de los clientes, sin ocultar las funciones de menor nivel a aquellos que necesiten acceder a ellas.

Motivación: Reducir la dependencia entre clases Fachada ofrece un punto de acceso al resto de clases si estas cambian o se sustituyen por otras solo hay que actualizar la clase Fachada sin que el cambio afecte a las aplicaciones cliente, No oculta las clases sino que ofrece una forma más sencilla de acceder a ellas, en los casos en que se requiere se puede acceder directamente a ellas.

En el desarrollo de esta solución la utilización del patrón estructural Fachada se ve en que cada componente tiene un paquete denominado servicios donde se encuentran las clases que en este caso serían PagosAdicionalesService.php y PuestoTrabajoService.php estas clases contiene los servicios que brindan estos componentes a otros a través del IOC interno a los componentes internos (del mismo subsistema) y a través del IOC externo a otros componentes de otros subsistemas. La Fachada sería entonces el IOC tanto interno como externo, ya que funciona como intermediario para la comunicación entre componentes.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

2.8.2 Diagramas de clases del diseño

Los diagramas de clases del diseño describen gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación, contiene información como. Clases, asociaciones y atributos, interfaces con sus operaciones y constantes, métodos, navegabilidad, dependencias. (49)

Se realizó un diagrama de clases del diseño por cada paquete de requisitos.

En los diagramas de clases a continuación se pueden observar como quedan definidas las clases con todas sus funcionalidades y como se relacionan entre ellas, con otras clases de otros componentes, subsistemas externos y con algunas de las tecnologías usadas, se evidencia en parte como se usaron los patrones de diseño anteriormente explicados. Las clases cuyo nombre termina en Controller se refieren a las clases controladoras que son las que contienen los atributos y los métodos necesarios y se encargan de seleccionar que clase modelo (las que el nombre termina en Model) es la adecuada para realizar la acción encomendada por el usuario, o sea Insertar, actualizar (modificar) o eliminar. Las relaciones o integración entre componentes antes mencionadas se llevan a cabo a través del IOC que se encuentra en el Zend_Ext Framework descrito en el CAPÍTULO 1. El IOC no es más que un fichero de extensión XML que será mapeado por el framework antes mencionado para su propio funcionamiento, el IOC registra las funcionalidades que ofrecen los métodos de las clases control de los componentes del sistema para otros componentes, especifica respuestas deseadas a sucesos o solicitudes de datos concretas, dejando que otro módulo o componente lleve a cabo las acciones de control que se requieran en el orden necesario y para el conjunto de sucesos que tengan que ocurrir.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

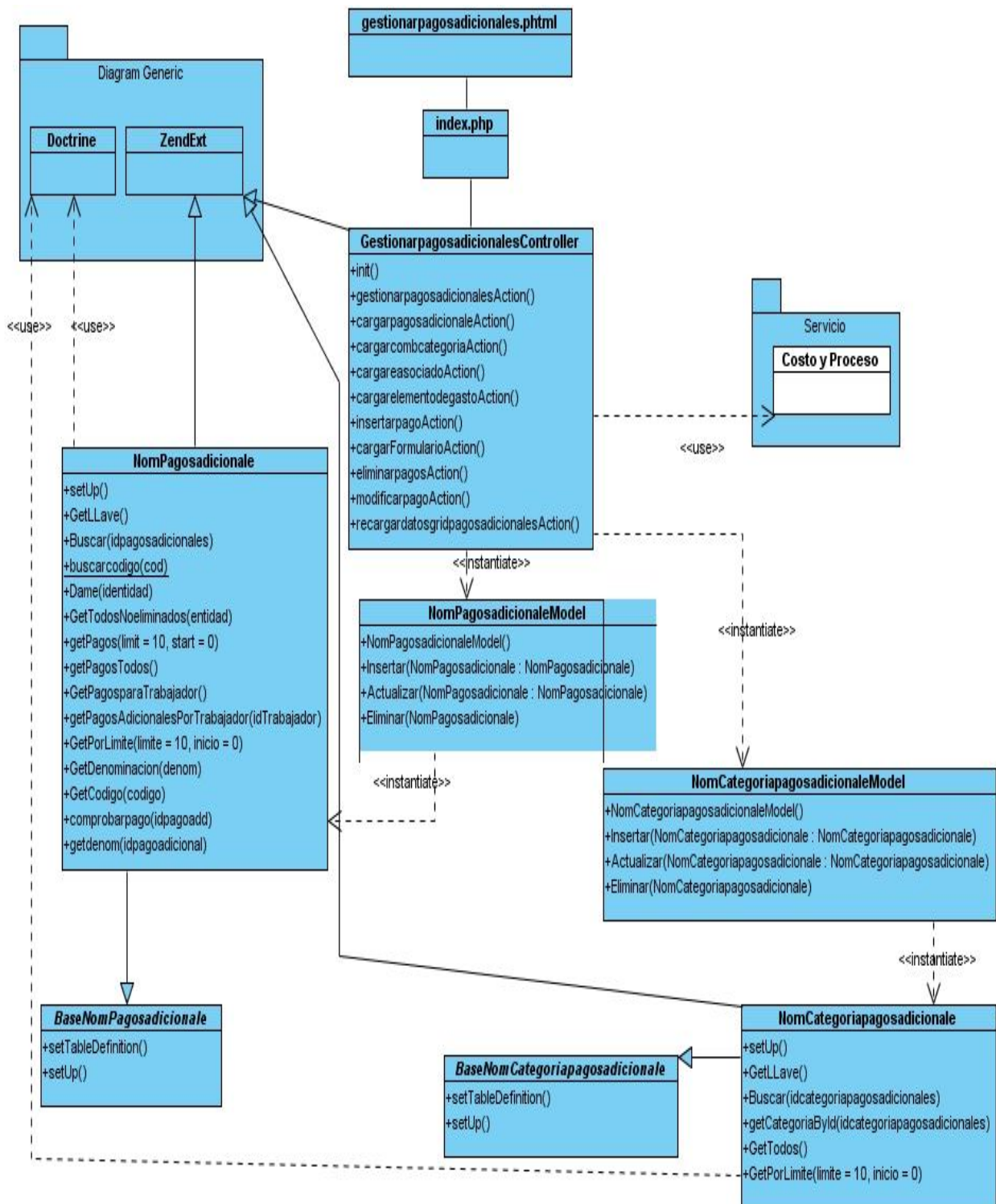


Figura 13. Diagrama de clases gestionar Pagos Adicionales

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

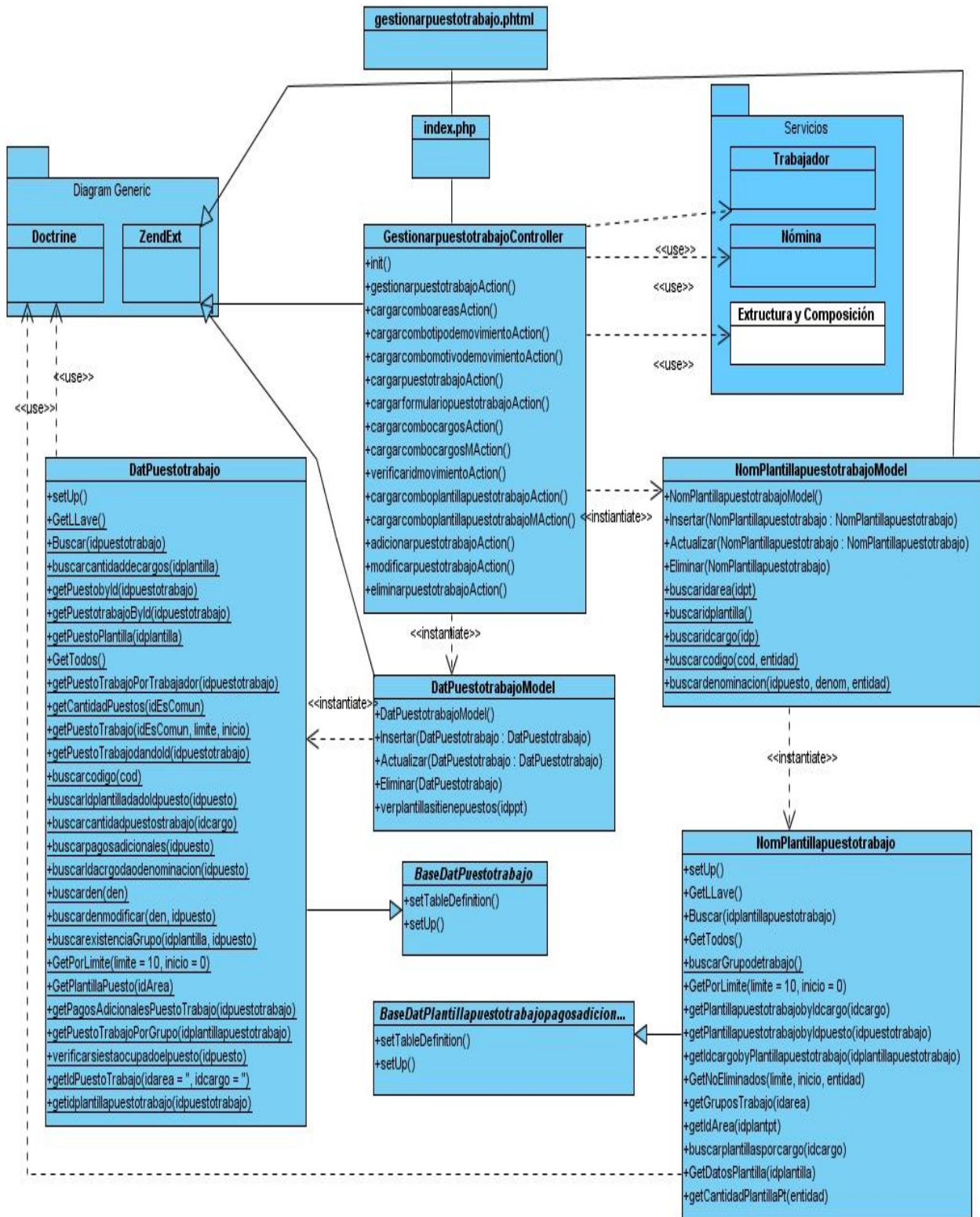


Figura 14. Diagrama de clases del diseño gestionar Puesto de Trabajo

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

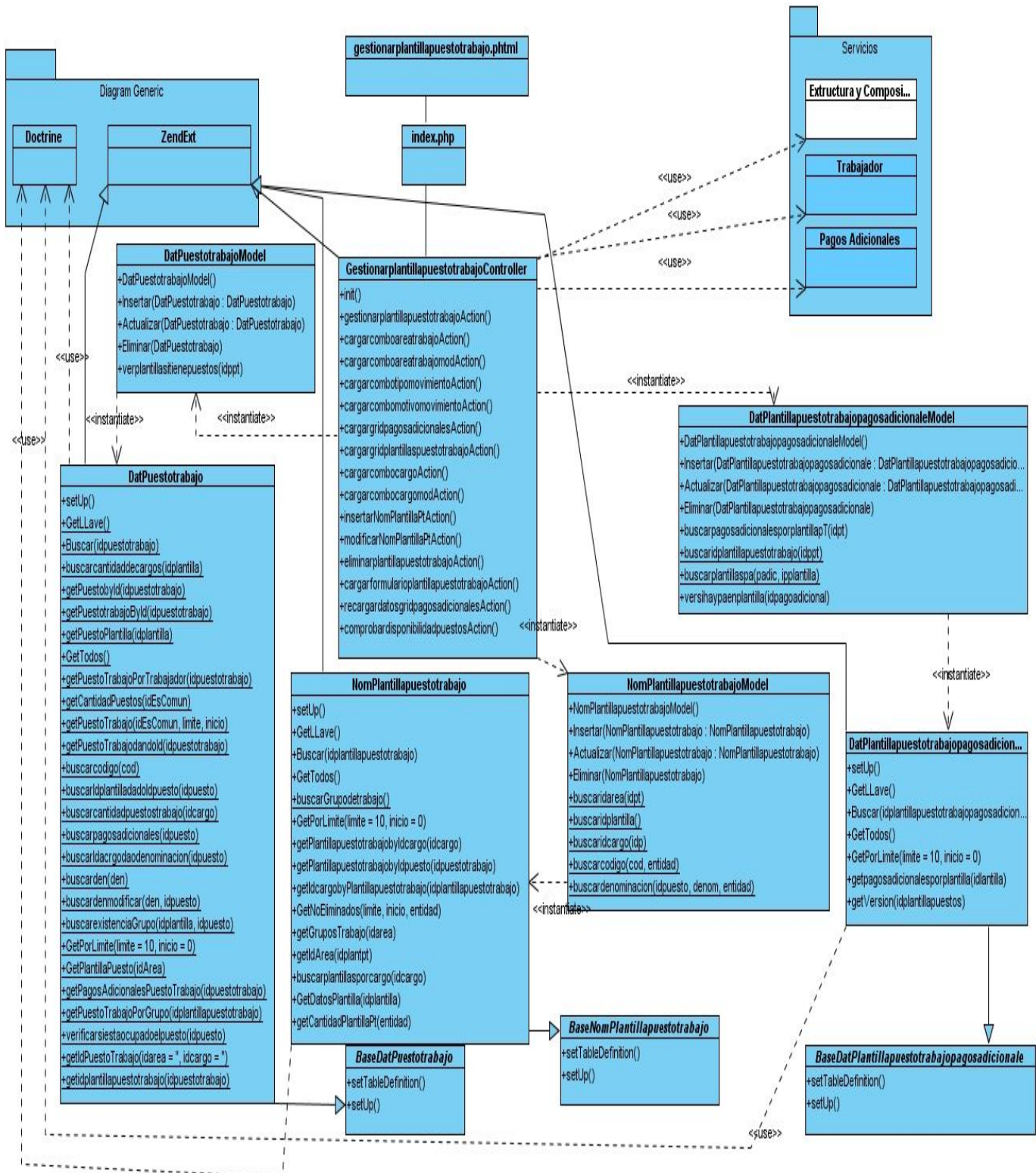


Figura 15. Diagrama de clases gestionar grupo de Puesto de Trabajo

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

2.9 Diseño de la Base de Datos

Uno de los pasos cruciales en la construcción de una aplicación que maneje una base de datos, es sin duda el diseño de la base de datos. En la base de datos que se utiliza en los componentes que se están desarrollando como propuesta de solución, para lograr el acceso eficiente a la información con redundancia mínima, se toman varias consideraciones, entre las que se encuentran: la velocidad y facilidad de acceso a la información para extraerla.

2.9.1 Modelo de datos

El modelo de datos representa los datos en el más bajo nivel y permite identificar algunos detalles de implantación para el manejo del hardware de almacenamiento, proporciona referencias de cómo se almacenan los datos en la computadora, el formato de los registros, la estructura de los ficheros (ordenados, desordenados) y los métodos de acceso utilizados (índices). Los conceptos de este modelo están dirigidos fundamentalmente al personal informático, no a los usuarios finales.

Teniendo en cuenta la extensión de este modelo se puede apreciar en el [\(anexo 1\)](#).

La descripción de las tablas usadas en el modelo de datos se puede apreciar en el [\(anexo 2\)](#).

2.10 Evaluación del modelo de diseño propuesto

Como objetivo de este trabajo en este epígrafe se realiza una evaluación del modelo de diseño propuesto, para ello se realizó en una primera fase un mapeo entre las características funcionales asociados a los procesos de los componentes Puestos de Trabajo y Pagos Adicionales y el diseño de clases construidos para dar solución a estos requisitos funcionales. De este modo se garantizó el chequeo del cumplimiento del atributo interno de calidad más importante que plantea la norma ISO 9126 (ISO/IEC, 2001), la Funcionalidad, que consiste en la capacidad del software de proveer las funciones que cumplen con las necesidades implícitas y explícitas cuando el mismo es utilizado bajo ciertas condiciones. (50) Demostrándose que las características funcionales más importantes definidas para ambos módulos han sido cubierta en el diseño propuesto.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

Componente Pagos Adicionales

Tabla 11. Requisitos asociados a Pagos Adicionales

Proceso	Requerimiento funcional asociado al proceso.	Clase del diseño que lo cubre.	Método de la clase que lo cubre.
Gestionar Pagos Adicionales	Adicionar Pago Adicional	NomPagosadicionaleModel	Insertar(NomPagosadicionale \$NomPagosadicionale)
	Modificar Pago Adicional	NomPagosadicionaleModel	Actualizar(NomPagosadicionale \$NomPagosadicionale)
	Eliminar Pago Adicional	NomPagosadicionaleModel	Eliminar(\$NomPagosadicionale)

Componente Puesto de Trabajo

Tabla 12. Requisitos asociados a Puestos de Trabajo

Proceso	Requerimiento funcional asociado al proceso.	Clase del diseño que lo cubre.	Método de la clase que lo cubre.
Gestionar Puesto de Trabajo	Adicionar Puesto de Trabajo	DatPuestotrabajoModel	Insertar(DatPuestotrabajo \$DatPuestotrabajo)
	Modificar	DatPuestotrabajoModel	Actualizar(DatPuestotrabajo \$d)

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

	Puesto de Trabajo		
	Eliminar Puesto de Trabajo	DatPuestotrabajoModel	Eliminar(\$DatPuestotrabajo)
Gestionar Grupo de Puesto de Trabajo	Adicionar Grupo de Puesto de Trabajo	NomPlantillapuestotrabajo Model	Insertar(NomPlantillapuestotrabajo o \$NomPlantillapuestotrabajo)
	Modificar Grupo de Puesto de Trabajo	NomPlantillapuestotrabajo Model	Actualizar(NomPlantillapuestotrabajo \$NomPlantillapuestotrabajo)
	Eliminar Grupo de Puesto de Trabajo	NomPlantillapuestotrabajo Model	Eliminar(\$NomPlantillapuestotrabajo)

Otro aspecto importante a tener en cuenta en la evaluación de la calidad del diseño ha sido la creación de métricas básicas basada en el estudio de la calidad del diseño orientado a objeto realizado por Pressman en (51); teniendo en cuenta que este estudio brinda un esquema sencillo de implementar y que a la vez cubre los principales atributos de calidad de software.

Atributos de calidad que se abarcan:

- Responsabilidad. Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.
- Complejidad del diseño. Consiste en la complejidad que posee una estructura de diseño de clases.
- Complejidad de implementación. Consiste en el grado de dificultad que tiene implementar un diseño de clases determinado.
- Reutilización. Consiste en el grado de reutilización de presente en una clase o estructura de clase, dentro de un diseño de software.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

- Acoplamiento. Consiste en el grado de dependencia o interconexión de una clase o estructura de clase, con otras, esta muy ligada a la característica de Reutilización.
- Complejidad del mantenimiento. Consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costes y la planificación del proyecto.
- Cantidad de pruebas. Consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad (Unidad) del producto (Componente, modulo, clase, conjunto de clases, etc.) diseñado.
- Nivel de Cohesión. Consiste en el grado de especialización de las clases concebidas para modelar un dominio o concepto específico.

Las métricas concebidas como instrumento para evaluar la calidad del diseño de los componentes Puestos de Trabajo y Pagos Adicionales y su relación con los atributos de calidad definidos en este trabajo son las siguientes:

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

Tabla 13. Tamaño operacional de clase (TOC)

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

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

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

Tabla 14. Relaciones entre clases (RC)

Atributo que afecta	Modo en que lo afecta
Acoplamiento	Un aumento del RC implica un aumento del Acoplamiento de la clase.
Complejidad del mantenimiento	Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
Reutilización	Un aumento del RC implica una disminución en el grado de reutilización de la clase.
Cantidad de pruebas	Un aumento del RC implica un aumento de la cantidad de pruebas de unidad necesarias para probar una clase.

Resultados del instrumento de evaluación de la métrica Tamaño operacional de clase (TOC).

Ver instrumentos y tabla de resultados en [\(Anexo 3 Instrumento de medición de la métrica Tamaño operacional de clase \(TOC\)\)](#).

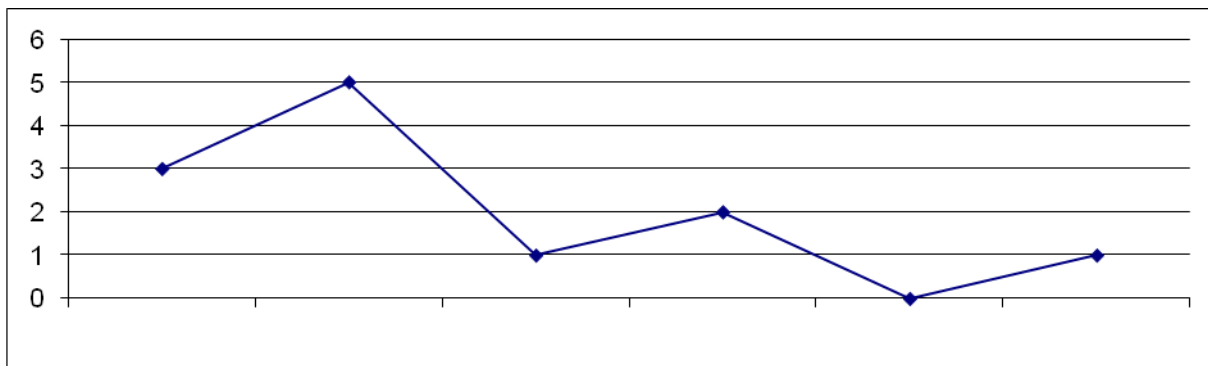


Figura 16. Representación de los resultados obtenidos en el instrumento agrupado en los intervalos definidos.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

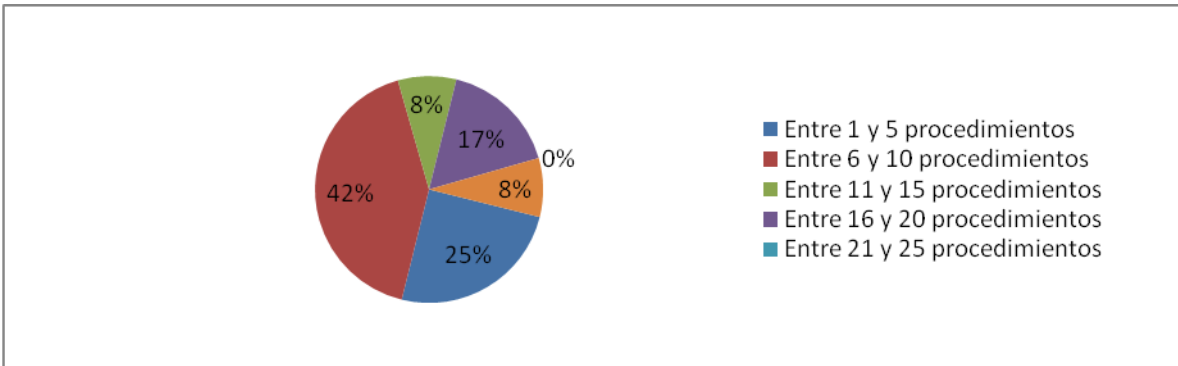


Figura 17. Representación en % de los resultados obtenidos en el instrumento agrupado en los intervalos definidos.

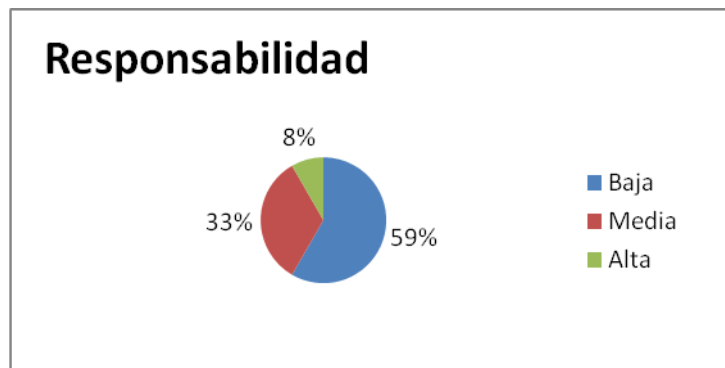


Figura 18. Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.

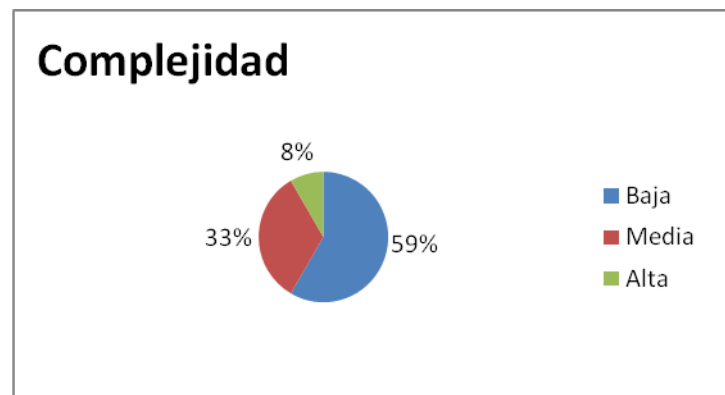


Figura 19. Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de Implementación.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

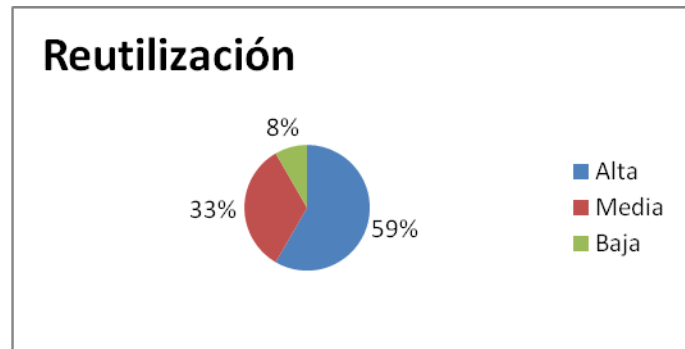


Figura 20. Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica TOC, se puede concluir lo siguiente:

- El 75 % de las clases incluidas en estos componentes posee menos cantidad de operaciones que la mitad del valor máximo registrado en las mediciones.
- El 75% de las clases poseen evaluaciones positivas en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización).

Por tanto el diseño de los Componentes Puesto de Trabajo y Pagos Adicionales tiene una calidad aceptable.

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

Ver instrumentos y tabla de resultados en [\(Anexo 4 Instrumento de medición de la métrica Relaciones entre clases \(RC\)\)](#).

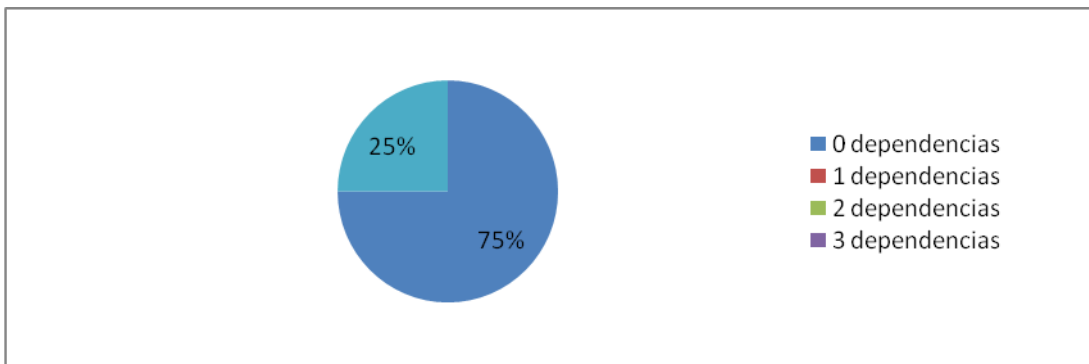


Figura 21. Representación en % de los resultados obtenidos en el instrumento agrupado en los intervalos definidos.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

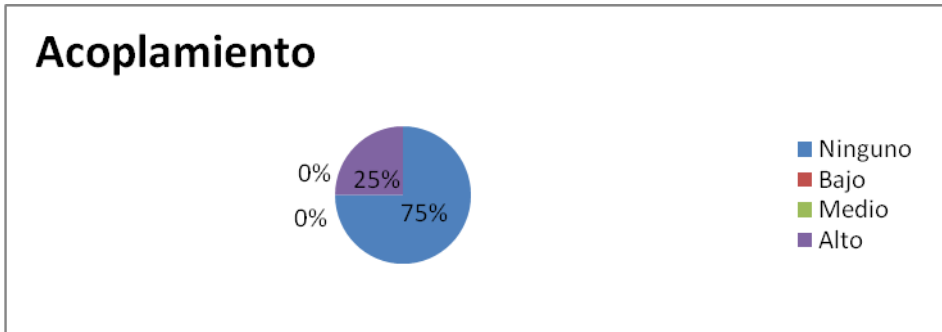


Figura 22. Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento.

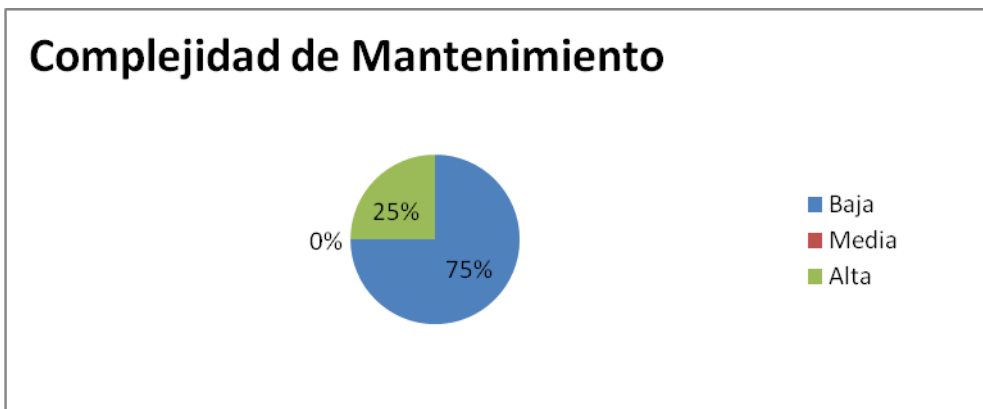


Figura 23. Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de Mantenimiento.

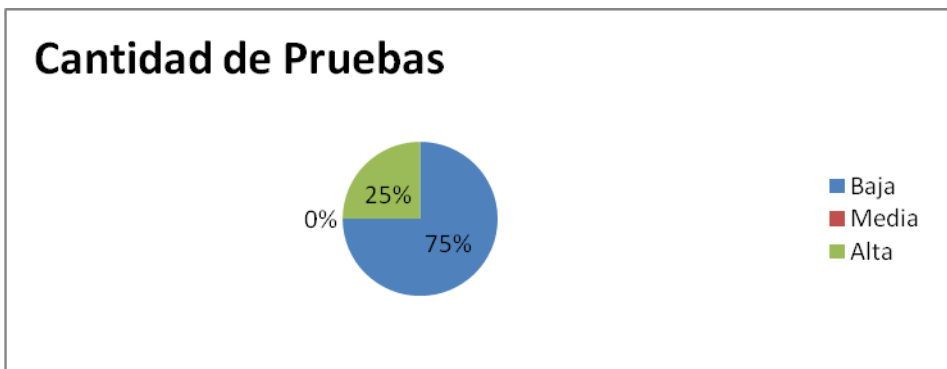


Figura 24. Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de Pruebas.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

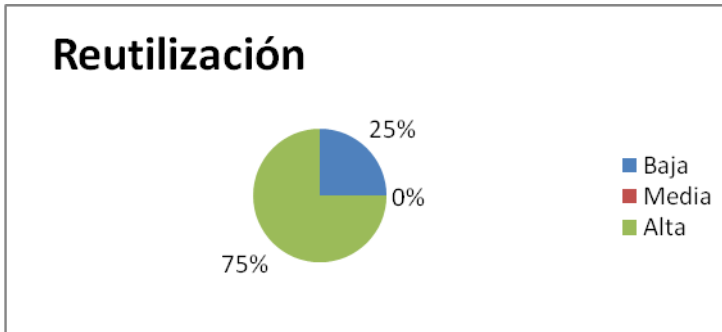


Figura 25. Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización.

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica RC, se puede concluir lo siguiente:

- El 75 % de las clases incluidas en estos componentes posee menos de 3 dependencias de otras clases.
- El 75 % de las clases no poseen acoplamiento con otras.
- El 75 % posee índices aceptables en cuanto a Acoplamiento.
- Los atributos de calidad Complejidad de Mantenimiento, Cantidad de Pruebas y Reutilización se comportan satisfactoriamente en un 75 % de las clases.

Se puede concluir que el diseño de los Componentes Puesto de Trabajo y Pagos Adicionales tiene una calidad aceptable.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

Conclusiones

El modelo de diseño es el artefacto de entrada principal para la implementación del sistema. Sin un diseño, se corre el riesgo de construir un sistema inestable, un sistema que fallará cuando se lleven a cabo cambios. A modo de conclusión se puede decir que con todo lo realizado en el capítulo se cumplieron con la mayor cantidad de objetivos propuestos en el presente trabajo dándole solidez a la solución al someter el diseño propuesto a una evaluación en la que se determinó el carácter idóneo del mismo. Se establecieron las bases por las cuales se registrarán los desarrolladores para poder pasar a la implementación de la solución.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

CAPÍTULO 3 IMPLEMENTACIÓN Y PRUEBA

Introducción

En este capítulo se muestra el modelo de implementación, clases de diseño en términos de componentes y se especifica el modelo de pruebas que comprueba las funcionalidades del sistema.

3.1 Modelo de implementación

El modelo de implementación describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables, etcétera. Describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros, además de los recursos necesarios para poder ejecutar el sistema desarrollado.

3.2 Diagrama de componentes

Los diagramas de componentes son utilizados para modelar la vista estática del sistema y representan las dependencias y organización entre los componentes que lo componen, estos diagramas muestran como quedan organizadas las clases del diseño en términos de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Uno de los usos principales es que puede servir para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

El diagrama de componentes de la solución muestra como los componentes Puestos de Trabajo y Pagos Adicionales se relacionan entre si y con otros componentes del mismo subsistema que en este caso es el componente Trabajador, además de relacionarse con componentes de otros subsistemas que no fue necesario especificar.

El componente Pagos Adicionales interactúa con el componente Puesto de Trabajo través de una interfaz en la que le brinda los pagos adicionales existentes para ser procesados. De esa misma forma el componente Puesto de Trabajo brinda al componente Trabajador los puestos de trabajo, el Componente Pagos Adicionales recibe del subsistema Costos y Procesos.

El diagrama de componentes realizado se puede ver en la figura 26.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

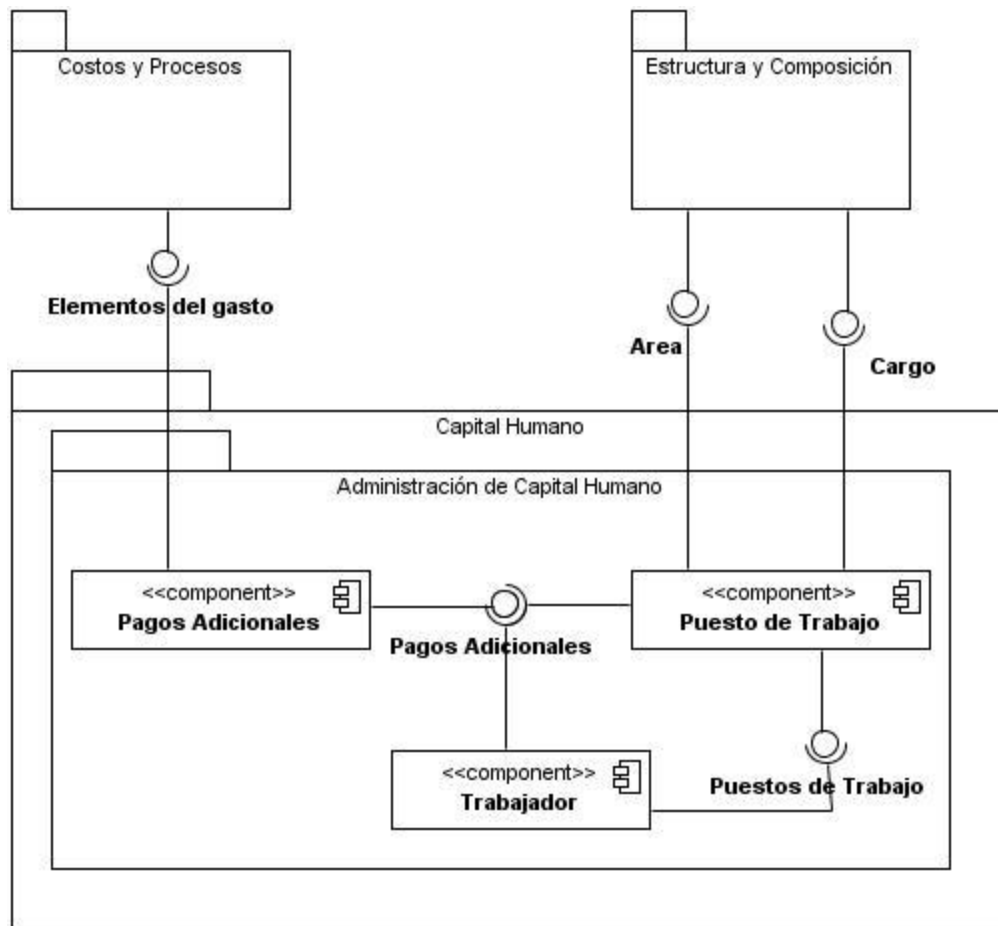


Figura 26. Diagrama de componentes

3.3 Modelo de prueba

Las pruebas del software son un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. Las pruebas son realizadas con el objetivo de detectar errores en el sistema, por lo que se llevan a cabo durante todo el ciclo de vida del producto. Los casos de prueba especifican una forma de probar el sistema, incluyendo la entrada o resultado con el que se ha de probar y las condiciones bajo las que ha de probarse.

Si se producen modificaciones en el programa, habrá que probar de nuevo todas las partes del programa afectadas por las modificaciones. Por lo dicho anteriormente, se desarrolló este epígrafe, con el objetivo de especificar las pruebas que serán hechas al software y describir todo lo encontrado en las mismas.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

3.3.1 Descripción de las pruebas

Las pruebas unitarias o pruebas de unidad son aplicadas por los implementadores sobre las unidades mínimas desarrolladas por ellos, estas unidades pueden ser clases, métodos, propiedades, componentes, etc.

Para realizar estas pruebas es necesario establecer una serie de reglas que sirven como objetivos de estas pruebas:

- La prueba es un proceso de ejecución de un programa con la intención de descubrir errores.
- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Las pruebas unitarias son orientadas casi siempre a las pruebas de caja blanca (se realiza sobre las funciones internas de un componente) aunque para realizar uno de estos test es necesario probar el flujo de datos desde la interfaz del componente. Si los datos no entran correctamente, todas las demás pruebas no tienen sentido. Una de las técnicas para ejecutar las pruebas de caja blanca es la del camino básico, el resultado de esta técnica es una medición cuantitativa de la complejidad lógica de un programa. El valor calculado como complejidad ciclomática, define el número de caminos independientes del conjunto básico de un programa y da un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez.

Otra prueba realizada a la solución obtenida fue la de caja negra, esta prueba se aplica sobre las interfaces del sistema, para ello las mismas deben de estar bien definidas en términos de entradas y salidas. Esta prueba se aplicó a la solución en los laboratorios de calidad del proyecto ERP-Cuba, los casos de prueba se realizaron para cada funcionalidad implementada.

3.3.2 Aplicación de pruebas de caja blanca

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

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

- Cada nodo del grafo corresponde a una o más sentencias de código fuente.
- Todo segmento de código de cualquier programa se puede traducir a un Grafo de Flujo.
- Se calcula la complejidad ciclomática del grafo.
- Para construir el grafo se debe tener en cuenta la notación para las instrucciones.

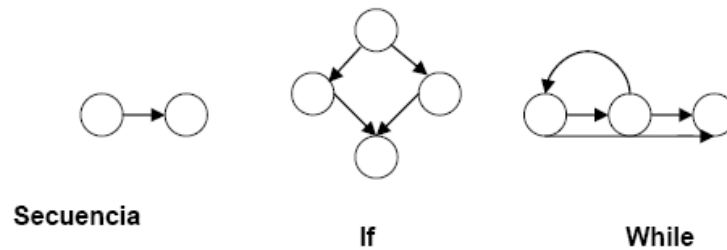


Figura 27. Notación de grafos de flujo para las instrucciones: Secuenciales, If, While

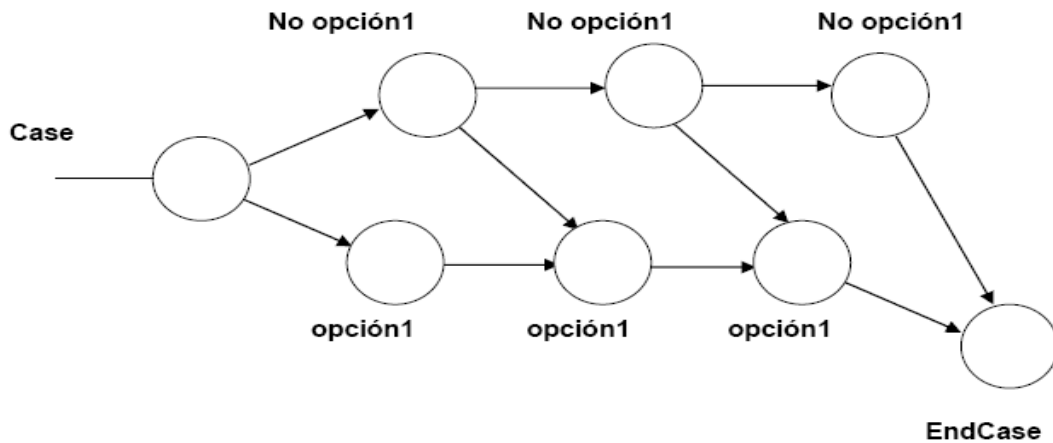


Figura 28. Notación de grafos de flujo para la instrucción Case

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

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

Componentes del grafo de flujo:

Nodo: son los círculos representados en el grafo de flujo, el cual representa una o más secuencias del procedimiento, donde un nodo corresponde a una secuencia de procesos o a una sentencia de decisión. Los nodos que no están asociados se utilizan al inicio y final del grafo.

Aristas: son constituidas por las flechas del grafo, son iguales a las representadas en un diagrama de flujo y constituyen el flujo de control del procedimiento. Las aristas terminan en un nodo, aun cuando el nodo no representa la sentencia de un procedimiento.

Regiones: son las áreas delimitadas por las aristas y nodos donde se incluye el área exterior del grafo, como una región más. Las regiones se enumeran siendo la cantidad de regiones equivalente a la cantidad de caminos independientes del conjunto básico de un procedimiento.

Para realizar la prueba de caja blanca específicamente la prueba del camino básico es necesario calcular antes la complejidad ciclomática del algoritmo o fragmento de código a analizar. A continuación se enumera las sentencias de código del procedimiento realizado sobre el método `insertarpagoAction()` el cual se encarga de adicionar un Pago Adicional en la tabla `nom_pagosadicionales`.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

```

function insertarpagoAction() {
    $entidad = $this->global->Estructura->idestructura; 1
    $NomPagosadicionale = new NomPagosadicionale ( ); 1
    $arrpadic = json_decode ( stripslashes ( $this->_request->getPost ( 'select' ) ) ); 1
    $ecuacio = $this->_request->getPost ( 'salario' ); 1
    if($ecuacio=='false' && count ( $arrpadic )==0){ 2
        $NomPagosadicionale->ecuacion='Null';) 3
    else (
        $ecuacion = $this->_request->getPost ( 'salario' ); 4
        for($i = 0; $i <= count ( $arrpadic ); $i ++){ 5
            $ecuacion = $ecuacion . ', ' . $arrpadic [$i]; 6
        } 7
        $NomPagosadicionale->ecuacion=$ecuacion; 7
    )
    $NomPagosadicionale->siesparaeltrabajador = $this->_request->getPost ( 'idasociado' ) - 10; 8
    $NomPagosadicionale->idcategoriapagosadicionales = $this->_request->getPost ( 'idcategoriapagosadicionales' ); 8
    $NomPagosadicionale->denom = $this->_request->getPost ( 'denom' ); 8
    $NomPagosadicionale->descripcion = $this->_request->getPost ( 'descripAdd' ); 8
    $NomPagosadicionale->codigo = $this->_request->getPost ( 'codigo' ); 8
    $NomPagosadicionale->valor = $this->_request->getPost ( 'valorAdd' ); 8
    $NomPagosadicionale->fechainicio = date ( 'd/m/Y' ); 8
    $NomPagosadicionale->idelementocomun = $this->_request->getPost ( 'id' ); 8
    $NomPagosadicionale->fechaefectiva = $this->_request->getPost ( 'startdt' ); 8
    $NomPagosadicionale->idestructuracomun =$entidad; 8
    $NomPagosadicionaleModel = new NomPagosadicionaleModel ( ); 8
    $NomPagosadicionaleModel->Insertar ( $NomPagosadicionale ); 8
    $this->showMessage ( 'El pago fue insertado correctamente.' ); 8
}

```

Figura 29. Método insertarpagoAction ()

Consecutivamente se construye el grafo de flujo asociado al código anterior.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

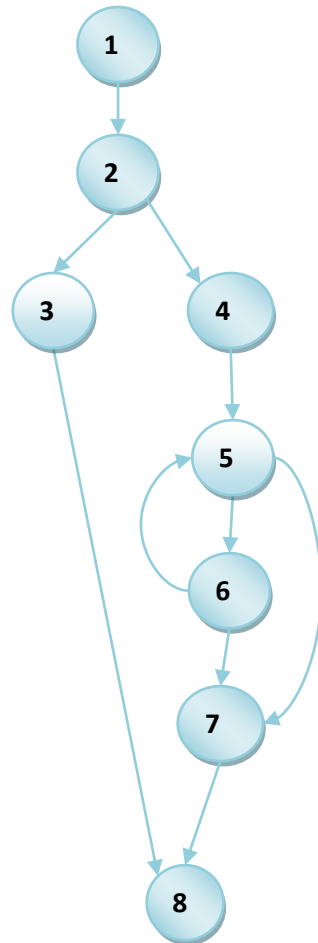


Figura 30. Grafo de flujo asociado al algoritmo insertarpagoAction ()

Cálculo de la complejidad ciclomática

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

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

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

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

$$V(G) = (10 - 8) + 2$$

$$V(G) = 4$$

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

Se puede usar también:

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

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

$$V(G) = 4$$

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

$$V(G) = R$$

$$V(G) = 4$$

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

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

Seguidamente es necesario representar los caminos básicos por los que puede recorrer el flujo. En estas representaciones se subrayan los elementos de cada camino que los hacen independientes a los demás.

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

Tabla 1 Caminos básicos del flujo

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

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

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

Caso de prueba para el camino básico 1:

Camino 1: [1 – 2 – 4 – 5 – 7 – 8]

Descripción:

Los datos de entrada cumplirán con los siguientes requisitos:

El valor de la entidad se tomará de los conceptos globales y será constante para todos los casos de pruebas, el `arrpadic` será un arreglo sin ningún pago adicional y `ecuacio` tendrá un valor de verdadero (`true`) donde ambos se obtienen del formulario de la vista.

Condición de ejecución:

- El `arrpadic` será un arreglo sin ningún pago adicional
- El `ecuacio` tendrá valor de `true`.

Entrada:

- `$arrpadic = array ()`.
- `$ecuacio = true`.

Resultados esperados:

- Se espera que se lance un mensaje con el siguiente texto: “El pago fue insertado correctamente”.

Caso de prueba para el camino básico 2

Camino 2: [1 – 2 – 4 – 5 – 6 – 7 – 8]

Descripción:

Los datos de entrada cumplirán con los siguientes requisitos:

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

- El valor de la entidad se tomará de los conceptos globales y será constante para todos los casos de pruebas, el `arrpadic` será un arreglo con un pago adicional y `ecuacio` tendrá un valor de verdadero (`true`) donde ambos se obtienen del formulario de la vista.

Condición de ejecución:

- El `arrpadic` será un arreglo con un pago adicional.
- El `ecuacio` tendrá valor de `true`.

Entrada:

- `$arrpadic` = array (1 pago).
- `$ecuacio` = `true`.

Resultados esperados:

- Se espera que se lance un mensaje con el siguiente texto: “El pago fue insertado correctamente”.

Caso de prueba para el camino básico 3

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

Descripción:

Los datos de entrada cumplirán con los siguientes requisitos:

- El valor de la entidad se tomará de los conceptos globales y será constante para todos los casos de pruebas, el `arrpadic` será un arreglo con dos pagos adicionales y `ecuacio` tendrá un valor de verdadero (`true`) donde ambos se obtienen del formulario de la vista.

Condición de ejecución:

- El `arrpadic` será un arreglo con dos pagos adicionales.
- El `ecuacio` tendrá valor de `true`.

Entrada:

- `$arrpadic` = array (2 pago).
- `$ecuacio` = `true`.

Resultados esperados:

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

- Se espera que se lance un mensaje con el siguiente texto: “El pago fue insertado correctamente”.

Caso de prueba para el camino básico 4

Camino 4: [1 – 2 – 3 – 8]

Descripción:

- Los datos de entrada cumplirán con los siguientes requisitos:

El valor de la entidad se tomará de los conceptos globales y será constante para todos los casos de pruebas, el `arrpadi` será un arreglo sin ningún pago adicional y `ecuacio` tendrá un valor de falso (`false`) donde ambos se obtienen del formulario de la vista.

Condición de ejecución:

- El `arrpadi` será un arreglo sin ningún pago adicional.
- El `ecuacio` tendrá valor de `false`.

Entrada:

- `$arrpadi = array ()`.
- `$ecuacio = false`.

Resultados esperados:

- Se espera que se inserte el pago pero el valor de la ecuación no estará incluido.

Luego de aplicar los distintos casos de pruebas, se pudo comprobar que el flujo de trabajo de la función está correcto ya que cumple con las condiciones necesarias planteadas anteriormente.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

Conclusiones

El modelo de implementación propicia un refinamiento de la vista de la arquitectura y sirve para mostrar parte de la vista estática del sistema a partir del diagrama de componentes, en el que se pueden observar las relaciones entre componentes que muestran la organización de dependencias lógicas entre los mismos. Se pudo apreciar como el modelo de implementación constituye una entrada fundamental para el modelo de prueba, del cual se obtuvieron los casos de prueba que ayudan a probar cada parte el producto realizado, el que ya se encuentra terminado.

CONCLUSIONES

Al culminar la investigación se le dio cumplimiento a los objetivos planteados, alcanzando el resultado propuesto, dos componentes capaces de cubrir los procesos de gestión de Puestos de Trabajo y Pagos Adicionales de forma correcta que a diferencia de otros sistemas permite una gestión integral con funcionalidades generales de los procesos y las particularidades de la economía cubana. Reafirmando así la utilidad y la validez del empleo de las tecnologías informáticas para apoyar las labores en cualquier esfera de la vida. Lo anterior se ve demostrado a través de lo siguiente:

- Se analizaron los fundamentos teóricos y las principales aplicaciones vinculadas al campo de acción, tanto a nivel nacional como internacional, demostrando la necesidad del nuevo sistema.
- Se realizó un estudio acerca de la metodología, tecnologías y herramientas definidas para obtener la nueva solución, reafirmando que eran las idóneas para ser utilizadas.
- Se logró obtener el modelo de negocio, análisis, diseño y se aplicaron las pruebas al sistema obteniéndose resultados positivos.

Es importante destacar que actualmente el sistema se encuentra en la fase de implantación donde se están probando todas sus funcionalidades en algunas entidades cubanas.

RECOMENDACIONES

RECOMENDACIONES

Teniendo como base los resultados de este trabajo y la experiencia adquirida durante el desarrollo del mismo, se recomienda:

- Agregar atributos al Componente Puestos de Trabajo de forma que permita una mejor descripción y comprensión de las características del puesto y lo que se hace en el mismo y facilitándose así el desarrollo de otros procesos que se nutren de la gestión de Puestos de Trabajo.
- Continuar con la investigación para garantizar nuevas mejoras en futuras versiones del sistema.
- Extender el despliegue del sistema a otras entidades.

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS BIBLIOGRÁFICAS

16. msdn . [En línea] Microsoft Corporation. [Citado el: 05 de 04 de 2009.] <http://msdn.microsoft.com/es-es/>.
17. Mitecnologico. [En línea] [Citado el: 02 de 04 de 2009.] <http://www.mitecnologico.com>.
18. Scribd . [En línea] [Citado el: 02 de 04 de 2009.] <http://www.scribd.com>.
19. Free Download Manager. [En línea] [Citado el: 02 de 04 de 2009.] <http://www.freedownloadmanager.org>.
20. Hernandez, Mildred Magdalena Vargas. INSTITUTO TECNOLOGICO DE MINATITLÁN. [En línea] [Citado el: 03 de 04 de 08.] www.uniopportunidades.com.mx.
21. Fundacion Universitaria Tecnolico Conmfenalco . [En línea] [Citado el: 03 de 04 de 2009.] <http://www.tecnologicocomfenalco.edu.co>.
22. Milestone. [En línea] [Citado el: 03 de 04 de 09.] <http://www.milestone.com.mx>.
23. Huacho. [En línea] [Citado el: 03 de 04 de 09.] <http://www.huacho.info>.
24. Viklund, Andreas. ACLibre. [En línea] [Citado el: 03 de 04 de 09.] <http://www.aclibre.org>.
25. Gracia, Joaquin. WebEstilo. [En línea] [Citado el: 03 de 04 de 09.] <http://www.webestilo.com/>.
26. Javier, Eguíluz Pérez. Libros Web. *Libros Web*. [En línea] 2007. [Citado el: 03 de 04 de 08.] <http://www.librosweb.es>.
27. HTML.net. Tutoriales sobre HTML y CSS . [En línea] [Citado el: 03 de 04 de 08.] <http://es.html.net>.
28. Taringa. [En línea] [Citado el: 04 de 04 de 08.] <http://www.taringa.net>.
29. Zend. [En línea] [Citado el: 04 de 04 de 08.] <http://www.zend.com/en/>.
30. Buenmaster. [En línea] [Citado el: 04 de 04 de 08.] <http://buenmaster.com>.
31. Leopoldo, Carlos. Carlos Leopoldo. [En línea] [Citado el: 04 de 04 de 08.] www.carlosleopoldo.com.
32. Xperimentos. [En línea] [Citado el: 03 de 04 de 09.] <http://www.xperimentos.com/>.
33. Planetacodigo. [En línea] [Citado el: 02 de 04 de 08.] <http://www.planetacodigo.com>.
34. El rincón del vago. [En línea] [Citado el: 03 de 04 de 08.] <http://html.rincondelvago.com>.

REFERENCIAS BIBLIOGRÁFICAS

35. I.E.S. San Vicente. [En línea] [Citado el: 02 de 04 de 08.] <http://www.iessanvicente.com/>.
36. ILSC. [En línea] [Citado el: 05 de 04 de 08.] <http://spanish.ilsc.org>.
37. fidesol. [En línea] [Citado el: 04 de 04 de 08.] <http://www.fidesol.org>.
38. ONUVA. [En línea] [Citado el: 04 de 04 de 08.] <http://www.ONUVA.com>.
39. Descargar Firefox. [En línea] [Citado el: 04 de 04 de 08.] <http://www.descargar-mozilla.com>.
40. MeRinde. [En línea] [Citado el: 05 de 04 de 09.] <http://merinde.rinde.gob.ve>.
41. SourceMaking. [En línea] [Citado el: 02 de 04 de 09.] <http://sourcemaking.com>.
50. ARREGUI, J. J. O. *Revisión Sistemática de Métricas de Diseño Orientado a Objetos*. Madrid : Facultad de Informática, 2005.
51. Pressman, R. S. *Ingeniería de software. Un enfoque practico*. s.l. : Mc Graw Hill, 1998.
52. Laura Bermejo, Zanz. *Eclipse como IDE*.
53. gestiopolis. [En línea] [Citado el: 08 de 04 de 209.] <http://www.gestiopolis.com>.
54. UNIVERSIDAD DE MURCIA. [En línea] [Citado el: 03 de 04 de 2009.] <http://www.um.es>.
55. LISI (Laboratorio de Investigacion en Sistemas de Informacion). [En línea] [Citado el: 02 de 04 de 2009.] <http://www.lisi.usb.ve>.
56. Enciclopedia Libre Universal en Español. [En línea] [Citado el: 04 de 04 de 09.] <http://enciclopedia.us.es>.
58. Procedimientos de configuración. [En línea] [Citado el: 02 de 04 de 08.] <http://megalinux.com.ar>.
59. Gracia, Joaquin. IngenieroSoftware. [En línea] [Citado el: 03 de 04 de 08.] <http://www.ingenierosoftware.com/>.
60. Tecnología y Synergix. [En línea] [Citado el: 04 de 04 de 09.] <http://synergix.wordpress.com>.
61. DSI(Departamento de Sistemas Informaticos). [En línea] [Citado el: 03 de 04 de 09.] <http://www.info-ab.uclm.es>.
62. MIT OPENCOURSEWARE . [En línea] 2003. [Citado el: 03 de 04 de 09.] <http://mit.ocw.universia.net>.
63. SPIN LATINOAMERICANO. [En línea] [Citado el: 03 de 04 de 09.] <http://www.latinspin.org>.

BIBLIOGRAFÍA

1. Constantino Nalda, Jose Ernesto Carrillo. *De la Administración del Personal a la Gestión de Capital Humano*. Madrid : Política y Sociedad, 1993. 13.
2. El prisma. [En línea] Derechos Reservados, 2000. [Citado el: 12 de abril de 2009.] <http://www.elprisma.com>.
3. Chiavenato, Idalberto. *Administración de Recursos Humanos*. Mexico : McGraw-Hill, 1992.
4. —. *Administración de Recursos Humanos*. Colombia : McGraw-Hill, 1999.
5. Noe, Mondy. *Administración de Recursos Humanos*. Mexico : Prentice Hall Hispanoamericana, 1997.
6. Oficina Nacional de Normalización. NORMA CUBANA 3000: 2007 . *SISTEMA DE GESTIÓN INTEGRADA DE CAPITAL HUMANO—VOCABULARIO*. Cuba : s.n., 2007.
7. “*Job Desing: Diseño del puesto de trabajo*”. Fredmund, Malik. 138, Madrid : Capital Humano, 2000.
8. *Gestión del talento humano*. Chiavenato. Bogotá : Prentice Hall, 2002.
9. Fernández, Manuel. “*Análisis y Descripción de Puestos de Trabajo*”. Madrid : Ediciones Díaz de Santos, 1995.
10. María García, Enrique Hierro, José Jiménez. *Selección de Personal- Sistema Integrado*. Madrid : ESIC, 2001.
11. Harper, Lynch. *Manuales de Recursos Humanos*. Madrid : La Gaceta de los Negocios, 1992.
12. GRH Guia de Servicios, Productos y Tecnologías para el área de Recursos Humanos. [En línea] INFORMACION CREATIVA. [Citado el: 02 de 04 de 2009.] <http://www.guiagrh.com/>.
13. SAYGER Sistema Administrativo y Gerencial . [En línea] [Citado el: 03 de 04 de 2009.] <http://www.sayger.com.ve>.
14. ASSET Sistema de Gestion Integral. [En línea] D’MARCO S.A. [Citado el: 04 de 04 de 2009.] <http://assets.co.cu>.
15. RODAS XXI Sistema Integrado Económico Administrativo. [En línea] CIMATEL. [Citado el: 07 de 04 de 2009.] <http://www.rodasxxi.cu>.
42. Universidad Católica Boliviana San Pablo. [En línea] [Citado el: 04 de 04 de 09.] <http://www.ucbcba.edu.bo>.

BIBLIOGRAFÍA

43. El Mundo Informático. [En línea] [Citado el: 05 de 04 de 09.] <http://jorgesaavedra.wordpress.com>.
44. Departamento de Ingeniería de Sistemas. [En línea] [Citado el: 03 de 04 de 09.] <http://sophia.javeriana.edu.co>.
45. UNRC / Facultad de Ciencias Exactas, Físico-Químicas y Naturales . [En línea] [Citado el: 04 de 04 de 09.] <http://dc.exa.unrc.edu.ar>.
46. UPNA(Universidad publica de Navarra). [En línea] [Citado el: 06 de 04 de 09.] <http://www.ayc.unavarra.es>.
47. Mugperu. [En línea] [Citado el: 04 de 04 de 09.] <http://www.mugperu.com>.
48. María, Jesús. INEI. [En línea] [Citado el: 04 de 04 de 09.] <http://www.inei.gob.pe>.
49. Web Docentes. [En línea] [Citado el: 04 de 04 de 09.] <http://virtual.usalesiana.edu.bo>.

GLOSARIO DE TÉRMINOS

Biblioteca: En ciencias de la computación, una biblioteca (o librería) es un conjunto de subprogramas utilizados para desarrollar software.

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

Cliente: Persona, organización o grupo de personas que encarga la construcción de un sistema, ya sea empezando desde cero, o mediante un refinamiento de versiones sucesivas.

Ciclo de vida de desarrollo software: Cubre cuatro fases en el siguiente orden: inicio, elaboración, construcción y transición.

Componente: Parte física y reemplazable del sistema que cumple y proporciona la realización de un conjunto de interfaces, ejemplo: ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares.

Debugger: Un depurador (en inglés, debugger), es un programa que permite depurar o limpiar los errores de otro programa informático.

Eclipse: Es una plataforma de desarrollo de código abierto, provee al programador con frameworks muy ricos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software y aplicaciones Web. Es una plataforma universal para integrar herramientas de desarrollo, con una arquitectura abierta y basada en plug-ins. Además, da soporte a todo tipo de proyectos que abarcan desde el ciclo de vida del desarrollo de aplicaciones, incluyendo soporte para modelado.

Framework: Un framework, en el desarrollo de software, es una estructura de soporte definida mediante la cual otro proyecto de software puede ser organizado.

GIF: (Graphics Interchange Format): En español Formato de Intercambio Gráfico Las imágenes GIF utilizan un máximo de 256 colores (8 bits) y permite configurar el fondo de las imágenes.

GLOSARIO DE TÉRMINOS

Herramientas de modelado: Las herramientas de modelado ayudan al ingeniero de software a visualizar el sistema a construir, permitan verificar que se comprenda correctamente el ambiente del usuario y que lo haya respaldado con información documental para que los diseñadores de sistemas y programadores puedan construir el sistema además de permitir discutir cambios y correcciones de los requerimientos del usuario, a bajo costo y con el riesgo mínimo.

HTTP: HyperText Transfer Protocol, protocolo cliente-servidor utilizado para el intercambio de páginas Web, Ver HTML.

Ingeniería de software: Aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software.

Licencia BSD: Licencia de software otorgada principalmente para los sistemas BSD. Pertenece al grupo de licencias de Software Libre; tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

Métricas: Mediciones para el software que se pueden aplicar al proceso de desarrollo con el intento de mejorarlo sobre una base continua. Se utilizan para la estimación, el control de la calidad, la evaluación de productividad y el control de proyectos. Ayuda a evaluar la calidad de los resultados de trabajos técnicos y en la toma de decisiones tácticas a medida que el proyecto evoluciona.

Navegador: Un navegador, o navegador web (en inglés, web browser) es un programa que permite visualizar la información que contiene una página web ya esté esta alojada en un servidor dentro de la World Wide Web (www) o en uno local.

Patrón: Solución común a un problema común de determinado contexto.

PDF: (acrónimo del inglés Portable Document Format, formato de documento portátil) es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems.

Plugins: Un complemento (o plug-in en inglés) es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

Producto: Artefactos que se crean durante la vida del proyecto, como los modelos, código fuente, ejecutables y documentación.

Programación del lado del cliente: Un lenguaje del lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en

GLOSARIO DE TÉRMINOS

cualquier sitio. El código, tanto del hipertexto como de los scripts, es accesible a cualquiera y ello puede afectar a la seguridad.

Programación del lado del servidor: La Programación del lado del servidor es una tecnología que consiste en el procesamiento de una petición de un usuario mediante la interpretación de un script en el servidor web para generar páginas HTML dinámicamente como respuesta.

Programación orientada a objetos: consiste en ordenar datos en conjuntos modulares de elementos de información del mundo real (denominado un dominio). Estos elementos de datos se llaman objetos. Estos datos se agrupan de acuerdo a las características principales del mundo real de estos elementos (tamaño, color, etc.).

Prototipo: Un prototipo es un ejemplar original o primer molde en que se fabrica una figura u otra cosa.

Refactoring: La refactorización (del inglés Refactoring) es una técnica de la ingeniería de software para reestructurar un código fuente, alterando su estructura interna sin cambiar su comportamiento externo.

Seguridad: Asegurar que los recursos del sistema de una organización sean utilizados de la manera que se decidió y que la información que se considera importante no sea fácil de acceder por cualquier persona que no se encuentre acreditada.

Servidor: En informática, un servidor es una computadora que, formando parte de una red, provee servicios a otros denominados clientes.

Subsistema: Sistema que es parte de otro sistema.

Transacciones: Conjunto de órdenes que se ejecutan formando una unidad de trabajo en forma indivisible o atómica.

Usuario: persona que utiliza o trabaja con algún objeto o que es destinataria de algún servicio público o privado, empresarial o profesional.

Versión: Es habitual que una aplicación software sufra modificaciones, mejoras o correcciones. El número de versión suele indicar el avance de los cambios. Suelen ser números correlativos y frecuentemente son dos cifras separadas por un punto.

Wizard: Programa que mediante pantallas interactivas facilitan y sirven de guía para el uso de otros programas o aplicaciones.

XHTML: es el acrónimo en inglés de Extensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto). Es una versión más estricta y limpia de HTML, que

GLOSARIO DE TÉRMINOS

nace precisamente con el objetivo de remplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos.

XML: eXtensible Markup Language, lenguaje de marcas extensible, metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Simplificación y adaptación del SGML que permite definir la gramática de lenguajes específicos. No es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

Zend Studio: Editor web orientado a la programación de páginas PHP, con ayudas en la gestión de proyectos y depuración de código. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. Consta de dos partes en las que se dividen las funcionalidades de parte del cliente y del servidor.

ANEXOS

Anexo 1 Modelo de datos

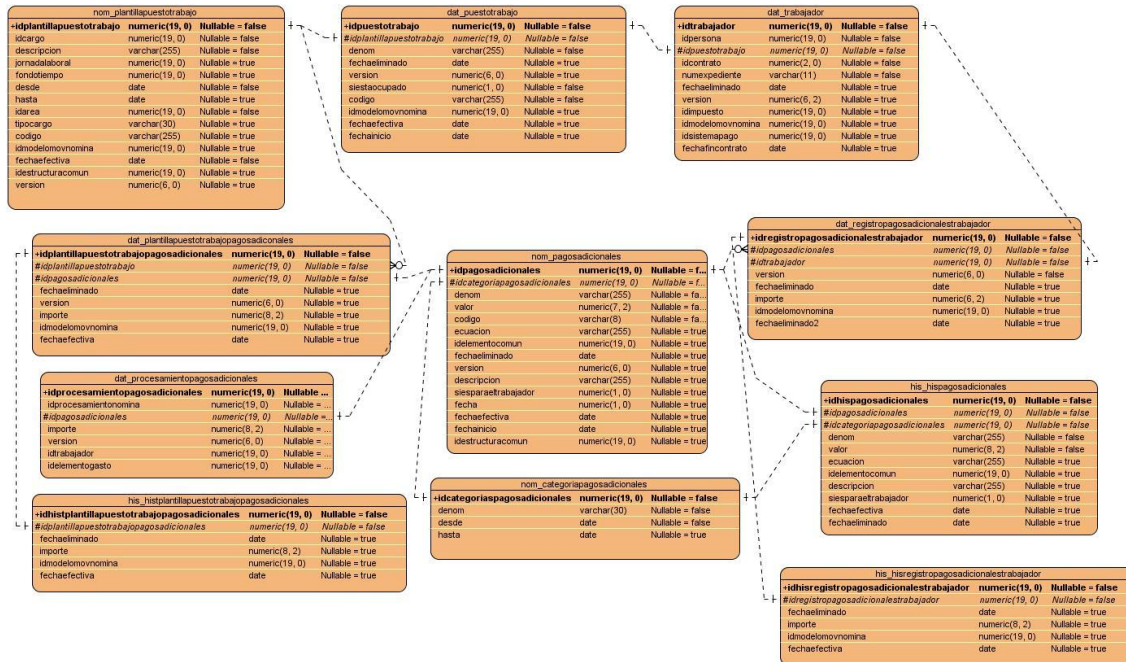


Figura 27. Modelo de datos

Anexo 2 Descripción de las tablas de la Base de Datos

En este epígrafe se hace una descripción de las principales tablas de la base de datos, en las que se almacena toda la información relacionada con los componentes.

Tabla 16. “nom_pagosadicionales”

Nombre: nom_pagosadicionales		
Descripción: Almacena los Pagos Adicionales.		
Atributo	Tipo	Descripción
idpagosadicionales	numeric(19,0)	Identificador de Pagos Adicionales
idcategoriapagosadicionales	numeric(19,0)	Identificador de categoría de Pagos Adicionales

denom	varchar(255)	Denominación de Pago Adicional
codigo	varchar(8)	Código de Pago Adicional
valor	numeric(7,2)	Equivalente al salario del Trabajador
ecuacion	varchar	Ecuación para calcular el salario del Trabajador
idelementocomun	numeric(19,0)	Identificador de elemento de gasto(lo provee Costos y Procesos)
fechaeliminado	date	Se utiliza para no eliminar físicamente de la Base de Datos
version	numeric(6,0)	Se utiliza para la Concurrencia.
descripcion	varchar(255)	Descripción de Pagos Adicionales
siesparaeltrabajador	numeric(1,0)	Indica si el pago adicional es para el Trabajador ejemplo (0 para el trabajador, 1 para el Puesto de Trabajo)
fechaefectiva	date	Fecha en que se hace efectivo el Pago Adicional
fechainicio	date	Fecha en que se registra el Pago Adicional
idestructuracomun	numeric(19,0)	Identificador de Entidad para Reporte de Pagos Adicionales

Tabla 17.” dat_puestotrabajo”

Nombre: dat_puestotrabajo		
Descripción: Almacena los Puestos de Trabajo.		
Atributo	Tipo	Descripción
idpuestotrabajo	numeric(19,0)	Identificador de Puesto de Trabajo
idplantillapuestotrabajo	numeric(19,0)	Identificador de Plantilla de Puesto de Trabajo
denom	varchar(255)	Denominación del Puesto de Trabajo
fechaeliminado	date	Se utiliza para no eliminar físicamente de la

Base de Datos		
version	numeric(6,0)	Se utiliza para la Concurrencia
siestaocupado	numeric(1,0)	Indica si el Puesto de Trabajo estáocupado o no (1 ocupado, 0 libre)
codigo	varchar(255)	Código del Puesto de Trabajo
idmodelomovnomina	numeric(19,0)	Identificador del Modelo de Movimiento de Nomina
fechaefectiva	date	Fecha en que se hace efectivo el Puesto de Trabajo
fechainicio	date	Fecha en que se crea el Puesto de Trabajo

Tabla 18.” nom_plantillapuestotrabajo”

Nombre: nom_plantillapuestotrabajo		
Descripción: Se almacenan grupos de puestos que sean similares		
Atributo	Tipo	Descripción
idplantillapuestotrabajo	numeric(19,0)	Identificador de Plantilla de Puesto de Trabajo
idcargo	numeric(19,0)	Identificador de Cargo (lo provee Estructura y Composición)
descripcion	varchar(255)	Descripción de Plantilla de Puesto de Trabajo
fondotiempo	numeric(3,0)	Fondo de tiempo de trabajo
jornadalaboral	numeric(3,0)	Jornada laboral que debe trabajar
desde	date	Fecha en que se crea la Plantilla
hasta	date	Fecha en la que se elimina la plantilla
idarea	numeric(19,0)	Identificador de Entidad (lo provee Estructura y Composición)
tipocargo	varchar(30)	si el cargo es civil o militar, lo da Estructura y Composición
codigo	varchar(255)	Código de la Plantilla

idmodelomovnomina	numeric(19,0)	Identificador del Modelo de Movimiento de Nomina
fechaefectiva	date	fecha efectiva en que entra en vigor
idestructuracomun	numeric(19,0)	identificador de Entidad (lo provee Estructura y Composición)
version	numeric(6,0)	Se utiliza para la Concurrencia

Tabla 19.” nom_categoriapagosadicionales”

Nombre: nom_categoriapagosadicionales		
Descripción: Almacena las categoría que puede tener un pago adicional		
Atributo	Tipo	Descripción
idcategoriapagosadicionales	numeric(19,0)	Identificador de Categoría de Pagos Adicionales
denom	varchar(30)	Denominación de Categoría
desde	date	Fecha inicial de donde se registra la Categoría
hasta	date	Fecha en que deja de ser vigente la Categoría

Tabla 20.” dat_plantillapuestotrabajopagosadicionales”

Nombre: dat_plantillapuestotrabajopagosadicionales		
Descripción: Almacena los Pagos Adicionales asociados al Puesto de Trabajo		
Atributo	Tipo	Descripción
idplantillapuestotrabajopagosadicionales	numeric(19,0)	Identificador de Pagos Adicionales asociados a Plantilla de Puesto de Trabajo

idplantillapuestotrabajo	numeric(19,0)	Identificador de Plantilla de Puesto de Trabajo
idpagosadicionales	numeric(19,0)	Identificador de Pagos Adicionales
fechaeliminado	date	Se utiliza para no eliminar físicamente de la Base de Datos
version	numeric(6,0)	Utilizado para la Concurrencia
importe	numeric(8,2)	Importe de Pagos Adicionales asociados a Plantilla de Puesto de Trabajo
idmodelomovnomina	numeric(19,0)	Identificador del Modelo de Movimiento de Nomina
fechaefectiva	date	Fecha a partir de la cual se hacen efectivos los cambios efectuados sobre la Plantilla de Puesto de Trabajo

Tabla 21.” his_hispagosadicionales”

Nombre: his_hispagosadicionales		
Descripción: Historial de los Pagos Adicionales		
Atributo	Tipo	Descripción
idhispagosadicionales	numeric(19,0)	Histórico de Pagos Adicionales
idpagosadicionales	numeric(19,0)	Identificador de Pagos Adicionales
idcategoriapagosadicionales	numeric(19,0)	Identificador de Categoría de Pagos Adicionales
denom	varchar(255)	Denominación de Pagos Adicionales
valor	numeric(8,2)	Pago Adicional asignado a un Trabajador
ecuacion	varchar	Ecuación para calcular el salario de un Trabajador
idelementocomun	numeric(19,0)	identificador de Elemento de Gasto (lo

		provee Costos y Procesos)
descripcion	varchar(255)	Descripción de Pagos Adicionales
siesparaeltrabajador	numeric(1,0)	Indica si el Pago Adicional esta asignado o no al Trabajador
fechaefectiva	date	Fecha en que se hace efectivo el Pago Adicional
fechaeliminado	date	Fecha en la que fue eliminado lógicamente de la Base de Datos

Tabla 22.” his_histplantillapuestotrabajopagosadicionales”

Nombre: his_histplantillapuestotrabajopagosadicionales		
Descripción: Historial de Pagos Adicionales asociados al Puesto de Trabajo		
Atributo	Tipo	Descripción
idhistplantillapuestotrabajopagosadicionales	numeric(19,0)	Identificador de Histórico de Plantilla de Puesto de Trabajo con asociación de Pagos Adicionales
idplantillapuestotrabajopagosadicionales	numeric(19,0)	Identificador de Plantilla de Puesto de Trabajo con Pagos Adicionales asociados
fechaeliminado	date	Se utiliza para no eliminar físicamente de la Base de Datos
importe	numeric(8,2)	Importe de Pagos Adicionales
idmodelomovnomina	numeric(19,0)	Identificador del Modelo de Movimiento de Nomina
fechaefectiva	date	Fecha en que se hace efectiva la modificación en la Plantilla de Puesto de

		Trabajo
--	--	---------

Anexo 3 Instrumento de medición de la métrica Tamaño operacional de clase (TOC)

	Categoría	Criterio
Responsabilidad	Baja	\leq Prom.
	Media	Entre Prom. y 2^* Pom.
	Alta	$> 2^*$ Prom.
Complejidad implementación	Baja	\leq Prom.
	Media	Entre Prom. y 2^* Pom.
	Alta	$> 2^*$ Prom.
Reutilización	Baja	$> 2^*$ Prom.
	Media	Entre Prom. y 2^* Pom.
	Alta	\leq Prom.

Tabla 48. Rango de valores de para la evaluación técnica de los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización) relacionados con la métrica TOC.

No	Componentes	Clase	Cantidad de Procedimientos	Responsabilidad	Complejidad	Reutilización
1	Pagos Adicionales	NomCategoriapagosadicio	4	Baja	Baja	Alta
2	Pagos Adicionales	NomPagosadicionaleMod	4	Baja	Baja	Alta
3	Pagos Adicionales	NomCategoriapagosadicio	6	Baja	Baja	Alta
4	Pagos Adicionales	NomPagosadicionale	13	Media	Media	Media
5	Puesto de Trabajo	DatPlantillapuestotrabajop	7	Baja	Baja	Alta
6	Puesto de Trabajo	DatPuestotrabajoModel	4	Baja	Baja	Alta
7	Puesto de Trabajo	NomPlantillapuestotrabajo	8	Baja	Baja	Alta
8	Puesto de Trabajo	DatPuestotrabajo	29	Alta	Alta	Baja
9	Puesto de Trabajo	DatPlantillapuestotrabajop	6	Baja	Baja	Alta
10	Pagos Adicionales	Gestionarpagosadicionale	10	Media	Media	Media
11	Puesto de Trabajo	Gestionarplantillapuestotr	16	Media	Media	Media
12	Puesto de Trabajo	GestionarpuestotrabajoCo	16	Media	Media	Media

Tabla 49. Resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización).

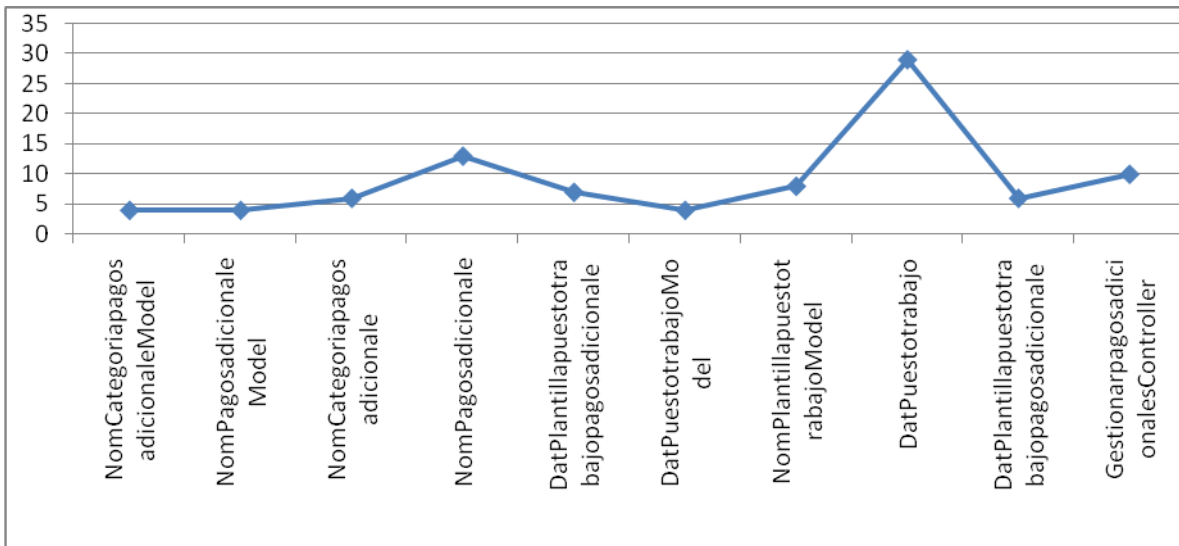


Figura 28 Gráfica de los resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización).

Anexo 4 Instrumento de medición de la métrica Relaciones entre clases (RC)

	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	>2
	Categoría	Criterio
Complejidad Mant.	Baja	\leq Prom.
	Media	Entre Prom. y $2*Prom.$
	Alta	$> 2*Prom.$
	Categoría	Criterio
Reutilización	Baja	$>2* Prom.$
	Media	Entre Prom. y $2*Prom.$
	Alta	\leq Prom.
	Categoría	Criterio
Cantidad de Pruebas	Baja	\leq Prom.
	Media	Entre Prom. y $2*Prom.$
	Alta	$> 2*Prom.$

Tabla 50. Rango de valores de para la evaluación técnica de los atributos de calidad (Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas) relacionados con la métrica RC.

No	Subsistema	Clase	Cantidad de Relaciones de Uso	Acoplamiento	Complejidad	Reutilización	Cantidad
1	Pagos Adicionales	NomCategoriapagosadicionaleModel	0	Ninguno	Baja	Alta	Baja
2	Pagos Adicionales	NomPagosadicionaleModel	0	Ninguno	Baja	Alta	Baja
3	Pagos Adicionales	NomCategoriapagosadicionale	0	Ninguno	Baja	Alta	Baja
4	Pagos Adicionales	NomPagosadicionale	0	Ninguno	Baja	Alta	Baja
5	Puesto de Trabajo	DatPlantillapuestotrabajopagosadicionale	0	Ninguno	Baja	Alta	Baja
6	Puesto de Trabajo	DatPuestotrabajoModel	0	Ninguno	Baja	Alta	Baja
7	Puesto de Trabajo	NomPlantillapuestotrabajoModel	0	Ninguno	Baja	Alta	Baja
8	Puesto de Trabajo	DatPuestotrabajo	0	Ninguno	Baja	Alta	Baja
9	Puesto de Trabajo	DatPlantillapuestotrabajopagosadicionale	0	Ninguno	Baja	Alta	Baja
10	Pagos Adicionales	GestionarpagosadicionaleController	4	Alto	Alta	Baja	Alta
11	Puesto de Trabajo	GestionarplantillapuestotrabajoController	9	Alto	Alta	Baja	Alta
12	Puesto de Trabajo	GestionarpuestotrabajoController	6	Alto	Alta	Baja	Alta

Tabla 51. Resultados de la evaluación de la métrica RC y su influencia en los atributos de calidad (Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas).

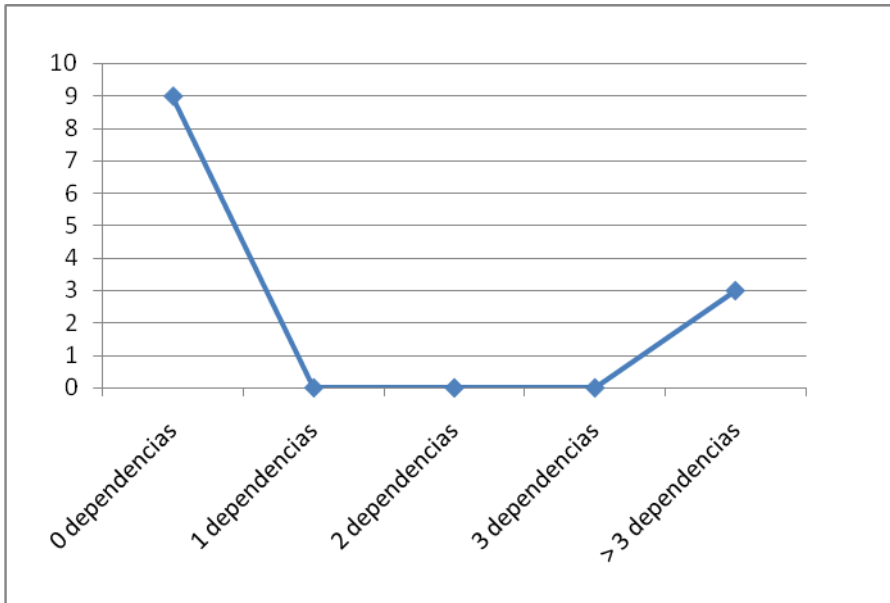


Figura 29 Gráfica de los resultados de la evaluación de la métrica RC agrupados por la tendencia de los valores.