



**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERÍA EN CIENCIAS INFORMÁTICAS**

**Personalización del módulo para la gestión del
sistema de pago a recursos humanos de la
Universidad de Ciencias Informáticas en Odoov10.0**

Autor: Raykof Reynier Peñalver García

Tutor: Ing. Noidis Barroso Hidalgo

Tutor: Ing. Yaicel Torres Garcés

La Habana, 29 de junio de 2018

“Año 60 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Raykof Reynier Peñalver García

Firma del Autor

Ing. Noidis Barroso Hidalgo

Firma del Tutor

Ing. Yaicel Torres Garcés

Firma del Tutor

DATOS DE CONTACTO

Síntesis del Tutor(es):

Ing. Yaicel Torres Garcés

Ingeniero en Ciencias Informáticas. Graduado en la Universidad de las Ciencias Informáticas en el 2015. Actualmente trabaja como desarrollador en el departamento de Aplicaciones de gestión empresarial del Centro de Informatización de Entidades, Facultad 3.

Correo: ygarces@uci.cu

Ing. Noidis Barroso Hidalgo

Ingeniero en Ciencias Informáticas. Graduada en la Universidad de las Ciencias Informáticas en el 2010. Actualmente trabaja como analista en el departamento de Aplicaciones de gestión empresarial del Centro de Informatización de Entidades, Facultad 3.

Correo: nbarroso@uci.cu

AGRADECIMIENTOS

*A los **maestros**, aquellos que marcaron cada etapa de nuestro camino universitario,*

*A mis **tutores** Noidis y Yaicel, por todo el apoyo y dedicación mostrado durante la realización de este trabajo, ¡Sin ustedes no hubiera sido posible!*

A mis amigos

Michel, Alexander, Chris, Ochill, Victor Manuel, Oscar, Yilena, Barbaro, Nailee, El Dasho, El Moreno y todos aquellos amigos que no recordé al momento de escribir esto. ¡Ustedes saben quiénes son!!!

A mi padre Reynaldo

Por los ejemplos de perseverancia y constancia que lo caracterizaban y que me infundó siempre, por el valor mostrado para salir adelante y por su amor.

A mi Tío

Por ser el ejemplo a seguir en mi formación, por todas las enseñanzas y experiencias adquiridas a tu lado.

A mi madre Regla

Por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor.

En general gracias a todas aquellas personas que han sido mi soporte y compañía durante todo el período de estudio.

DEDICATORIA

A mi padre, mi tío y a ti **REGLA**, recuerdo que cuando era pequeño, observaba cómo trabajabas, empeñosa y ambiciosa. Quizás a esa edad no podía comprender totalmente por qué dedicabas tantas horas a tus labores. Luego entendí que existía una razón que te movía a luchar constantemente, y esa era yo. “Es por ti, Madre que he llegado a donde estoy, por ti es que soy el hombre que soy, porque con tu esfuerzo pudiste darme lo que tal vez tú nunca tuviste y por eso yo te daré lo que tal vez nunca soñaste con tener. Te admiro, te adoro, te estoy eternamente agradecido, siempre te tendré como un ejemplo de lo que una persona que ama de verdad es capaz de hacer”

RESUMEN

Los sistemas de pago son las modalidades de las formas de pago que se adoptan en correspondencia con las características técnico-organizativas del proceso laboral, de la organización del trabajo y de las posibilidades de control de sus resultados (Ministerio de Trabajo y Seguridad Social, 2009)

En la Universidad de las Ciencias Informáticas (UCI), a partir de la decisión adoptada por el Gobierno de incrementar el nivel salarial para los profesores, investigadores y demás trabajadores que laboran en la educación superior fue autorizado un sistema pago adicional especial con características específicas (Ministerio de Trabajo y Seguridad Social, 2009). La presente investigación se centra en los procesos que rigen el sistema de pago de la Universidad de las Ciencias Informáticas.

Para guiar el desarrollo de la solución se definieron procesos de negocio y los requisitos del sistema, los cuales permitieron obtener la información requerida para realizar el diseño, sentando las bases para la implementación de la solución propuesta.

Palabras claves: Sistemas de pago, UCI, procesos.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
INTRODUCCIÓN	5
1.1 CONCEPTOS ASOCIADOS	5
1.2 ESTADO DEL ARTE	5
1.3 METODOLOGÍA DE DESARROLLO	9
1.3.1 DISCIPLINA MODELADO DE NEGOCIO	9
1.3.2 DISCIPLINA REQUISITOS	10
1.3.3 DISCIPLINA ANÁLISIS Y DISEÑO	12
1.3.4 DISCIPLINA IMPLEMENTACIÓN	15
1.3.5 DISCIPLINA PRUEBAS INTERNAS	19
CONCLUSIONES PARCIALES.....	23
CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN	24
INTRODUCCIÓN	24
2.1 DESCRIPCIÓN DEL NEGOCIO.....	¡ERROR! MARCADOR NO DEFINIDO.
2.1.1 REGLAS DEL NEGOCIO	25
2.2 MODELADO DE LA PROPUESTA DE SOLUCIÓN	26
2.2.1 DESCRIPCIÓN DE LOS PROCESOS DE NEGOCIO	26
2.2.2 PAGO SALARIO BÁSICO	26
2.2.3 PAGO SALARIO ADICIONAL	27
2.2.4 EVALUACIÓN DEL DESEMPEÑO.....	28
2.3 MODELO CONCEPTUAL	29
2.4 DICCIONARIO DE DATOS	30
2.5 REQUISITOS DEL SISTEMA.....	32
2.5.1 TÉCNICAS UTILIZADAS PARA LA IDENTIFICACIÓN DE REQUISITOS....	32
2.5.2 ESPECIFICACIÓN DE REQUERIMIENTOS FUNCIONALES	33

ÍNDICE DE CONTENIDOS

2.5.3 DESCRIPCIÓN DE REQUISITOS FUNCIONALES.....	34
2.5.5 REQUISITOS NO FUNCIONALES.....	37
2.5.6 TÉCNICAS DE VALIDACIÓN DE REQUISITOS	39
2.6 DISEÑO DE LA PROPUESTA DE SOLUCIÓN	39
2.6.1 PATRONES DE DISEÑO	40
2.6.2 PATRÓN ARQUITECTÓNICO	41
2.6.3 DIAGRAMA DE CLASES CON PROTOTIPOS WEB;¡ERROR! MARCADOR NO DEFINIDO.	
2.6.4 VALIDACIÓN DEL DISEÑO	42
2.7 MODELO DE DATOS	46
CONCLUSIONES PARCIALES.....	47
CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN	49
INTRODUCCIÓN	49
3.1 IMPLEMENTACIÓN.....	49
3.1.1 ESTÁNDARES DE CODIFICACIÓN	49
3.2 PRUEBAS INTERNAS	50
3.2.1 PRUEBAS DE CAJA BLANCA	50
3.2.2 PRUEBA DE CAJA NEGRA.....	53
3.2.3 PRUEBAS INTERNAS	56
3.2.4 PRUEBAS DE ACEPTACIÓN	56
CONCLUSIONES PARCIALES.....	59
CONCLUSIONES GENERALES	60
RECOMENDACIONES	61
BIBLIOGRAFÍA	62
ANEXOS;¡ERROR! MARCADOR NO DEFINIDO.	

ÍNDICE DE FIGURAS

Figura 1 Patrón Vista Controlador (Fabien Potencier, 2016)	14
Figura 2 Descripción de proceso Pago de salario básico	27
Figura 3 Descripción de subproceso Elaboración de prenomina	27
Figura 4 Descripción de proceso Pago Adicional.....	28
Figura 5 Descripción de proceso Evaluación de desempeño	29
Figura 6 Modelo Conceptual.....	30
Figura 7 Prototipo de Interfaz de Requisito Adicionar Cargo	37
Figura 8 Diagrama de clases de diseño Gestionar Cargo	40
Figura 9 Patrón GOF Adaptador	41
Figura 10 Patrón Arquitectónico Modelo Vista Controlador.....	42
Figura 11 Resultado de la evaluación de la métrica para los atributos responsabilidad y complejidad	44
Figura 12 Resultado de la evaluación de la métrica para el atributo reutilización	44
Figura 13 Resultado de la evaluación de la métrica para los atributos acoplamiento y complejidad de mantenimiento.....	46
Figura 14 Resultado de la evaluación de la métrica para los atributos cantidad de pruebas y reutilización	46
Figura 15 Modelo de datos	47
Figura 16 Diagrama de componentes	49
Figura 17 Método def transform_incidencias(self)	51
Figura 18 Grafo resultante de aplicar técnica Camino Básico.	51

ÍNDICE DE TABLAS

Tabla 1 Diccionario de datos	30
Tabla 2 Descripción de Requisito Adicionar Cargo Civil	34
Tabla 3 Requisitos no funcionales	37
Tabla 4 Criterios de evaluación.....	43
Tabla 5 Atributos de calidad a evaluar por la métrica RC.....	44
Tabla 6 Criterios y categorías a medir durante la evaluación	45
Tabla 7 Diseño caso prueba de caja negra.....	54
Tabla 8 No Conformidades detectadas	56
Tabla 9 Valores asociados a cada respuesta	57
Tabla 10 Resultados de la aplicación.....	58
Tabla 11 Evaluación de las preguntas aplicadas	58

Introducción

El desarrollo de mecanismos de optimización de procesos empresariales, se ha convertido en una necesidad para manejar la información de forma confiable y segura, posibilitando compartir información entre todos los componentes de la organización. Una de las áreas que se integra a la planificación de las empresas es Recursos Humanos la cual incluye un número de procesos que son de interés para las entidades y el personal que en estas laboran; el sistema de pagos es uno de los procesos antes mencionado.

En la Universidad de las Ciencias Informáticas (UCI), a partir de la decisión adoptada por el Gobierno de incrementar el nivel salarial para los profesores, investigadores y demás trabajadores que laboran en la educación superior fue autorizado un pago adicional especial con características específicas. (Ministerio de Trabajo y Seguridad Social, 2009)

Esta actividad de pago es gestionada en la universidad en el sistema ASSETS NS¹. Sin embargo, en el software utilizado, según los especialistas que laboran en él, existen deficiencias que dificultan la ejecución de dichos procesos:

Como se puede ver:

- En la realización del movimiento de nómina del trabajador, el sistema muestra errores en las fechas, mostrando la fecha asignada al último movimiento del trabajador.
- El sistema tiene predefinido el 5% de las afectaciones por la seguridad social para todos los trabajadores, por lo que en caso de no corregirlo manualmente se incurrirían en errores de cálculo de salarios.
- En el sistema, se realizan consultas (movimientos de altas) que no reflejan el número de expediente del trabajador, resultando en algunos casos engorroso el trabajo.
- En el Anexo 14 B de la plantilla de cargo cubierto no están detallados el pago correspondiente a la categoría Master y Doctor.
- En la confección de un movimiento a un trabajador el sistema, elimina la Maestría y el Doctorado, por lo que es necesario estar al tanto de ello cada vez que se realice un movimiento para no incurrir posteriormente en errores de cálculo en la formación del salario.
- El módulo de recursos humanos no se gestiona la pre-nómina.
- Existen deficiencias en la salva de los promedios mensuales referidos a la seguridad social, por lo que resulta agotador realizar el seguimiento de manera mensual.
- En la realización de un movimiento asociado a un trabajador, en caso de que un mes conste de

¹ Sistema de Gestión Integral

INTRODUCCIÓN

un período laboral diferente al definido por defecto de 24hr, ejemplo 25, 22 días laborales, se debe definir de forma manual en el sistema.

- El promedio de certificado de una embarazada, es necesario añadir lo datos al sistema de manera manual, incurriendo en pérdida de tiempo.

El Centro de Informatización de Entidades (CEIGE) en aras de desarrollar un ERP, por las características presentes en esta herramienta adoptó al Odoo como base para la implementación.

El sistema de planificación de recursos empresariales (ERP²) Odoo constituye una herramienta capaz de integrar y manejar negocios relacionados con la actividad productiva y distributiva de compañías de bienes y servicios. Odoo como ERP es un software integrado que automatiza y optimiza prácticamente todos los departamentos de la empresa, con lo cual se convierte en una herramienta esencial para todo negocio con ansias de lograr mayor organización, control, optimización de recursos, calidad y eficacia en los procesos que intervienen en una determinada entidad (ODOO, 2016).

Este sistema permite la integración de ciertas operaciones de la empresa como logística, inventario, recursos humanos, contabilidad, entre otros. Sin embargo, el módulo provisto por defecto para gestionar los recursos humanos está condicionado a intereses ajenos de nuestra nación, en ocasiones no cumple con las expectativas que se requieren y no se adapta a las características específicas del sistema de pago de la UCI, esto dificulta en gran medida los procesos de gestión afectando el éxito de las tareas que la organización realiza como parte de su accionar diario.

Por tales razones se hace necesario desarrollar un módulo que gestione el sistema de pago del subsistema de recursos humanos de Odoo, que permita mejorar el trabajo del personal encargado de gestionar la nómina y pre-nómina de los empleados de una forma sencilla, realizar cálculos de forma rápida y conforme a la ley del salario de cada trabajador por concepto de sueldo, bonificaciones, horas extras y deducciones relacionadas con el pago de impuestos y retenciones. Logrando con ello que esté acorde a la realidad existente en la UCI y en consecuencia con el perfeccionamiento empresarial y los lineamientos.

A partir de la problemática antes planteada se define como **Problema a resolver**: ¿Cómo contribuir a mejorar el proceso de gestión del sistema de pago de la UCI?

Objeto de estudio: Informatización de las formas y sistemas de pagos.

Campo de acción: Gestión de sistemas de pago de la UCI.

² Planificación de recursos empresariales.

Objetivo general: Personalizar el módulo para gestionar el sistema de pago de los recursos humanos de la UCI.

Objetivos específicos

- Elaborar el marco teórico conceptual de la investigación, relacionado con los sistemas de planificación de recursos empresariales para definir una propuesta de solución.
- Realizar el análisis y diseño del proceso de gestión del sistema de pago de recursos humanos de la UCI.
- Implementar el módulo de gestión de sistema de pago a los recursos humanos de la UCI.
- Validar la solución propuesta a partir de pruebas funcionales.
- Realizar la validación de la investigación mediante la aplicación de los métodos de validación.

Posibles resultados: El módulo funcional, los distintos artefactos generados y el documento de Tesis.

Idea a defender: Con la personalización del módulo de sistemas de pago a recursos humanos de la Universidad de Ciencias Informáticas en Odoo se contribuye a mejorar los procesos de gestión de sistema de pago.

Métodos de investigación

- **Histórico-Lógico:** Analizan la trayectoria completa del fenómeno, su condicionamiento a los diferentes períodos de la historia, revela las etapas principales de su desenvolvimiento, las conexiones históricas fundamentales, además de poner de manifiesto la lógica interna de su desarrollo. Expresa de forma teórica la esencia del objeto, permitiendo unir el estudio de la estructura del objeto de investigación con su concepción histórica (Carlos Tinchet Varela, 2014). Su aplicación se evidenció en el estudio de Odoo a través del informe de CEIGE
- **Hipotético – deductivo:** A partir de la hipótesis y siguiendo reglas lógicas de deducción se llega a nuevos conocimientos y predicciones, las que posteriormente son sometidas a verificaciones empíricas (Carlos Tinchet Varela, 2014). Su aplicación se evidenció cuando se realizó un profundo estudio de los sistemas encargados de la gestión del sistema de pagos a recursos humanos, en el cual se analizó el estado actual de estos sistemas, recolectando así mucha información, lo que trajo consigo formular la idea a defender anteriormente planteada
- **Modelación:** La modelación es justamente, el método mediante el cual se crean abstracciones con vistas a explicar la realidad. Su aplicación en este trabajo se evidencia

en la elaboración de los diferentes diagramas, mapas conceptuales y prototipos durante el desarrollo del sistema (Carlos Tinchet Varela, 2014)

- **Entrevista:** Comunicación verbal entre dos o más personas con el objetivo de obtener información acerca de determinadas cuestiones (Carlos Tinchet Varela, 2014). Su aplicación permitió realizar entrevistas individuales con especialistas de la Dirección de Contabilidad y RRHH³, obteniendo una valiosa descripción de la información referente al negocio
- **Observación:** La observación científica es la percepción planificada dirigida a un fin y relativamente prolongada de un hecho o fenómeno. Es el instrumento universal del científico, se realiza de forma consciente y orientada a un objetivo determinado (Carlos Tinchet Varela, 2014). Se utilizó la observación al apreciar sistemáticamente el módulo de Nóminas del sistema Assets para lograr un mejor entendimiento del funcionamiento del mismo

Estructuración de los capítulos

La presente investigación consta de tres capítulos y contiene varios anexos con los artefactos generados. A continuación, se describen los principales objetivos de cada uno de los capítulos.

Capítulo 1: Fundamentación teórica

Enuncia la problemática actual en los procesos de gestión del sistema de pago en la UCI y la no correspondencia del ERP Odoó con las políticas de los lineamientos, incluyendo un estado del arte de los sistemas existentes que gestionan los procesos de Nóminas y Recursos Humanos, de las tendencias, técnicas y tecnologías utilizadas en el mundo de hoy para desarrollarlos, así como la fundamentación de la metodología y herramientas a utilizar para lograr el producto deseado.

Capítulo 2: Análisis y diseño de la propuesta de solución

En este capítulo se realiza el modelado del negocio, el análisis y diseño de la propuesta de solución, identificándose los requisitos funcionales y no funcionales del sistema, obteniendo la arquitectura base, los patrones de diseño, uso y beneficios de los mismos. La elaboración de los diagramas de procesos de negocio, los diagramas de clases, el modelo conceptual, su diccionario de datos.

Capítulo 3: Implementación y validación de la propuesta de solución

Se realiza la implementación y validación de la propuesta de solución aplicando las pruebas de software correspondiente en cada caso.

³ Recursos Humanos

Capítulo 1: Fundamentación Teórica

Introducción

En este capítulo se analizan diferentes sistemas automatizados que gestionan el pago a trabajadores y se encuentran funcionando en otros países del mundo. Se estudian al mismo tiempo los implantados en Cuba, de los cuales se ha hecho un análisis de las principales dificultades que presentan y funcionalidades que me ayudaran a desarrollar la propuesta de solución, por otra parte, se definen las herramientas, técnicas y metodologías a utilizar para guiar el desarrollo; además, se tratarán los conceptos fundamentales del tema y la propuesta de solución para el mismo.

1.1 Conceptos Asociados

Se hace referencias a los diferentes conceptos utilizados, posibilitando el claro entendimiento por parte del lector a medida que avanza en el trabajo de investigación.

Los **sistemas de pago** son las modalidades de las formas de pago que se adoptan en correspondencia con las características técnico-organizativas del proceso laboral, de la organización del trabajo y de las posibilidades de control de sus resultados (Ministerio de Trabajo y Seguridad Social, 2009).

El **salario** es la retribución monetaria que el trabajador recibe en función de la eficiencia obtenida en el trabajo realizado en correspondencia con el sistema de pago aplicado (Ministerio de Trabajo y Seguridad Social, 2009).

El **salario escala** es el pago por complejidad y responsabilidad aprobado para cada ocupación o cargo, según el grupo de la escala salarial vigente (Ministerio de Trabajo y Seguridad Social, 2009).

El **pago por resultado** se realiza, teniendo en cuenta los niveles de cumplimiento de los indicadores establecidos en los sistemas de pago en cualquiera de sus modalidades. En correspondencia con los resultados alcanzados se obtiene un salario que puede ser igual, superior o inferior al salario de la escala más los incrementos debidamente aprobados y que procedan, según tiempo real trabajado (Ministerio de Trabajo y Seguridad Social, 2009).

El **pago adicional** constituye un pago por trabajar en condiciones laborales anormales, nocturnidad, albergamiento, rotación de turnos, altura; los aprobados centralmente por el Gobierno por concepto de antigüedad, interés económico y social, pagos por maestrías y doctorados, pago adicional por aplicar el Perfeccionamiento Empresarial, otros pagos adicionales por factores extracalificatorios y otros que se decidan por el Gobierno. (RESOLUCIÓN No. 6/2016, 2016)

1.2 Estado del Arte

Se realizó el análisis de un grupo de sistemas que informatizan el departamento de Nóminas y Recursos Humanos.

Sistemas existentes que gestionan los procesos de pago y nóminas a Recursos Humanos La suite ERP Unit4 Business World On!

Proporciona un apoyo integral para todos los aspectos de los procesos organizativos de la gestión de capital humano, desde el ciclo de la contratación hasta la finalización del contrato, con una serie de funciones integradas en el software de gestión de RR.HH⁴ incluyendo:

- Formación y gestión del desempeño
- Nómina
- Remuneración
- Promociones y cambios de personal
- Procedimientos disciplinarios y quejas
- Gastos
- Patrones de cambio
- Gestión de ausencias

Diseñado para ser un software ERP de gestión de capital humano accesible e intuitivo para los usuarios cotidianos, ¡Unit4 Business World On! refuerza la tarea de empresas centradas en personas o servicios, asegurándose de que se puedan realizar las actualizaciones y ajustes continuos, para adaptar el sistema a los requerimientos y procesos específicos en la aplicación de recursos humanos de la organización y para mantenerlos en sintonía con los cambios del negocio (Unit4 Business World, 2018).

ADAM 5

Sistema integral para la eficiente administración de los recursos humanos dentro de las empresas e instituciones. Este sistema tiene como soluciones del negocio ERP, Nómina, Recursos Humanos, Tiempo/Acceso, y Activos Fijos. Incluye buenas prácticas en la administración de los recursos humanos y administración de personal. (ADAM, 2018)

Entre sus principales funcionalidades presenta:

- Implementar fórmulas de pago complejas como demande la empresa
- Corregir y recalcular interactivamente la nómina
- Manejar pagos normales, especiales y extraordinarios
- Definir formatos de pago
- Contabilizar automáticamente la nómina con cualquier sistema financiero.

⁴ Recursos Humanos

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

Siscont 5

Sistema certificado que se aviene a las definiciones y conceptos del Ministerio de la Industria Básica (MINBAS), es desarrollado y comercializado por TECNOMÁTICA y sus acciones contables financieras le han permitido ser utilizado en otras entidades nacionales fuera del entorno del MINBAS. En su versión 5, abarca diversos módulos: contabilidad general, costos, estados financieros, activos fijos, inventarios, almacenes, pagos y nóminas.

Versat-Sarasola

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

A continuación, un listado de funcionalidades comprendidas en dicho sistema:

- Nóminas
- Documentos de ajuste
- Prenóminas
- Descarga de retenciones
- Trabajadores
- Ocupaciones y cargos

ODOO (módulo Recursos Humanos: Gestión de nóminas)

Este módulo permite automatizar la creación de los asientos contables para las nóminas de los empleados. Permite definir varios tipos de complementos (vacaciones pagadas, enfermedad, etc...) y gestionarlos ya sea a nivel de empresa, a nivel de departamento o en una base de caso por caso (por ejemplo: tiempo de compensación por las horas extra). Deja peticiones que pueden ser registradas por los empleados y validadas por sus directores.

Entre sus principales funcionalidades presenta:

- Hojas de Gestión

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

- Seguimiento del tiempo
- Gastos
- Proceso de evaluación
- Contratación
- Informes

ASSETS NS (módulo Recursos Humanos)

Permite controlar toda la información relacionada con Personal y Nóminas. Calcula nóminas a sueldistas, vinculados y contratados. Permite confeccionar nóminas por subsidios y vacaciones, realizar deducciones al salario y emitir análisis estadísticos de personal y nóminas, incluido el registro de salario y tiempo laborado. Se pueden realizar pagos mensuales y quincenales. Software de licencia privada. Alto grado de parametrización y fácil personalización.

Entre sus principales funcionalidades presenta:

- Nominillas
- Prenóminas
- Nóminas de retenciones
- Submayor de vacaciones
- Reintegros de salarios, subsidios y vacaciones
- Pagos de horas extras, subsidios, vacaciones

Actualmente en nuestra universidad se encuentra implementado el sistema de gestión empresarial Assets, anteriormente descrito.

Sobre la base de las consideraciones anteriores a continuación, se realizan por parte del autor una serie de valoraciones.

Mediante el estudio realizado al grupo de sistemas informáticos, antes mencionados, que se encuentran en explotación a nivel mundial y en nuestro país, atendiendo a las características y funcionalidades individuales, se concluyó que estos, independientemente de que fueron desarrollados para entornos específicos, son multiplataforma, brindan amplias ventajas en la gestión de los diferentes procesos de nóminas. Permiten gestionar el tratamiento de la remuneración de los trabajadores y realizar los cálculos relacionados con la formación y distribución del salario.

Por otra parte, tuvo gran significación el estudio y análisis realizado a las principales funcionalidades de sistemas de pago presentes en dichas aplicaciones informáticas, ya que permitieron esclarecer, definir e incluir algunas de ellas en la propuesta de solución,

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

que a su vez son de vital importancia en la gestión del pago, posibilitando posteriormente la realización de las descripciones de los requisitos funcionales.

Sin embargo, debido a que la mayoría fueron desarrollados bajo herramientas y softwares propietarios, de utilizarse alguno de ellos se incurriría en gastos excesivos para nuestra universidad en términos de pago de licencia y soporte. Sistemas como el Unit4 Business y el ODOO a pesar de su adaptabilidad fueron diseñados para empresas capitalistas con modelos de gestión diferentes al cubano, en los módulos definidos para la gestión de la nómina se realizan algunos procedimientos que no están en correspondencia con las normas cubanas establecidas y no se evidencia de forma clara la formación del salario, ya que más bien están enfocados a la gestión de gastos.

1.3 Metodología de Desarrollo

El desarrollo de todo software debe estar guiado por una metodología. De esta depende, en gran medida, que el software tenga la calidad requerida. Existen dos grupos de metodologías: ágiles y tradicionales. No existe una metodología universal para cada tipo de proyecto. Por ello, se define una metodología según las características del equipo de desarrollo, el dominio de aplicación, el tipo de contrato, la complejidad y la envergadura del proyecto.

Dada la necesidad de desarrollar la propuesta de solución en un breve período de tiempo, garantizando además la flexibilidad necesaria en cuanto a la variación de los requisitos y el manejo de los riesgos técnicos, tal como reducir la generación de documentos y artefactos, y la no existencia de un contrato tradicional, siendo el cliente parte del equipo de desarrollo; se hace necesario optar por un enfoque ágil de desarrollo de software en lugar de un enfoque tradicional o pesado. Teniendo en cuenta lo anterior han sido evaluadas varias metodologías que siguen el enfoque ágil de desarrollo de software, tal es caso de Scrum (Extreme Programming and Scrum, 2017) , Agile Unified Process (AUP, por sus siglas en inglés), Extreme Programming (XP) por sus siglas en inglés y AUP-UCI. Finalmente, el autor de la investigación adopta la variación de la metodología AUP para la UCI.

1.3.1 Disciplina Modelado de Negocio

Es la disciplina destinada a comprender los procesos de negocio de la organización. Se comprende cómo funciona el negocio que se desea informatizar para tener garantías de que el software desarrollado va a cumplir su propósito (Metodología de desarrollo para la Actividad Productiva de la UCI, 2015).

Artefactos que genera

En esta disciplina se generan una serie de artefactos que conducen a la captura de

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

requisitos del cliente tales como:

- Modelo Conceptual
- Descripción de Procesos de Negocio
- Descripción de Requisitos por Proceso

Agrupados estos en el escenario número tres, aplicado a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez y su continuidad teniendo presente que este escenario es muy conveniente si se desea representar una gran cantidad de niveles de detalles y la relaciones entre los procesos identificados. Utilizando para ello tecnologías y herramientas tales como:

- **Visual Paradigm 8.0**

Es una herramienta para el desarrollo de aplicaciones utilizando modelado UML ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos. Ofrece un entorno de creación de diagramas para UML. El diseño es centrado en casos de usos y enfocado al negocio.

Para la etapa de diseño del sistema se seleccionó Visual Paradigm por ser una herramienta de modelado multiplataforma que no se inclina por ninguna metodología específica, además de ser un estándar ampliamente utilizado en las empresas para el modelado de software. (Paradigm, 2016)

- **Lenguaje de modelado UML⁵**

El Lenguaje Unificado de Modelado (UML) 2.4 prescribe un conjunto de notaciones y diagramas estándares para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan; posibilitando así visualizar, especificar y documentar los artefactos o toda información que se obtiene o modifica durante un proceso de desarrollo de software, además de poder utilizarse para modelar distintos tipos de sistemas de software, hardware y organizaciones del mundo real (Larman, 2003).

1.3.2 Disciplina Requisitos

El esfuerzo principal en esta disciplina es desarrollar un modelo del sistema que se va a construir. Incluye un conjunto de artefactos que describen todas las interacciones que tendrán los usuarios con el software y que responden a los requisitos funcionales del

⁵Lenguaje Unificado de Modelado.

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

sistema. Se especifican los requisitos funcionales y no funcionales (Metodología de desarrollo para la Actividad Productiva de la UCI, 2015).

Artefactos que genera

- Especificación de requisitos de software
- Descripción de requisitos

Para la obtención de los requerimientos se aplicaron técnicas como:

- **Entrevistas**

Las entrevistas se emplean para recolectar información en forma verbal, a través de preguntas elaboradas por el analista, estas preguntas se les realiza a quienes se encuentren afectado por la aplicación, a usuarios con gran nivel de conocimiento del sistema o a personas que pueden proporcionar datos. Durante la entrevista, la analista conversa con el encuestado para obtener información deseada. El éxito de esta técnica, depende de la habilidad del entrevistador y de su preparación para la misma. Para la realización de las entrevistas se debe coordinar previamente la fecha y hora, y realizar un plan de agenda, en el cual se realiza un punteo del objetivo de la entrevista. Para realizar las entrevistas, es conveniente llevar preparado un cuestionario (Pressman, 2010).

- **Tormenta de ideas**

La tormenta de ideas es una técnica simple para la generación de ideas. Se reúne un conjunto de personas relacionadas al producto y éstas sugieren y exploran sus ideas libremente. Generalmente las reuniones se realizan con 4 a 10 personas; una de ellas deberá actuar como líder para comenzar, pero no para restringir la expresión de los restantes participantes. Se genera una amplia variedad de puntos de vistas, se estimula el pensamiento creativo, se logra construir una imagen más completa del problema, se provee un ambiente social mucho más confortable y de fácil aprendizaje (Pressman, 2010).

- **Revisión de Documentación**

La documentación difícilmente refleja la forma en que realmente se desarrollan las actividades, o donde se encuentra el poder de la toma de decisiones. Sin embargo, puede ser de gran importancia para introducir al analista al dominio de operación y el vocabulario que utiliza (Pressman, 2010).

Una vez obtenidos los requerimientos, se aplicaron las técnicas de validación

- ✓ **Revisión técnica formal:** el equipo de revisión incluye ingenieros del sistema, clientes, usuarios y otros intervinientes que examinan la especificación del

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

sistema buscando errores en el contenido o en la interpretación, áreas donde se necesitan aclaraciones, información incompleta, inconsistencias, requisitos contradictorios, o requisitos imposibles o inalcanzables (Pressman, 2002).

- ✓ **Construcción de prototipos:** se muestra un modelo ejecutable del sistema a los usuarios finales y a los clientes. Éstos pueden experimentar con este modelo para ver si cumple sus necesidades reales (Sommerville, 2002)
- ✓ **Generación de casos de prueba:** los requisitos deben poder probarse. Si las pruebas para éstos se conciben como parte del proceso de validación, a menudo revela los problemas en los requisitos (Sommerville, 2002)

1.3.3 Disciplina Análisis y Diseño

Durante esta disciplina es modelado el sistema para que soporte todos los requisitos. Esto contribuye a una arquitectura sólida y estable que se convierte en un plano para la próxima disciplina. En caso de llevarse a cabo la reutilización de componentes de software ya desarrollados, durante esta disciplina se ajusta el modelado existente a los requisitos actuales (Metodología de desarrollo para la Actividad Productiva de la UCI, 2015)

Artefactos que genera

- Modelo de datos
- Diagrama de clases del diseño
- Diseño de casos de pruebas

Para lograr un adecuado análisis y diseño se utilizan patrones tales como:

Los **patrones GRASP**⁶, estos son guías o principios que sirven para asignar responsabilidades a las clases (Patrones de diseño, 2017)

- **Experto:** se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele ser útil en el diseño orientado a objetos.
- **Creador:** guía la asignación de responsabilidades relacionadas con la creación de objetos. Permite crear instancias de otras clases según la responsabilidad de la misma. Esto posibilita un bajo acoplamiento y mejores oportunidades de reutilización
- **Controlador:** consiste en asignar la responsabilidad de administrar un mensaje de eventos del sistema a una clase que represente: el negocio, la organización global o el sistema global
- **Bajo acoplamiento:** asigna una responsabilidad para mantener bajo acoplamiento

⁶General Responsibility Assignment Software Patterns (Patrones Generales de Software para Asignar Responsabilidades).

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

- **Alta cohesión:** Asigna una responsabilidad de modo que la cohesión siga siendo alta.

Los **Patrones GOF** (gang of four), representan una serie de patrones que permiten ampliar el lenguaje, aprender nuevos estilos de diseño y además se introduce más notación UML, son una descripción de clases y objetos que se comunican entre sí, adaptada para resolver un problema general de diseño en un contexto particular (Larman, 2003) dentro de los patrones utilizados se encuentran:

- **Creacional:** Se encargan de la creación de instancias de los objetos. Abstraen la forma en que se crean los objetos, permitiendo tratar las clases a crear de forma genérica, dejando para después la decisión de que clase crear o cómo crearla (Larman, 2003)
- **Estructural:** Son los que plantean las relaciones entre clases, las combinan y forman estructuras mayores. Tratan de conseguir que los cambios en los requisitos de la aplicación no ocasionen cambios en las relaciones entre los objetos (Larman, 2003)
- **Comportamiento:** Plantea la interacción y cooperación entre las clases. Los patrones de comportamiento estudian las relaciones de llamadas entre los diferentes objetos, normalmente ligados con la dimensión temporal (Larman, 2003)

Se definieron los patrones anteriormente mencionados, gracias a que constituyen soluciones ya probadas, correspondiéndose a la abstracción de un problema que ya alguien más utilizó y le funcionó para resolverlo; la mayoría están pensados para que sean flexibles y extensibles. Estos encapsulan lo que varía, y lo que no cambia, para evitar que futuros cambios impliquen modificar módulos ya existentes en la aplicación. Un vocabulario que los desarrolladores pueden entender a la hora de resolver problemas de diseño, además permiten que todos puedan estar en la misma página en cuanto a la solución de diseño acordada.

Patrones de arquitectura

Según Pressman, los patrones de arquitectura están orientados a representar los diferentes elementos que componen una solución de software y las relaciones entre ellos. A diferencia de los patrones de diseño de software que están orientados a objetos y clases (patrones creacionales, estructurales, de comportamiento, de interacción), los patrones de arquitectura están a un mayor nivel de abstracción. Los patrones de arquitectura forman parte de la llamada Arquitectura de Software (arquitectura lógica de un sistema) (Pressman, 2010).

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

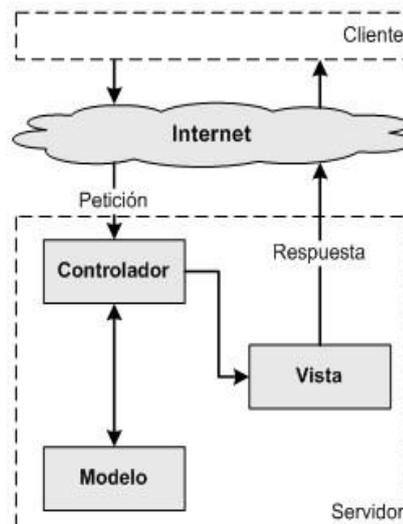


Figura 1 Patrón Vista Controlador (Fabien Potencier, 2016)

Esta separación en capas con responsabilidades bien definidas permite, además de mejorar la organización del código, la posibilidad de representar de diferentes formas la misma información.

El **modelo** es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema).
- Lleva un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero Bath que actualiza los datos, un temporizador que desencadena una inserción, entre otros agentes externos).

El **controlador** es responsable de:

- Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W".

Las **vistas** son responsables de:

- Recibir datos del modelo y mostrarlos al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de "Actualización ()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

1.3.4 Disciplina Implementación

A partir de los resultados del análisis y diseño se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ejecutables y similares. Al reutilizar componentes software ya implementados se lleva a cabo el desarrollo necesario para ajustar a los requisitos actuales y posteriormente realizar la integración de los componentes (Metodología de desarrollo para la Actividad Productiva de la UCI, 2015).

Artefactos que genera

- Se obtiene toda la implementación del módulo Sistema de Pagos.

Utilizando herramientas y tecnologías tales como:

- **Marco de trabajo Odoo 10.0**

La arquitectura de referencia es un diagrama que permite separar los distintos componentes de una solución del datawarehousing⁷ e integrar en una clasificación común los distintos tipos de información necesarios para construir un datawarehousing (ODOO, 2016).

Odoo (conocido anteriormente como OpenERP) es un sistema de ERP integrado de código abierto producido por OpenERP s.a. Actualmente han cambiado el nombre de OpenERP por el de Odoo.

El mismo cumple las 4 libertades del software libre, basado en estándares abiertos y desarrollado con plataformas libres.

Entre las principales características de Odoo se encuentra su escalabilidad, modularidad y flexibilidad. Posee una importante comunidad de desarrolladores que están constantemente ampliando y mejorando el proyecto (amplia documentación, foros, listas de correo y desarrollo comunitario (CEIGE, 2015).

La arquitectura que utiliza Odoo es cliente-servidor donde el servidor se ejecuta independientemente del cliente y maneja la lógica de negocio y comunica con la aplicación de base de datos (CEIGE, 2015).

Odoo también incluye un sistema de reportes con integración con OpenOffice.org, lo que permite personalizar los informes. También hay motores de reportes alternativos utilizando Webkit o Jaspersoft (CEIGE, 2015).

⁷Es una colección de datos orientada a un determinado ámbito, integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza.

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

- **Marco de trabajo OpenObject 1.0**

Es un marco de trabajo de código abierto, inteligente, profesional y rápido en el desarrollo de aplicaciones en Python. Está basado en la arquitectura modelo-vista-controlador, además de poseer Inteligencia de Negocios, Mapeador Relacional de Objetos(ORM), casos de pruebas, motores de flujos de trabajo, grabador de módulos, envases de módulos, entre otros. OpenObject ofrece, en un solo paquete el componente básico para la construcción de una aplicación de negocios: multilinguaje, servicios web, campos traducibles, ingeniería de reportes, PostgreSQL, Python como lenguaje de programación y licencia GNU AGPL v3 (ODOO, 2016)Se utilizó como framework OpenObject por las ventajas que posee al estar integrado dentro de Odo.

- **Lenguaje de Programación Python 2.7.6**

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, funcional. Es un lenguaje interpretado, legible, utiliza tipado dinámico y es multiplataforma (Python, 2018).

Para el desarrollo de la aplicación se utilizó Python debido a que es el lenguaje de programación que utiliza Odo.

- **Entorno de desarrollo integrado PyCharm**

Un entorno de desarrollo integrado, llamado también IDE (sigla en inglés de Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios (PyCharm, 2018).

PyCharm es un IDE o entorno de desarrollo integrado multiplataforma utilizado para desarrollar en el lenguaje de programación Python. Proporciona análisis de código, depuración gráfica, integración con VCS / DVCS y soporte para el desarrollo web con Django, entre otras bondades. Entre las características fundamentales que posee el PyCharm se encuentran el autocompletado, resaltador de sintaxis, herramientas de análisis y refactorización. Posee un depurador avanzado, además de la integración con lenguajes de plantillas como Mako, Jinja2, Django. Soporta entornos virtuales e intérpretes de Python 2.x, 3.x, PyPy, Iron Python y Jython. Posee también compatibilidad con SQLAlchemy (ORM), Google App Engine, Cython (Python, 2018).

Se utilizó el PyCharm porque es el IDE para la programación en Python y tiene como ventaja principal que es multiplataforma. El editor de código inteligente de PyCharm

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

brinda soporte de primera clase para Python. Proporciona la finalización inteligente de códigos, inspecciones de códigos, resaltado de errores sobre la marcha y soluciones rápidas, junto con refactorizaciones automáticas de códigos y capacidades de navegación avanzadas.

- **Lenguaje de Marcas Extensible XML 1.2**

XML, siglas en inglés de Xtensible Markup Language (Lenguaje de marcas extensible), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible. Este lenguaje de marcado auto descriptivo encuentra una aplicación cada vez más amplia como herramienta de transmisión y procesamiento de datos. XML es eficiente en la descripción y definición de datos y, por lo tanto, se usa cada vez más en aplicaciones intensivas en datos. De esta forma, XML es diferente de HTML, que fue diseñado para mostrar datos (XML, 2016).

A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información.

Para el desarrollo de la aplicación se utilizó XML debido a que es el lenguaje de marcas que utiliza Odo para la generación de las vistas y reportes.

- **Herramientas para administrar el Gestor de Base de Datos**

Un SGBD o Sistema Gestor de Base de Datos⁸ es una colección de datos relacionados entre sí, estructurados y organizados, un conjunto de programas que acceden y gestionan esos datos. La colección de esos datos se denomina Base de Datos (BD). Los sistemas gestores de base de datos están diseñados para gestionar grandes bloques de información, que implica tanto la definición de estructuras para el almacenamiento como de mecanismos para la gestión de la información. El SGBD es una aplicación que permite a los usuarios definir, crear y mantener la DB y proporciona un acceso controlado a la misma (Iruela, 2018).

- **PostgreSQL 9.6**

Es un sistema de gestión de bases de datos relacional orientado a objetos, el cual incluye características como herencia, restricciones, tipos de datos, reglas e integridad transaccional. Tiene soporte total para transacciones, disparadores, vistas, procedimientos almacenados, almacenamiento de objetos de gran tamaño. Se destaca en ejecutar consultas complejas, consultas sobre vistas, sub-consultas y joins. Permite

⁸ DBMS: Siglas del inglés Data Base Management System.

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

la definición de tipos de datos personalizados e incluye un modelo de seguridad completo. Utiliza el modelo cliente servidor y es un manejador de base de datos de código abierto liberado bajo la licencia BSD8. PostgreSQL está diseñado para administrar grandes volúmenes de datos (PostgreSQL, 2016).

Se utilizó como sistema gestor de base de datos por las ventajas que posee al estar integrado dentro de Odo.

- **PgAdmin4 v2.0**

Es una aplicación gráfica para gestionar y administrar las bases de datos PostgreSQL. PgAdmin se diseña para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y hace simple la administración. Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Microsoft Windows, Linux, Mac OSX y Solaris (PgAdmin, 2016).

- **Navegador Mozilla Firefox 55.0.3 o superior**

Firefox es un navegador de Internet con interfaz gráfica de usuario desarrollado por la Corporación Mozilla y un gran número de voluntarios. Se empleará por sus ventajas:

- ❖ Es de código libre y multiplataforma.
- ❖ Permite insertar complementos desarrollados por terceros.
- ❖ Posee función de auto completamiento en la barra de navegación, lo cual facilita la navegación por internet.
- ❖ Guarda la contraseña para entrar a un sitio determinado.
- ❖ La restauración de las ventanas y pestañas que se estuvieran usando antes de cerrar el navegador.
- ❖ Seguridad avanzada en la navegación pues permite la identificación de algún sitio web de forma instantánea, posibilita navegar de forma privada y sin registro de lo accedido en internet.
- ❖ Establece conexiones seguras a sitios web.
- ❖ La integración al antivirus que posea la computadora, soporte para varios idiomas y alta configuración.
- ❖ Protección en cuanto a malware se refiera, puesto que, en caso de acceso a un sitio sospechoso, el navegador le notificará y le dará las razones.

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

1.3.5 Disciplina Pruebas Internas

Durante esta disciplina se desarrollan las pruebas del grupo de calidad del centro verificando el resultado de la implementación. Permite identificar posibles errores en la documentación y el software, es decir requisitos que el producto debería cumplir y que aún no los cumple (Metodología de desarrollo para la Actividad Productiva de la UCI, 2015).

Artefactos que genera:

- No conformidades
- Acta de liberación

Pruebas de calidad

Las pruebas son un conjunto de actividades que se planean con anticipación y se realizan de manera sistemática con el objetivo de detectar errores en el software. Hay dos maneras de probar cualquier producto construido: uno si se conoce la función específica para la que se diseñó el producto, se aplican pruebas que demuestren que cada función es plenamente operacional, mientras que se buscan los errores de cada función; dos si se conoce el funcionamiento interno, se aplican pruebas para asegurarse que “todas las piezas encajan”, es decir que las operaciones internas se realizan de acuerdo con las especificaciones, y se han probado todos los componentes internos de manera adecuada. Al primer enfoque de prueba se le denomina prueba de caja negra, y al segundo prueba de caja blanca (Pressman, 2002)

Pruebas unitarias (caja blanca)

La prueba de caja blanca del software se basa en un examen cercano al detalle procedimental. Se prueban las rutas lógicas del software y la colaboración entre componentes, al proporcionar casos de pruebas que ejerciten conjuntos específicos de condiciones, bucles o ambos. Al emplear los métodos de prueba de caja blanca el ingeniero del software podrá derivar casos de prueba que: 1) garanticen que todas las rutas independientes dentro del módulo se han ejercido al menos una vez. 2) ejerciten los lados verdadero y falso de todas las decisiones lógicas. 3) ejecuten todos los bucles en sus límites y dentro de sus límites operacionales y 4) ejerciten estructuras de datos internos para asegurar su validez (Pressman, 2002)

Técnica de prueba: Camino básico

La técnica del camino básico es empleada en las pruebas de caja blanca, la misma tiene como objetivo comprobar que cada camino se ejecute independiente de un componente o programa, obteniéndose una medida de la complejidad lógica del diseño. Para comprobar que cada sentencia de código se ejecuta al menos una vez, se realizan

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

pruebas al código de las funcionalidades más complejas desde el punto de vista de la programación (Pressman, 2010)

Para obtener los casos de prueba a partir la técnica seleccionada se debe construir el grafo de flujo correspondiente al código de la funcionalidad. Luego se determina la complejidad ciclomática $V(G)$ del grafo resultante, la cual es un indicador del número de caminos independientes que existen en un grafo, es decir, es cualquier camino dentro del código que introduce por lo menos un nuevo conjunto de sentencias de proceso o una nueva condición. (Pressman, 2010)

Esta métrica se calcula sobre un grafo y se puede realizar mediante tres formas distintas:

- $V(G) = E - N + 2$, donde E es el número de aristas, y N el número de nodos de la gráfica de flujo.
- $V(G) = R$, donde R es el número de regiones que favorece a estimar el valor de la complejidad ciclomática.
- $V(G) = P + 1$, donde P es el número de nodos predicados incluidos en el grafo.

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

Pruebas funcionales (caja negra)

Las pruebas de caja negra, también denominadas, pruebas de comportamiento, se concentran en los requisitos funcionales del software. Permiten derivar conjuntos de condiciones de entrada que ejercitarán por completo todos los requisitos funcionales del programa. (Pressman, 2010)

La prueba de caja negra intenta encontrar errores de las siguientes categorías (Pressman, 2010):

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

Técnica de prueba: Partición de equivalencia

La partición equivalente es un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Los casos de pruebas que se diseñan para este tipo de técnica se basan en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada. (Pressman, 2010)

Las clases de equivalencia se definen de acuerdo a las siguientes directrices:

- Si una condición de entrada especifica un rango, se define una clase de equivalencia válida y dos no válidas.
- Si una condición de entrada requiere un valor específico, se define una clase de equivalencia válida y una no válida.
- Si una condición de entrada especifica un miembro de un conjunto, se define una clase de equivalencia válida y una no válida.
- Si una condición de entrada es lógica, se define una clase de equivalencia válida y una no válida

Al aplicar dichas directrices en la derivación de clases de equivalencia, se desarrollarán y ejecutarán los casos de prueba por cada objeto de los datos del dominio de entrada

1.4 Validación de la investigación mediante Juicio de Expertos: Método Delphi

El método Delphi es una metodología estructurada para recolectar sistemáticamente juicios de expertos sobre un problema a resolver, posteriormente procesar la

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

información y a través de recursos estadísticos, construir un acuerdo general de grupo (García Valdés, y otros, 2013)

Este método es considerado como uno de los métodos subjetivos de pronosticación más confiables. El procesamiento estadístico y matemático de la información es la característica más importante del método que lo diferencia del resto de los métodos de pronosticación de base subjetiva, ya que la decisión final que toma el investigador es un criterio fuertemente avalado por la experiencia y conocimiento del colectivo consultado, así como por indicadores objetivos (Win2PDF, 2015)

Para la aplicación del método se tienen en cuenta dos aspectos fundamentales, primero la selección de los expertos y luego la elaboración y análisis del cuestionario a aplicar. Atendiendo a los aspectos anteriormente descritos (García Valdés, y otros, 2013) plantean las siguientes fases:

I. Preparatoria

- Selección de expertos.
- Preparación del instrumento.
- Decisión de la vía de consulta.

II. Consulta

- Rondas de consulta.
- Procesamiento estadístico.
- Retroalimentación.

III. Consenso

- Construcción de consenso.
- Reporte de resultados.

Por su importancia se realizará un énfasis en la fase de consulta, específicamente en el procesamiento estadístico, ya que dicha fase incluye el tratamiento matemático y estadístico en el cuestionario aplicado. Proponiendo (Win2PDF, 2015) el uso del procedimiento:

1. Determinación de la media aritmética por pregunta:

$$\bar{C}_j = \frac{\sum_{i=1}^{m_j} C_{ij}}{m_j}$$

Donde:

m: cantidad de expertos.

n: cantidad de preguntas.

m_j: cantidad de expertos que evalúan la pregunta j.

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

C_{ij} : evaluación en puntos de la pregunta j realizada por el experto i .

2. Grado de concordancia de los expertos por pregunta:

$$\sigma_j^2 = \frac{\sum_{i=1}^{m_j} (C_{ij} - \bar{C}_j)^2}{m_j - 1}$$

A partir de la fórmula anterior para determinar la varianza, se determina el coeficiente de variación (V_j), este coeficiente es una medida del grado de concordancia de los expertos por cada pregunta, donde mientras mayor sea el valor de V_j menor será el grado de concordancia de los expertos.

$$V_j = \frac{\sqrt{\sigma_j^2}}{c_j}$$

Conclusiones Parciales

- ✓ En consecuencia del estudio realizado en este capítulo se obtuvo como resultado la realización del diseño teórico de la investigación, de igual forma quedando evidentes los principales conceptos relacionados con el tema.
- ✓ Como resultado del estudio del módulo de Nóminas de Odoo se demostró la necesidad de realizar una personalización debido a la poca adaptación a las normas y políticas establecidas en correspondencia con el sistema de pago UCI. Además, se analizaron algunos de los sistemas existentes en la actualidad, permitiendo tomar en cuenta algunas funcionalidades y aspectos positivos de cada uno para la construcción de la propuesta de solución.
- ✓ De acuerdo con el análisis de las herramientas de desarrollo se decidió utilizar la metodología AUP en su versión UCI como guía para el desarrollo del software.

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

CAPÍTULO 2: Análisis y Diseño de la propuesta de solución

Introducción

Antes de comenzar el desarrollo del módulo es necesario comprender el entorno de trabajo, bajo el estudio de los procesos que rigen las actividades a implementar. Primeramente, se realiza la modelación del negocio, durante esta etapa se definen las reglas del negocio y los procesos implicados en el mismo con sus respectivas descripciones, así como las entidades que se relacionan entre sí, posterior a ello se realiza el diseño de la propuesta, elaborándose el modelo conceptual y su diccionario de datos, se definen y analizan los requerimientos funcionales para dar paso a la solución del problema y la especificación de los mismos, estas serían las funcionalidades a implementar en el sistema.

Período de pago

El período de evaluación de los resultados y del pago adicional especial será mensual, efectuándose este último en las fechas acordadas para el pago del salario de los trabajadores.

En el caso de los dirigentes, personal docente y especialistas de nivel superior, el período de evaluación será trimestral, a los efectos de poder comprobar con mayor precisión los resultados alcanzados, efectuándose el pago mensual sobre la base del trimestre vencido (Ministerio de Trabajo y Seguridad Social, 2009)

Cuantías de pago adicional

Las cuantías del pago adicional (C) son fijas, diferenciadas por grupos de trabajadores y podrán ser modificadas a partir del índice resultante de la evaluación de los resultados del trabajo expresado en factor de desempeño (FD), de manera que cada trabajador recibirá como pago adicional (PA) el resultado de:

$$PA = C \times FD$$

El índice de la evaluación de los resultados, expresado en factor de desempeño (FD) será:

- ✓ Desempeño Laboral Superior 1.00
- ✓ Desempeño Laboral Adecuado 0.60
- ✓ Desempeño Laboral Deficiente 0

La evaluación de los resultados del trabajo se realiza por el jefe al que se subordina directamente cada trabajador, y la administración de la UCI establece el procedimiento para efectuarla (Ministerio de Trabajo y Seguridad Social, 2009)

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

Precisiones del sistema de pago adicional UCI

1. Tendrán derecho a recibir el pago adicional, de acuerdo a los resultados de las evaluaciones de la actividad desempeñada, los trabajadores que se encuentren en las situaciones siguientes:
 - Movilizados en tareas relacionadas con la preparación para la defensa
 - Prestando servicio como jueces legos
 - En prestación de servicios en otros centros y tareas y que no estén acogidos a sistemas de pago o de estimulación en los mismos.
2. Los trabajadores que se encuentren de vacaciones reciben el estímulo, igual que el resto del salario, en correspondencia con los días trabajados.

En el caso de evaluación deficiente no se recibe el pago adicional.

2.1 Reglas del Negocio

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algunos aspectos del negocio. Ver documento CEIGE-ODOO Reglas_de_Negocio.odt

Reglas textuales.

Pagos de salarios: Para realizar estas operaciones es necesario registrarse por las legislaciones vigentes actualmente sobre los Sistemas de Pagos.

Términos: En el documento se tratan conceptos asociados al contexto del negocio: sistemas de pagos, salario, pagos adicionales.

Reglas del modelo de datos

Controla que la información básica almacenada para cada atributo o propiedad de un concepto sea válida.

- El sexo de un trabajador solo puede ser masculino o femenino
- Las fechas deben ser siempre válidas
- Las escalas salariales deben estar en correspondencia con su grupo de complejidad
- Los cargos deben establecerse teniendo en cuenta su grupo de complejidad y categoría ocupacional

Reglas de relación

- Las nóminas deben realizarse por el funcionario encargado de la tarea en cada área
- Las certificaciones de nóminas podrán realizarse una vez concluido el proceso de confección de las nóminas

Reglas de derivación

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

Formación del salario:

- El salario se puede calcular a partir de las distintas líneas que lo componen, teniendo en cuenta el cargo, su grupo de complejidad y escala salarial correspondiente
- El pago adicional, se puede calcular a partir PA (pago adicional) = C (cuantía) x FD (factor de desempeño)

Una vez definidas las reglas del negocio, se realiza el modelado de la propuesta de solución.

2.2 Modelado de la Propuesta de Solución

La propuesta es personalizar el módulo de gestión, adicionándole características y funcionalidades necesarias para alcanzar el producto deseado, que llevará a cabo la integración del departamento de nómina de la UCI, trayendo consigo que se evite la pérdida de la información que se maneja en el mismo, que los documentos sean utilizados correctamente, ya que los mismos son una parte importante para el pago a los trabajadores y evitar que estos se almacenen repetidamente en distintas áreas de la empresa. Dicho tratamiento de la información permitirá ver la estructura del salario, lo cual no es más que tener un control sobre la forma de pago que se está utilizando.

2.2.1 Descripción de los Procesos de Negocio

Por las consideraciones anteriores a continuación, se realizan las descripciones correspondientes de cada proceso de negocio definido en la propuesta de solución.

2.2.2 Pago Salario Básico

Este proceso tiene las configuraciones del cumplimiento de los indicadores y de las penalizaciones, así como los pagos por resultados y los trabajadores abarcados, los cuales estarán asociados a un tipo de pago, un grupo de complejidad y una escala salarial. Para ello es necesario antes confeccionar la prenómina, en este reporte se recogen todas las incidencias y requisitos a evaluar de cada trabajador durante el período a realizar el pago.

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

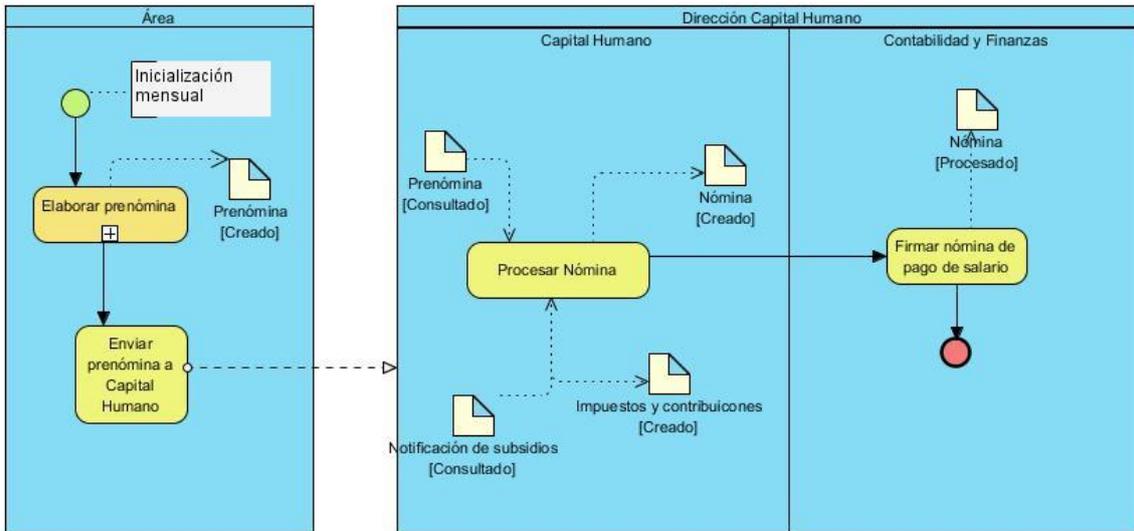


Figura 2 Descripción de proceso Pago de salario básico

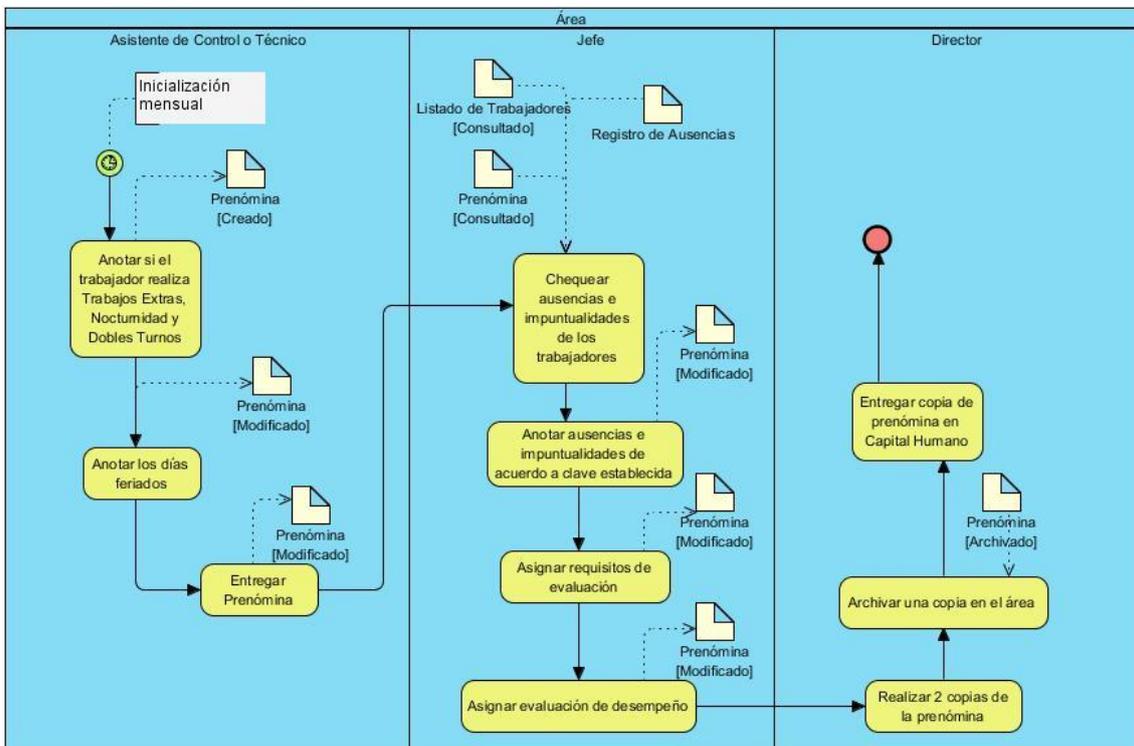


Figura 3 Descripción de subproceso Elaboración de pre nómina

2.2.3 Pago Salario Adicional

Este proceso tiene las configuraciones de los indicadores y requisitos a cumplir para recibir el pago adicional ajustado de la UCI, atendiendo a una evaluación de desempeño.

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

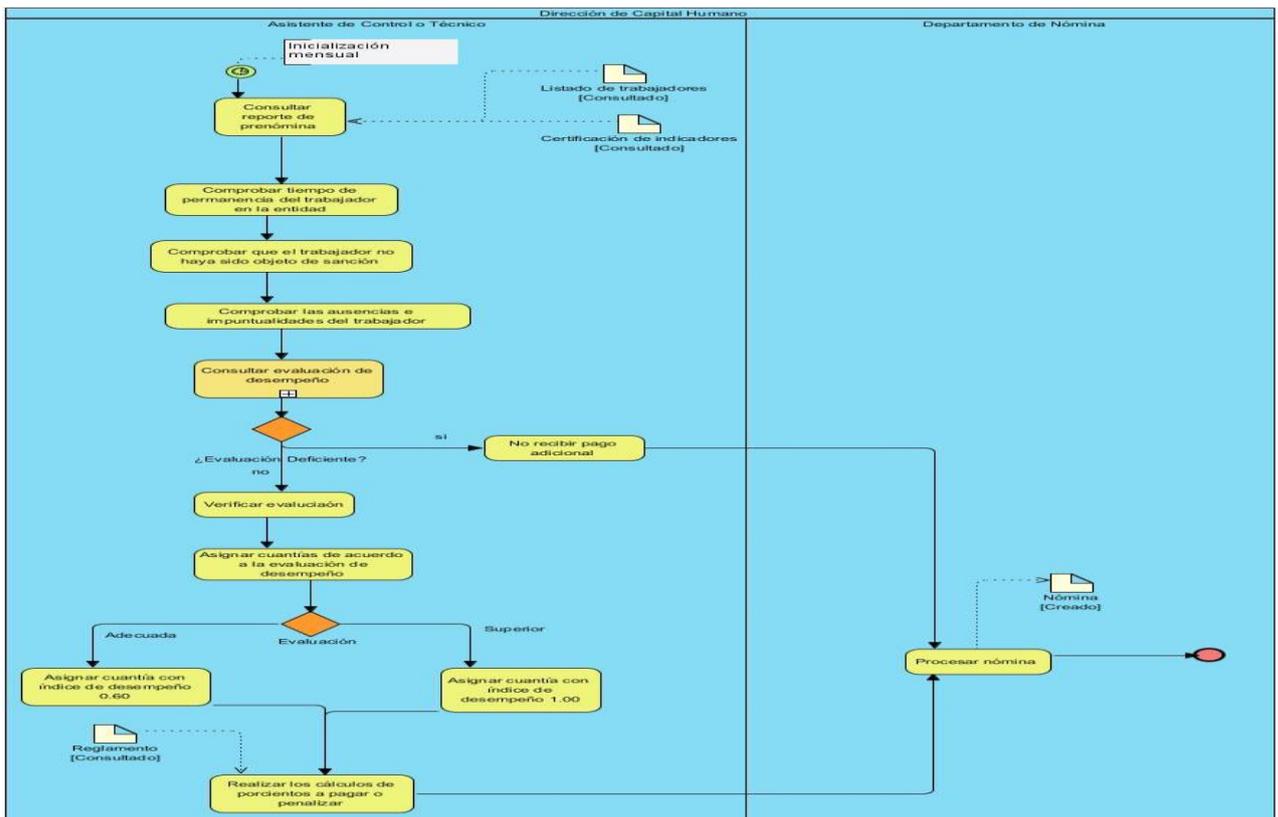


Figura 4 Descripción de proceso Pago Adicional

2.2.4 Evaluación del Desempeño

De este proceso se obtendrá una evaluación de desempeño atendiendo a las evidencias, competencias laborales y el convenio de trabajo, para con ello establecer un índice de evaluación del trabajador, expresado en factor de desempeño siendo este aplicable en la formación del salario adicional.

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

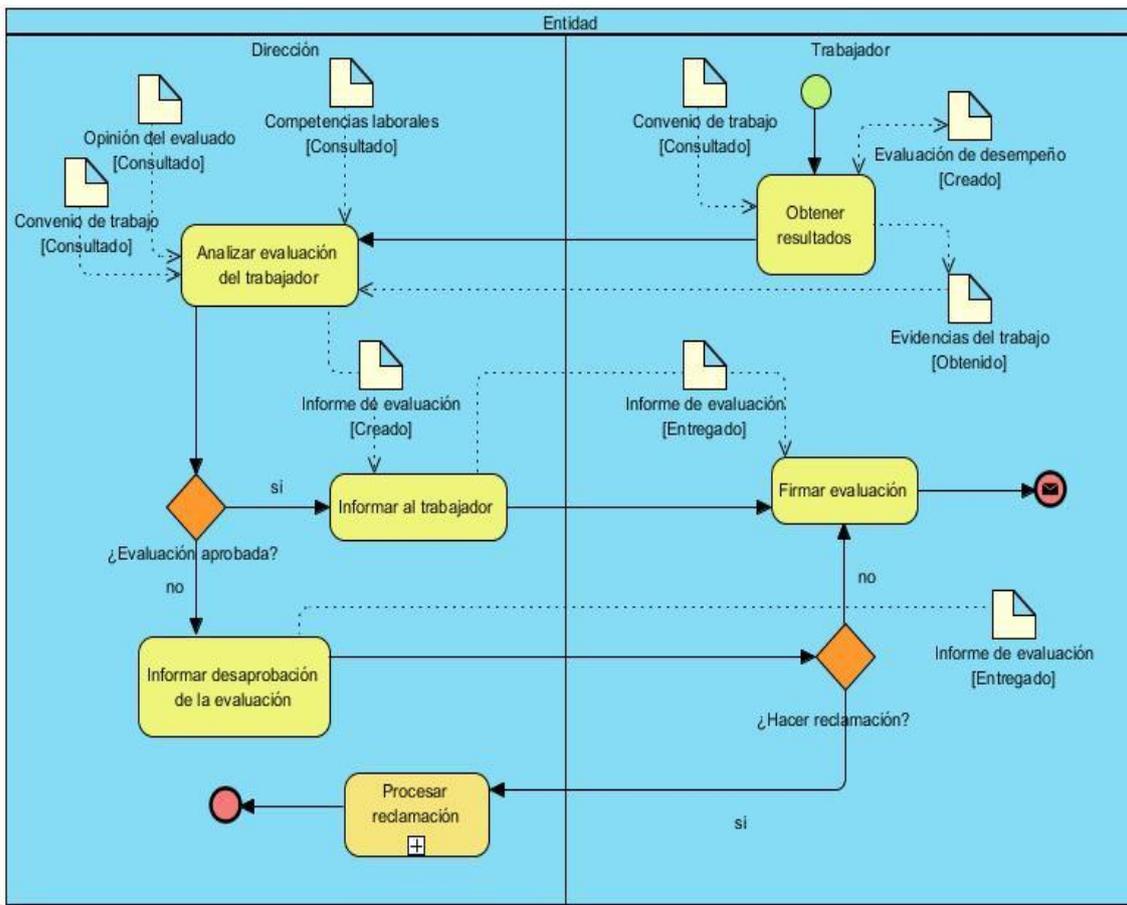


Figura 5 Descripción de proceso Evaluación de desempeño

2.3 Modelo Conceptual

El Modelo conceptual es una representación visual de los conceptos u objetos que se manejan en el dominio del sistema. Para un mejor entendimiento de los conceptos del modelo conceptual, consultar en el expediente el documento CEIGE-ODOO Modelo Conceptual.

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

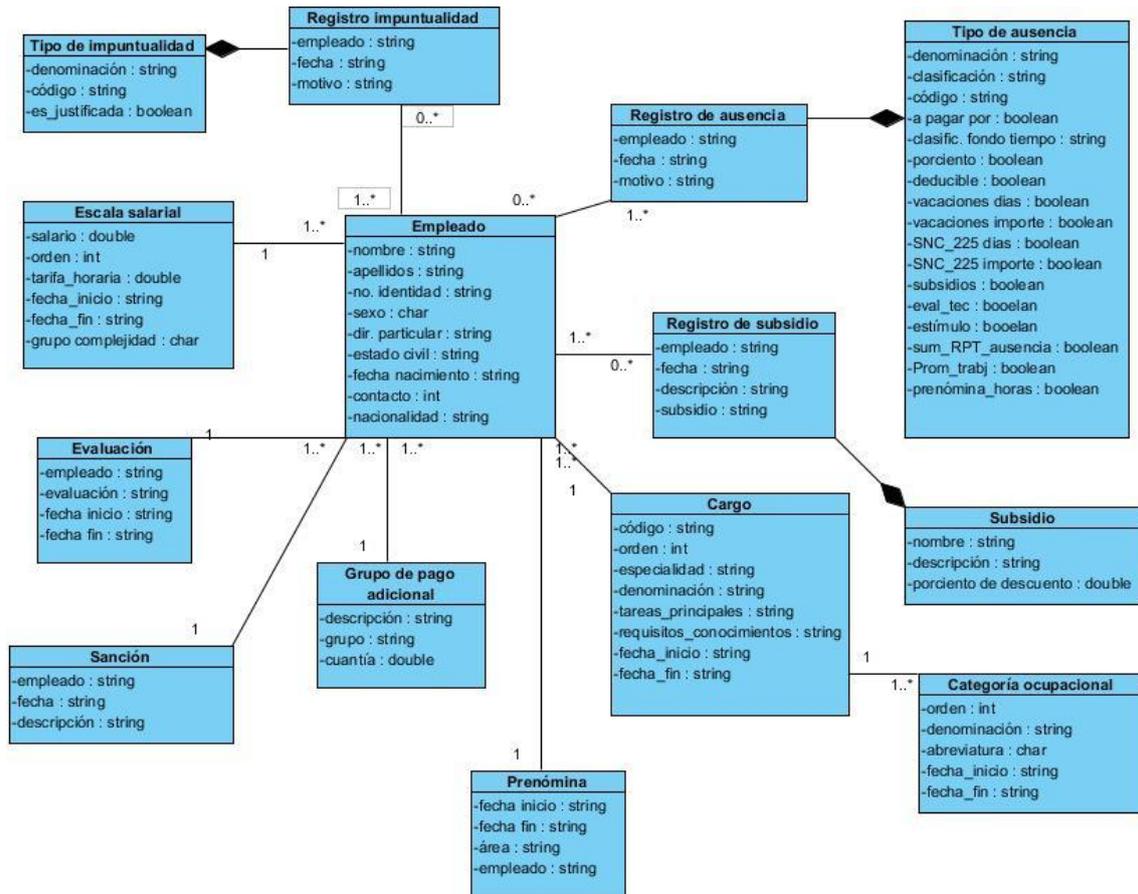


Figura 6 Modelo Conceptual

2.4 Diccionario de Datos

Tabla 1 Diccionario de datos

Descripción						
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
Empleado	Persona que se desempeña en un cargo determinado realizando tareas sobresalientes,	Cadena de caracteres	No	Si	Cadena de caracteres	Campos nulos, caracteres extraños

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

	el cual recibe un salario a cambio						
Grupo de pago adicional	Grupo de retribución monetaria adicional que se le asigna a un trabajador a parte de su salario.	de	Numéri co	No	Si	Cadena de caracteres	Campos nulos, caracteres extraños
Ausencia	Abandono del puesto de trabajo y de los deberes ajenos al mismo, incumpliendo las condiciones establecidas en el contrato de trabajo.	del	Cadena de caracteres	No	Si	Cadena de caracteres	Campos nulos, caracteres extraños y espacios
Evaluación	Valoración de los resultados obtenidos en la realización de las tareas definidas	de	Cadena de caracteres	No	Si	Cadena de caracteres	Campos nulos
Prenómina	Artefacto donde se resumen todas las incidencias y evaluaciones de los trabajadores en un período	donde		No	Si	Cadena de caracteres	Campos nulos

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

	definido de tiempo					
Sanción	Medida disciplinaria aplicada al trabajador por incumplimiento de sus labores	Cadena de caracteres	No	Si	Cadena de caracteres	Campos nulos
Cargo	Responsabilidad laboral asignada a un trabajador	Cadena de caracteres	No	Si	Cadena de caracteres	Campos nulos, caracteres extraños
Categoría ocupacional	Categoría asociada a un cargo	Cadena de caracteres	No	Si	Cadena de caracteres	Campos nulos, caracteres extraños
Impuntualidad	Falta de puntualidad a algo	Cadena de caracteres	No	Si		Campos nulos, caracteres extraños

2.5 Requisitos del Sistema

Un requerimiento o requisito es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste. Son la pieza fundamental en un proyecto de desarrollo de software, debido a que marcan el punto de partida para actividades como la planeación, así como la definición de recursos necesarios y la elaboración de cronogramas. (Pressman, 2010)

2.5.1 Técnicas Utilizadas para la Identificación de Requisitos

Con el objetivo de minimizar las dificultades que se presentan en la captura de requisitos, se aplican un conjunto de técnicas para la obtención de la información. El propósito general es obtener una descripción correcta de lo que debe hacer el sistema y delimitar su alcance. Estos juegan un papel importante durante el ciclo de vida de un proyecto, ya que establece las características que debe poseer el sistema, así como las condiciones que debe cumplir. Se clasifican en funcionales y no funcionales. A continuación, se detallarán algunas de estas técnicas (Sommerville, 2002)

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

Entrevista

La entrevista es de gran utilidad para obtener información cualitativa como opiniones, o descripciones subjetivas de actividades. Se entrevistó a la analista del proyecto para un mejor entendimiento de los procesos de negocio, por otra parte, se entrevistaron a especialistas de capital humano, realizando un estos un aporte considerable para la captura de requisitos.

Tormenta de ideas

Es una herramienta de trabajo grupal que facilita el surgimiento de nuevas ideas sobre un tema o problema determinado. Se utilizó la tormenta de ideas para la captura de requisitos, la agrupación de ellos y su esclarecimiento.

Revisión de Documentación

Esta técnica permitió en gran medida la captura y definición de los requisitos, pues se estudiaron diferentes legislaciones, documentos y el manual de normas y procedimientos del sistema Asset.

2.5.2 Requisitos Funcionales

Luego de la aplicación de las diferentes técnicas para la obtención de los requisitos se definieron 70 requisitos funcionales, resumiéndose en 14 agrupaciones de requisitos

Una agrupación está compuesta por los siguientes requisitos:

RF1. Gestionar cargo

- Adicionar cargo
- Modificar cargo
- Eliminar cargo
- Buscar cargo
- Listar cargos

RF2. Gestionar ausencias

RF3. Gestionar categoría ocupacional

RF4. Gestionar empleado

RF5. Gestionar escala salarial

RF6. Gestionar evaluaciones

RF7. Gestionar grupo de pago adicional

RF8. Gestionar impuntualidades

RF9. Gestionar pre Nómina

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

RF10. Gestionar registro de subsidio al trabajador

RF11. Gestionar sanciones

RF12. Gestionar subsidios

RF13. Gestionar tipo de ausencia

RF14. Gestionar tipo de impuntualidad

Para mejor comprensión ver documento CEIGE-ODOO Especificación_de_requisitos_de_software.odt de la carpeta de proyecto.

2.5.3 Descripción de Requisitos Funcionales

La descripción de los requisitos funcionales del módulo Sistema de Pagos permitió realizar una descripción textual de cada uno de los requisitos identificados, donde se reflejan los flujos por los que transita el requisito, la información que muestra, así como las restricciones que posee. A continuación, se muestra un ejemplo de descripción del requisito Adicionar Cargo, perteneciente a la Agrupación de requisitos Gestionar Cargo:

Tabla 2 Descripción de Requisito Adicionar Cargo Civil

Precondiciones	Se ha adicionado algún empleado en el sistema.
Flujo de eventos	
Flujo básico Adicionar cargo	
1.	Se introducen los datos del cargo: Nombre Puesto Código Abreviatura Categoría Ocupacional Responsabilidad Escala Salarial Requisitos de Conocimiento Resolución Salario
2.	El sistema valida (ver validación 1) los datos introducidos.
3.	Si los datos son correctos el sistema los registra.

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

4.	El sistema confirma el registro de los datos.
5.	Se listan los cargos existentes. Listar cargos; en la agrupación Gestionar cargos.

Pos-condiciones

1.	Se registró en el sistema un nuevo cargo.
2.	Se asoció el cargo a una categoría ocupacional, un grupo de complejidad, una escala salarial.

Flujos alternativos

Flujo alternativo 3.a Información errónea

1	El sistema señala los datos erróneos y permite corregirlos.
2	El usuario corrige los datos.
3	Volver al paso 2 del flujo básico.

Pos-condiciones

1	N/A
---	-----

Flujo alternativo 3.b Información incompleta

1	El sistema señala los datos vacíos y permite corregirlos.
2	El usuario corrige los datos.
3	Volver al paso 2 del flujo básico.

Pos-condiciones

1	N/A
---	-----

Flujo alternativo *.a El usuario cancela la acción

1	Concluye el requisito.
---	------------------------

Pos-condiciones

1	No se registran los datos.
---	----------------------------

Validaciones

1	Se validan los datos según lo establecido en el Modelo conceptual ODOO-ERP-FENIX
---	--

Conceptos	Cargo	Visibles en la interfaz:
		Nombre
		Puesto
		Código
		Abreviatura
		Categoría Ocupacional
		Responsabilidad
		Escala Salarial

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

		<p>Requisitos de Conocimiento</p> <p>Resolución</p> <p>Salario</p> <p>Utilizados internamente:</p> <p>N/A.</p>
	Abreviatura	<p>Visibles en la interfaz:</p> <p>Denominación</p> <p>Utilizados internamente:</p> <p>N/A.</p>
	Categoría Ocupacional	<p>Visibles en la interfaz:</p> <p>Denominación</p> <p>Utilizados internamente:</p> <p>N/A.</p>
	Responsabilidad	<p>Visibles en la interfaz:</p> <p>Denominación</p> <p>Utilizados internamente:</p> <p>N/A.</p>
	Escala Salarial	<p>Visibles en la interfaz:</p> <p>Denominación</p> <p>Utilizados internamente:</p> <p>N/A.</p>
	Requisitos de Conocimiento	<p>Visibles en la interfaz:</p> <p>Denominación</p> <p>Utilizados internamente:</p> <p>N/A.</p>
	Resolución	<p>Visibles en la interfaz:</p> <p>Denominación</p> <p>Utilizados internamente:</p> <p>N/A.</p>
	Salario	<p>Visibles en la interfaz:</p> <p>Denominación</p> <p>Utilizados internamente:</p> <p>N/A.</p>

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

Requisitos especiales	N/A
Asuntos pendientes	N/A

Prototipo elemental de interfaz gráfica de usuario

The screenshot shows a window titled "Cargo/ Nuevo" with a standard Windows-style title bar. At the top left, there are two buttons: "Guardar" and "Descartar". The form contains the following fields and controls:

- Nombre:** A text input field.
- Responsabilidad:** A dropdown menu.
- Categoría ocupacional:** A dropdown menu.
- Abreviatura:** A text input field.
- Puesto:** A text input field.
- Grupo:** A dropdown menu.
- Código:** A text input field.
- Resolución:** A text input field.
- Salario:** A text input field with the value "0,00" pre-filled.
- Requisitos conocimiento:** A large empty text area.

Figura 7 Prototipo de Interfaz de Requisito Adicionar Cargo

El resto de información sobre la descripción del requisito anteriormente mostrado se encuentra en el documento entregable CEIGE-ODOO Descripción de la agrupación de requisitos Gestionar Cargo.odt. Los requisitos funcionales del módulo Sistema de Pago se encuentran descritos en los documentos entregables.

2.5.4 Requisitos No Funcionales

Los requisitos no funcionales especifican las propiedades o cualidades que debe tener la solución a desarrollar. Representan las características que hacen. Al producto atractivo, usable, rápido o confiable. Estos requisitos pueden marcar la diferencia entre un producto bien aceptado y otro con poca aceptación. Los requisitos no funcionales del módulo Sistema de Pago se encuentran descritos en el expediente de proyecto, documento entregable CEIGE-ODOO Especificación_de_requisitos_de_software.odt.

A continuación, se listan algunos de ellos:

Tabla 3 Requisitos no funcionales

No	Requisitos no funcionales
Seguridad	

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

RNF1	<p>El sistema manejará la seguridad de acceso y administración de usuarios mediante el otorgamiento de privilegios y roles, así como la asignación de perfiles.</p> <p>Se concederá acceso al sistema a partir de un nombre de usuario y una contraseña.</p> <p>El sistema concederá acceso a cada usuario autenticado solo a las funciones que le estén permitidas, de acuerdo a la configuración del sistema.</p>
Usabilidad	
RNF2	<p>El idioma de todas las interfaces de la aplicación será el español.</p> <p>La salva de información se hará sólo cuando la información básica del concepto asociado esté completa, de no estarlo el usuario recibirá una notificación de que son necesarios dichos datos y no se continuará el flujo.</p> <p>Los errores cometidos por el usuario les serán notificados y los mensajes incluirán sugerencias de las posibles soluciones.</p>
Fiabilidad	
RNF3	<p>El sistema estará disponible durante 8 horas, los 7 días de la semana, los 365 días del año.</p> <p>El sistema tendrá un respaldo de la información diaria, permitiendo la recuperación ante la pérdida parcial o total de la información.</p>
Mantenimiento	
RNF5	El sistema debe ser fácil de mantener después de desarrollado.
Software	
RNF6	<p>Plataforma de desarrollo Odoon 10.0</p> <p>Se empleará como Gestor de Base de Datos, PostgreSQL 9.3.</p> <p>El sistema se desarrollará con tecnología Python 2.7.X.</p> <p>Se empleará como Servidor de Aplicaciones Web, Apache 2.2.9.</p> <p>Para el acceso de los clientes solo basta tener una computadora con navegador Google Chrome 9.0 en adelante o Firefox 4.0 en adelante.</p>
Hardware	

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

RNF7	Sistemas operativos compatibles: Windows XP o superior, Linux. El sistema para su instalación en las máquinas clientes requiere: Procesador: 1.40 GHZ. RAM: 512 MB (recomendado 1024 GB). Tarjeta de Red: 1. El sistema para su instalación en el servidor de aplicación requiere: Procesador: 3.00 GHZ. RAM: 1GB (recomendado 4GB). Disco duro: 160 GB (recomendado 500GB).
------	--

2.5.5 Técnicas de Validación de Requisitos

La validación de requisitos examina las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos, y que el resultado del trabajo se ajusta a los estándares establecidos para el proceso, el proyecto y el producto (Pressman, 2002)

A continuación, se relacionan las técnicas de validación de requisitos:

Construcción de prototipos: se obtuvieron 70 prototipos de interfaz que permitieron tener una visión clara de cada funcionalidad del sistema, para ello se utilizó la herramienta case Visual Paradigm.

Generación de casos de prueba: se realizaron 70 diseños de casos de pruebas con el objetivo verificar el cumplimiento de cada requisito funcional.

Revisión técnica formal: se realizó una revisión todos los requisitos aceptándose los 70 requisitos funcionales definidos para dar solución a la propuesta.

De estas se seleccionaron las técnicas construcción de prototipos, generación de casos de prueba y revisión técnica formal para validar los requisitos.

2.6 Diseño de la Propuesta de Solución

Durante el desarrollo de esta disciplina se modela el sistema, teniendo en cuenta la arquitectura, para que soporte todos los requisitos tanto funcionales como no funcionales. De este modo se propicia el desarrollo de una arquitectura sólida para el sistema y la adaptación del diseño para que sirva de base a la etapa de implementación.

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

2.6.1 Patrones de Diseño

Para propiciar la robustez y flexibilidad, se utilizaron en la elaboración del diseño los siguientes patrones **Grasp**:

- Experto: en la presente investigación se pone en práctica en aquellas clases que cuentan con la información necesaria para cumplir una responsabilidad que se le haya asignado, de acuerdo al tipo de información que manejan. Los componentes diseñados están compuestos por clases controladoras, modelo y de entidad, que realizan funciones determinadas de acuerdo a la información que gestionan
- Creador: aplicado en las clases controladoras, las cuales son responsables de crear el objeto de los modelos y estos a su vez de las entidades
- Bajo acoplamiento: su aplicación mantiene una mínima relación entre las clases, posibilitando que, en caso de producirse modificaciones en alguna de ellas, se tenga poca repercusión en la demás, potenciando la reutilización y disminuyendo la dependencia entre ellas
- Alta cohesión: Con la utilización de este patrón se mejoran la claridad y la facilidad con que se entiende el diseño. Se simplifica el mantenimiento y las mejoras en funcionalidad

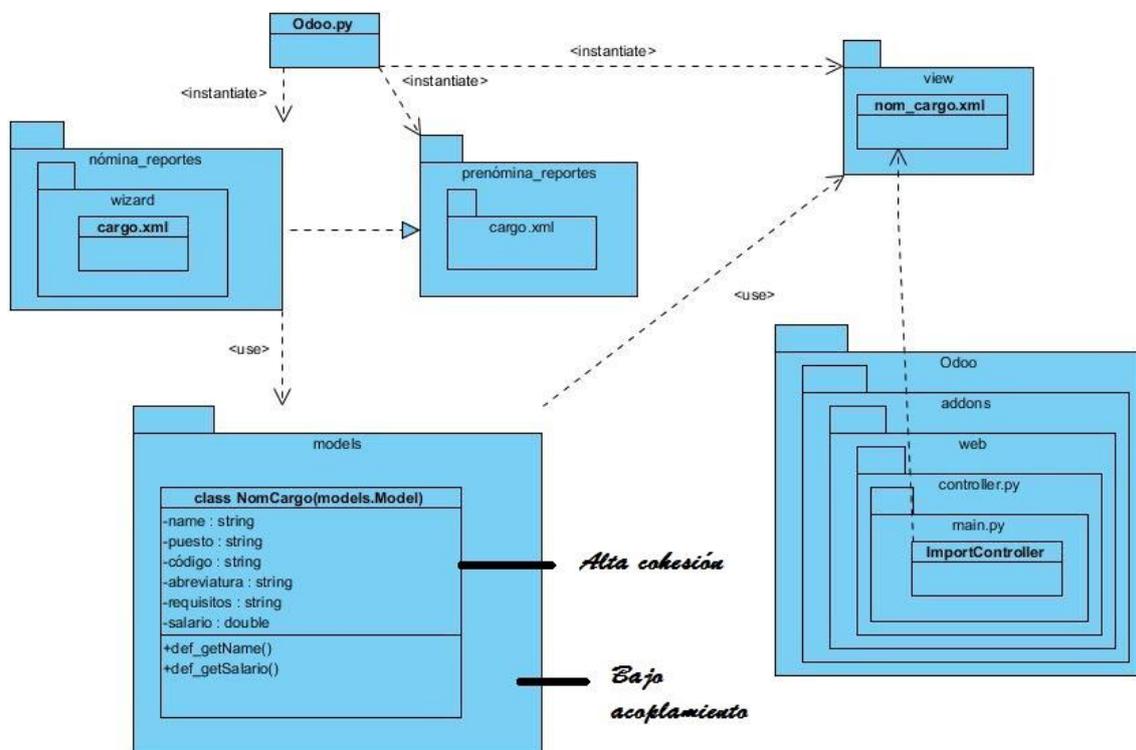


Figura 8 Diagrama de clases de diseño Gestionar Cargo

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

Patrón GOF

Los patrones de creación muestran la guía de cómo crear objetos cuando sus creaciones requieren tomar decisiones.

Adaptador: El patrón adaptador (*adapter*) se utiliza para transformar una interfaz en otra, de tal modo que una clase que no pueda utilizar la primera haga uso de ella a través de la segunda.

La Figura 7 muestra cómo se crea una nueva vista de la clase hr.Employee. A esta nueva vista se le añaden y eliminan los atributos necesarios para que sea usada por la clase Empleado.

```
<record model="ir.ui.view" id="empleado_form">
  <field name="name">Empleado</field>
  <field name="model">hr.employee</field>
  <field name="inherit_id" ref="hr.view_employee_form"/>
  <field name="arch" type="xml">
    <!-- pag 1 Información Pública -->
    <!-- para agragar campos -->
    <xpath expr="//field[@name='job_id']" position="after">
      <field name="cargo"/>
    </xpath>
    <!-- boton a la parte derecha superior -->
    <xpath expr="//button[@name='toggle_active']" position="after">
      <button type="object" class="oe_stat_button" id="absence_button" icon="fa-pencil-square-o"
        name="open_absence_history">
        <field name="absence_total" widget="statinfo" modifiers="{ 'readonly': true }"/>
      </button>
    </xpath>
    <xpath expr="//field[@name='work_email']" position="after">
      <field name="no_expediente"/>
    </xpath>
    <xpath expr="//page[@name='hr_settings']" position="after">
      <page name="pago_adicional" string="Pago adicional">
        <group>
          <field name="pago_adicional"/>
        </group>
      </page>
    </xpath>
  </field>
</record>
```

Adaptador

Figura 9 Patrón GOF Adaptador

2.6.2 Patrón Arquitectónico

En los diagramas de clases diseñados se emplea el patrón arquitectónico modelo vista controlador (MVC). En el caso de la presente investigación, los archivos.xml representan la vista, y el modelo está representado por las clases de la lógica del negocio y las de dominio.

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

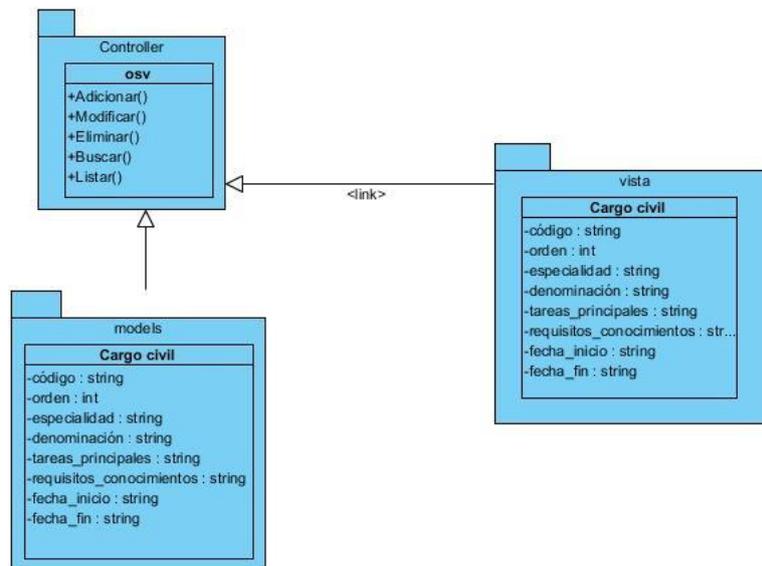


Figura 10 Patrón Arquitectónico Modelo Vista Controlador

Modelo: Objetos Python cuyos datos son almacenados en una base de datos PostgreSQL. Los modelos se encuentran dentro de la carpeta models. El mapeo de la base de datos es gestionado automáticamente por Odo, y el mecanismo responsable por esto es el modelo objeto relacional, (ORM - object relational model).

Vista: Las vistas en Odo manejan la presentación visual de los datos representados por el modelo a través de archivos XML. Se encuentran almacenadas dentro de la carpeta views.

Controlador: En Odo el controlador se manifiesta a través del modelo controller.py, presente en el archivo base_import, el cual es el encargado de hacer peticiones al modelo cuando se hace alguna solicitud de la información por parte del cliente.

2.6.3 Validación del Diseño

La evaluación de un producto, mediante métricas, es un aspecto fundamental a tener en cuenta, aunque las métricas del producto del software no suelen ser absolutas, brindan la posibilidad de evaluar la calidad a partir de varias reglas definidas claramente.

Métricas para la validación del diseño

La realización de métricas en la ingeniería de software permite tener una visión más clara y proporcionan un mecanismo para la evaluación de la calidad del software. Para comprobar cuán bien están definidas las clases, se utilizan las métricas Tamaño Operacional de la Clase y Relaciones entre Clases, las cuales han sido diseñadas para evaluar los siguientes atributos de calidad:

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

Para la validación del diseño se aplicaron las métricas Tamaño operacional de clase (TOC) y Relaciones entre clases (RC) debido al conjunto de atributos de calidad de diseño que ambos miden.

Métrica de Tamaño Operacional de Clases (TOC)

Esta métrica se encuentra entre las planteadas por Lorenz y Kidd y pertenece al grupo de las métricas de tamaño de clase. La misma se centra en la cantidad de operaciones para cada clase individual y los valores promedio para el sistema como un todo. Se encarga de medir la calidad de acuerdo a los atributos Responsabilidad, Complejidad de implementación y Reutilización de las clases (Chidamber and Kemerer's metrics suite: a measurement theory perspective, 1996).

La aplicación de la métrica TOC define los siguientes atributos de calidad:

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

Tabla 4 Criterios de evaluación

Atributos	Categoría	Criterio
Responsabilidad	Baja	\leq Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$>$ Promedio
Complejidad de implementación	Baja	\leq Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$>$ Promedio
Reutilización	Baja	\leq Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$>$ Promedio

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

Resultados obtenidos al aplicar la métrica TOC

Una vez obtenidos los resultados de la evaluación del instrumento de medición de la métrica TOC, se concluyó que la propuesta de diseño presenta una elevada calidad, teniendo en cuenta que el 82% de las clases tienen una baja responsabilidad y complejidad, y a su vez un alto nivel de reutilización entre ellas.

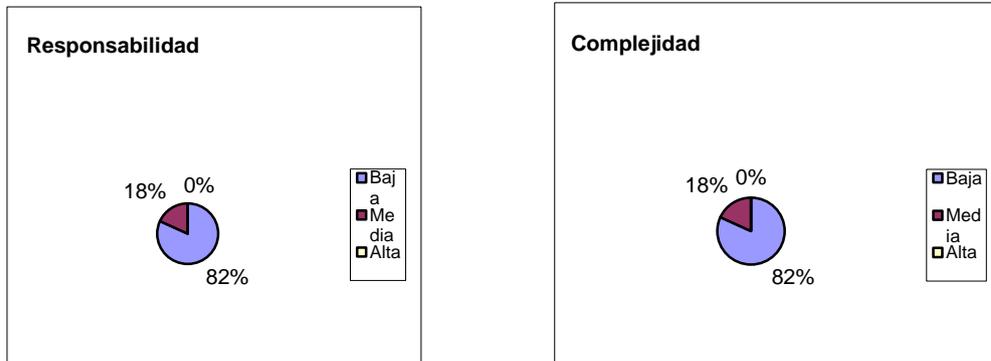


Figura 11 Resultado de la evaluación de la métrica para los atributos responsabilidad y complejidad

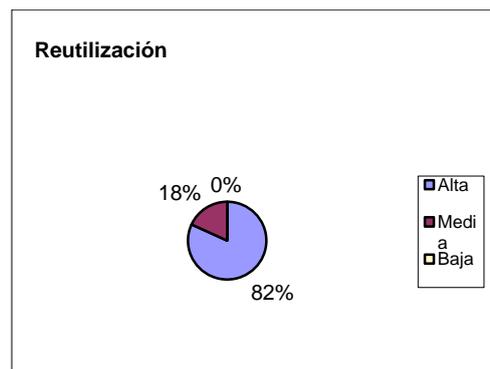


Figura 12 Resultado de la evaluación de la métrica para el atributo reutilización

Métrica de Relaciones entre Clases (RC)

Esta métrica se basa en la cantidad de relaciones de uso que presenta una clase determinada con las demás clases. Utiliza los atributos Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas para medir la calidad del diseño de la clase.

Tabla 5 Atributos de calidad a evaluar por la métrica RC.

Atributo de calidad	Modo en que lo afecta
Acoplamiento	Un aumento del RC implica un aumento del acoplamiento de la clase.
Complejidad de mantenimiento	Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

Reutilización	Un aumento en el 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 necesaria para probar una clase.

Para su aplicación se definen un conjunto de criterios y categorías de evaluación:

Tabla 6 Criterios y categorías a medir durante la evaluación

Atributos	Categoría	Criterio
Acoplamiento	Baja	1
	Medio	2
	Alto	> 2
Complejidad de mantenimiento	Baja	\leq Promedio
	Media	Entre Promedio y $2 \cdot$ Promedio
	Alta	> Promedio
Reutilización	Baja	\leq Promedio
	Media	Entre Promedio y $2 \cdot$ Promedio
	Alta	> Promedio
Cantidad de pruebas	Baja	\leq Promedio
	Media	Entre Promedio y $2 \cdot$ Promedio
	Alta	> Promedio

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

Resultados obtenidos al aplicar la métrica RC

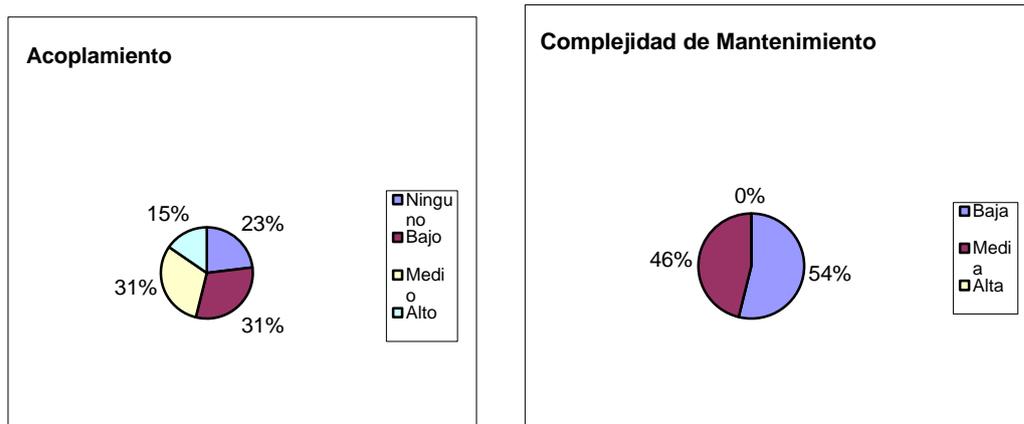


Figura 13 Resultado de la evaluación de la métrica para los atributos acoplamiento y complejidad de mantenimiento

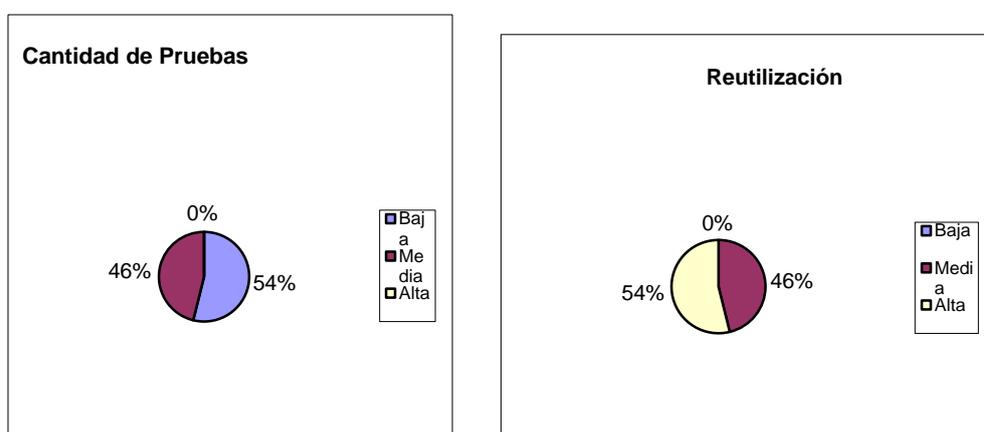


Figura 14 Resultado de la evaluación de la métrica para los atributos cantidad de pruebas y reutilización

Una vez obtenidos los resultados de la evaluación del instrumento de medición de la métrica RC, se concluyó que el diseño propuesto consta de una calidad fiable, apreciándose que el 77% de las clases empleadas en la propuesta presentan 3 o menos dependencias entre otras clases existentes, posibilitando evaluaciones positivas de los atributos de calidad (acoplamiento, complejidad de mantenimiento, cantidad de pruebas y reutilización).

2.7 Modelo de Datos

Un diagrama o modelo entidad-relación (DER) es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información, así como sus interrelaciones y propiedades (Pressman, 2002).

Se realizó el diseño de la base de datos mediante la utilización de la herramienta case Visual Paradigm es su versión 8.0 la misma muestra las tablas que son utilizadas en la

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

solución propuesta para el almacenamiento de la información, las cuales están normalizadas, cumpliendo con las normas establecidas para el diseño de bases de datos. La base de datos se encuentra en tercera forma normal pues sus atributos dependen directamente de la clave primaria, o sea no se relacionan a través de otros atributos, eliminando de esta manera la dependencia transitiva.

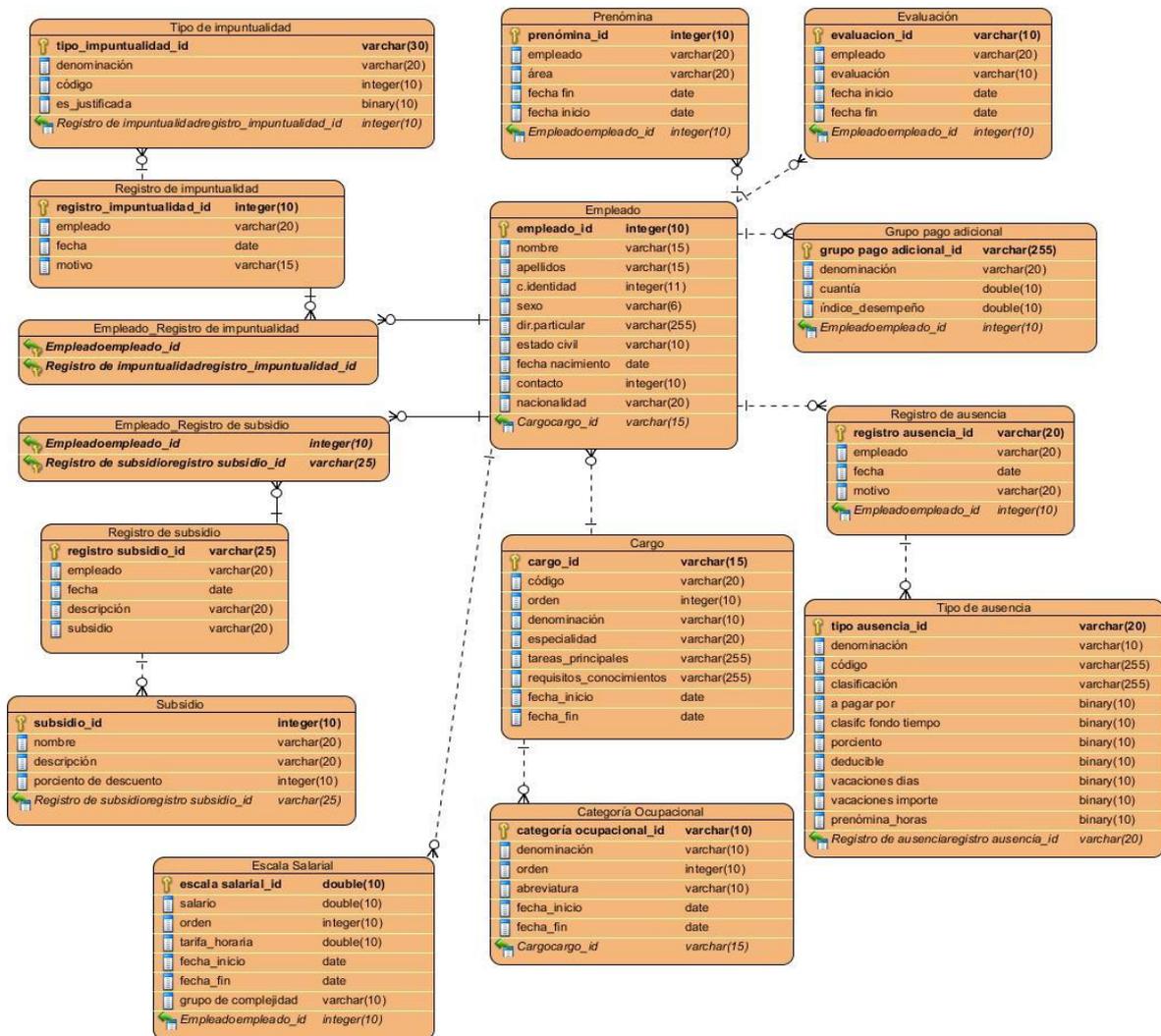


Figura 15 Modelo de datos

Conclusiones Parciales

- ✓ Con la realización de cada disciplina propuesta por la metodología de desarrollo adoptada se logró como resultado la generación de distintos artefactos como las descripciones de los procesos de negocios, las descripciones de requisitos por procesos, la especificación de los requisitos del sistema, el diagrama de clases del diseño, el modelo conceptual y su diccionario de datos, así también como el modelo de datos, que permitieron en gran medida realizar el diseño de la propuesta de solución.
- ✓ La aplicación de las diferentes técnicas de obtención de requisitos arrojó como

ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

resultado la identificación de 70 requisitos funcionales, los cuales deben satisfacer la propuesta de solución.

- ✓ En consecuencia de la utilización de las métricas de validación TOC y RC se validó el diseño propuesto, lográndose un diseño del sistema simple y confiable.

DISEÑO, IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

CAPÍTULO 3: Implementación y Validación de la propuesta de solución

Introducción

En este capítulo se abordan todos los elementos relacionados con la implementación, la validación y la realización de pruebas internas propuestos por la metodología, además, se definen los estándares de codificación a utilizar.

3.1 Implementación

Según (Metodología de desarrollo para la Actividad Productiva de la UCI, 2015) en la implementación se construye el sistema a partir de los resultados del Análisis y Diseño.

3.1.1 Diagrama de Componentes

El diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software, sean estos ficheros de código fuente, binarios o ejecutables. Los elementos de modelado que lo conforman son los componentes y paquetes que muestran la estructura del sistema en términos de implementación a un alto nivel (Pressman, 2010). El diagrama de componentes proporciona una visión física de la implementación del software. Muestra la organización de los componentes y sus relaciones.

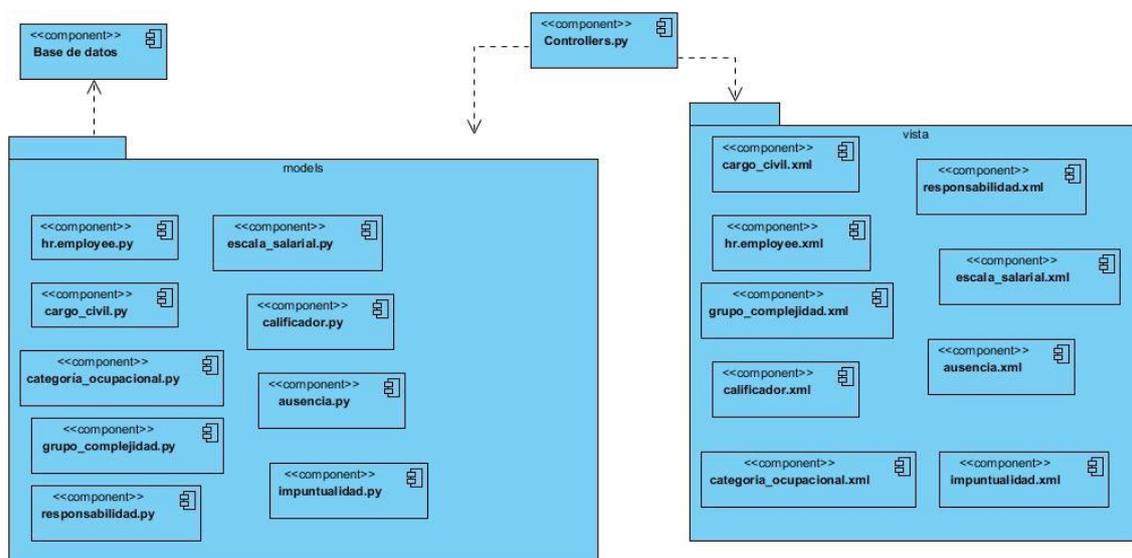


Figura 16 Diagrama de componentes

3.1.2 Estándares de Codificación

Los estándares de código son una forma de “normalizar” la programación de forma tal que, al trabajar en un proyecto, cualquier persona involucrada en el mismo tenga acceso y comprenda el código (Microsoft, 2016)

DISEÑO, IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

Definir los estándares de codificación a utilizar en la investigación permite tener un estilo único en la implementación de la solución, lo cual facilitará el estudio y comprensión del código haciendo que sea más fácil a la hora de realizar cambios o dar soporte.

A continuación, varios ejemplos de estilos de codificación definidos para la implementación de la propuesta de solución, a partir del análisis de los estándares utilizados en la arquitectura de referencia Odoos:

Nombre de archivos

Vista: nom_cargo.xml

 nom_cargo

Clases de un modelo

Declaración

```
class NomCargo(models.Model):  
    _name = 'nomina.nom.cargo'  
    _desc = 'Nomenclador cargo'
```

Vistas

form: `<record model="ir.ui.view" id="cargo_form">`

tree: `<record model="ir.ui.view" id="cargo_tree">`

3.2 Pruebas Internas

En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas.

Pruebas de Software

Las pruebas de software son un elemento crítico para la garantía de la calidad de software y una revisión final de las especificaciones del diseño y de la codificación. El objetivo principal de estas pruebas consiste en medir el grado en que el software satisface los requerimientos definidos (Pressman, 2010).

3.2.1 Pruebas de Caja Blanca

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

DISEÑO, IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

por lo menos una vez todos los caminos independientes de cada módulo (Pressman, 2010).

Técnica del camino básico

Es una técnica de prueba de caja blanca propuesta inicialmente por Tom McCabe. El método del camino básico permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución (Pressman, 2002)

La complejidad ciclomática es una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa. Cuando se usa en el contexto del método de prueba del camino básico, el valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y nos da un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez (Pressman, 2002)

La Figura 17 muestra el código de la funcionalidad def transform_incidencias(self)

```
@api.one
@api.depends('incidencia_ids')
1 def transform_incidencias(self):
    # cargar evaluacion del mes segun las fecha
2 if self.incidencia_ids:
    2 text = ''
3 for i in self.incidencia_ids:
    4 if text:
        5 text = text + ',' + i.insidencia.codigo
    else:
        6 text = i.insidencia.codigo
7 self.incidencia_ids_str = text
8 return self.incidencia_ids_str
```

Figura 17 Método def transform_incidencias(self)

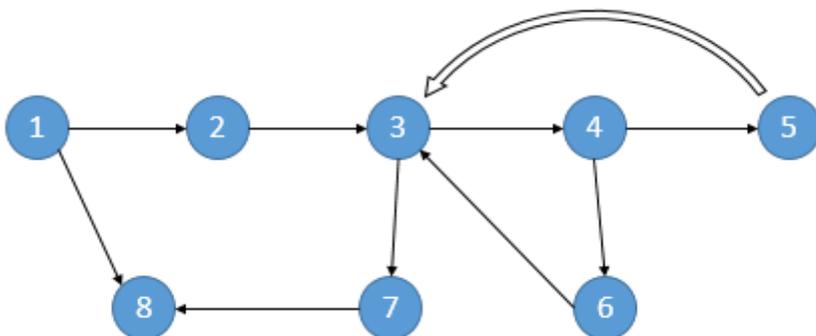


Figura 18 Grafo resultante de aplicar técnica Camino Básico.

DISEÑO, IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

$$\begin{aligned}
 V(G) &= A - N + 2 \\
 &= 10 - 8 + 2 \\
 &= 4
 \end{aligned}$$

$V(G)$ = Complejidad ciclomática
 A = Cantidad de aristas del grafo
 N = Cantidad de nodos del grafo

$$\begin{aligned}
 V(G) &= P + 1 \\
 &= 3 + 1 \\
 &= 4
 \end{aligned}$$

P = Nodos predicados.

$$\begin{aligned}
 V(G) &= \text{Regiones} \\
 &= 4
 \end{aligned}$$

Regiones = Áreas delimitadas por nodos y aristas del grafo

Cantidad de caminos básicos = 4

Camino 1: 1, 8

Camino 2: 1, 2, 3, 7, 8

Camino 3: 1, 2, 3, 4, 5, 3, 7, 8

Camino 3: 1, 2, 3, 4, 6, 3, 7, 8

Tabla 7 Caso de prueba para el camino 1

Descripción	Verificar los resultados de evaluar la función def transform_incidencias(self)
Condición de ejecución	If not self.incidencias(self)
Entrada	incidencias_id
Resultado	Devuelve las incidencias agrupadas

Tabla 8 Caso de prueba para el camino 2

Descripción	Verificar los resultados de evaluar la función def transform_incidencias(self)
Condición de ejecución	If self.incidencias(self)
Entrada	incidencias_id
Resultado	Devuelve las incidencias agrupadas

Tabla 9 Caso de prueba para el camino 3

Descripción	Verificar los resultados de evaluar la función def transform_incidencias(self)
-------------	--

DISEÑO, IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

Condición de ejecución	If self.incidentes(self)
Entrada	incidentes_id
Resultado	Devuelve las incidencias agrupadas

Tabla 10 Caso de prueba para el camino 4

Descripción	Verificar los resultados de evaluar la función def transform_incidentes(self)
Condición de ejecución	If self.incidentes(self)
Entrada	incidentes_id
Resultado	Devuelve las incidencias agrupadas

Luego de haberle aplicado el método del camino básico a la funcionalidad anterior se comprobó que cada sentencia es ejecutada al menos una vez. El valor calculado como complejidad ciclomática definió el número de caminos independientes del conjunto básico, lo que facilitó el límite superior para el número de pruebas que se deben realizar.

3.2.2 Prueba de Caja Negra

Las pruebas de caja negra, también denominada prueba de comportamiento, se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa (Pressman, 2002).

Método de Partición de Equivalencias

Dentro de las pruebas de caja negra se utiliza el método de partición de equivalencia es un método que divide el dominio de entrada de un programa en clases de datos de los que pueden derivarse casos de prueba. Se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada (Pressman, 2002).

Para realizar las pruebas de caja negra se utilizó la técnica **Partición Equivalente**. Definiendo para cada uno de los requisitos identificados un caso de prueba. A continuación, se muestra el flujo central del caso de prueba de la agrupación de requisitos Adicionar cargo:

DISEÑO, IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

Tabla 11 Diseño caso prueba de caja negra

Escenario	Descripción	Nombre	Puesto	Responsabilidad	Grupo	Categoría Ocupacional	Código	Abreviatura	Resolución	Requisitos de Conocimientos
EC 1.1: Adicionar cargo correctamente	Se adiciona correctamente un cargo.	V	V	V	V	V	V	V	V	V
		Chofer D	Chofer D	Jefe de brigada	IV	Operario	1	CD	240	Habilidades de manejo
EC 1.3: Adicionar cargo con datos vacíos	Se muestra un mensaje indicando que no se han introducido los datos obligatorios.	I	I	V	V	I	V	V	V	I
		Vacío	Vacío	Vacío	Vacío	Vacío	Vacío	Vacío	Vacío	Vacío
EC 1.4: Adicionar cargo datos incorrectos	Se muestra un mensaje de error indicando que se han introducido datos incorrectos.	I	I	I	I	I	I	I	I	I
		@#%&/*-	@#%&/*-	{/@#~½	{/@#~½	{/@#~½	{/@#~½	{/@#~½	{/@#~½	{/@#~½

DISEÑO, IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

EC 1.5: Cancelar.	Cancela la acción de Adicionar cargo retorna a la página que le dio origen.	NA								
-------------------	---	----	----	----	----	----	----	----	----	----

En el expediente de proyecto entregable, se encuentra el documento CEIGE-ODOO Diseño de caso de pruebas, donde se evidencian el resto de los casos de pruebas de caja negra aplicados en la solución.

DISEÑO, IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

Resultados de la prueba

Con la aplicación de las pruebas de caja negra se demostró el correcto funcionamiento de las funcionalidades implementadas en el módulo Sistema de Pago, pues se obtuvieron los resultados esperados como respuesta al conjunto de condiciones de entrada definidas en cada diseño de caso de prueba.

3.2.3 Pruebas Internas

Se ejecutaron las pruebas internas, definidas como disciplina en la metodología AUP-UCI que guía la investigación, estas pruebas fueron realizadas por el grupo de calidad del Centro CEIGE, a partir de los diseños de casos de prueba de los requisitos funcionales del sistema. El método de prueba aplicado demostró resultados satisfactorios desde el punto de vista funcional. Las no conformidades detectadas fueron analizadas y corregidas debidamente en el tiempo establecido, logrando un correcto comportamiento ante diferentes situaciones (entradas válidas y no válidas). A continuación, se muestra una tabla con el número de no conformidades detectadas en cada iteración de prueba realizada.

Tabla 12 No Conformidades detectadas

Tipo de NC	1era Iteración	2da Iteración
Ortografía	4	-
Funcional	4	-
Validación	5	1
Otras	2	-
No Procede	-	-

3.2.4 Pruebas de Aceptación

Las pruebas de aceptación se realizan para que el cliente certifique que el sistema es válido para él. La planificación detallada de estas pruebas debe haberse realizado en etapas tempranas del desarrollo del proyecto, con el objetivo de utilizar los resultados como indicador de su validez: si se ejecutan las pruebas documentadas a satisfacción del cliente, el producto se considera correcto, y por tanto, adecuado para su puesta en producción (Isabel Ramos, 2007).

Las pruebas de aceptación se realizan para que el cliente certifique que el sistema es válido para él. La planificación detallada de estas pruebas debe haberse realizado en etapas tempranas del desarrollo del proyecto, con el objetivo de utilizar los resultados como indicador de su validez: si se ejecutan las pruebas documentadas a satisfacción del

DISEÑO, IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

cliente, el producto se considera correcto, y por tanto, adecuado para su puesta en producción (Isabel Ramos, 2007).

El cliente, luego de haber revisado los productos de trabajo entregados, determina que está de acuerdo con el producto final mostrado, el cual fue desarrollado bajo los requisitos previamente definidos. Una vez comprobada la implementación de los requisitos, el cliente emitió el acta de aceptación con el cliente, la cual puede ser consultada en los anexos.

3.3 Resultados de la validación de la investigación

Como resultado del análisis de la población objeto de aplicación de la encuesta, se decidió aplicar un muestreo a conveniencia. Consiste en una técnica de muestreo no probabilístico donde los sujetos son seleccionados, dada la conveniente accesibilidad y proximidad de los sujetos para el investigador.

En correspondencia con estos elementos se diseñó un cuestionario orientado a obtener información sobre la valoración de los expertos, para ello se propone un cuestionario de un total de 5 preguntas de respuesta cerrada, facilitándose de esta forma el procesamiento y la cuantificación. La vía utilizada para la aplicación fue el cuestionario impreso.

Las respuestas a las preguntas están representadas en una escala de Likert. Es una escala psicométrica comúnmente utilizada en cuestionarios y se considera la escala de uso más amplio en encuestas para la investigación, principalmente en ciencias sociales.

Para el procesamiento de la información obtenida en las encuestas se hizo corresponder un valor numérico a cada posible resultado de la escala utilizada. Estos valores utilizados se muestran en la siguiente tabla:

Tabla 13 Valores asociados a cada respuesta

Respuesta	Valor
Mucho	5
Bastante	4
Poco	3
Muy poco	2
Nada	1

DISEÑO, IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

Procedimiento para determinar el grado de consenso entre expertos:

A partir de la tabla 14 se puede afirmar que existe un alto grado de concordancia en el criterio de los expertos en todas las preguntas realizadas. En todos los casos el coeficiente de variación muestra valores por debajo de 0,22 lo que sustenta la afirmación anterior. De este modo se puede concluir que, de acuerdo al consenso en el juicio de expertos, la solución propuesta contribuye a mejorar el proceso de gestión del sistema de pago de la UCI. Destacando las preguntas 2 y 3 con un coeficiente de variación por debajo de 0,15.

Tabla 14 Resultados de la aplicación

Expertos	Preguntas				
	1	2	3	4	5
1	4	4	4	3	5
2	5	5	5	5	3
3	3	4	5	5	4
4	4	5	5	5	3
5	5	5	4	5	5
6	5	4	4	5	4
7	4	5	4	4	5
8	4	5	5	4	3
9	4	4	4	4	4
10	5	4	4	5	3
Cj media	4,30	4,5	4,4	4,5	3,9
Varianza	0,46	0,28	0,27	0,50	0,77
Coef. De Variación	0,16	0,12	0,12	0,16	0,22

El análisis de la tabla anterior permitió concluir lo siguiente por cada uno de las preguntas:

Tabla 15 Evaluación de las preguntas aplicadas

Preguntas	Valoración de expertos
1. La solución propuesta contribuirá a mejorar el proceso de gestión del sistema de pago de la UCI	4,3
2. La solución propuesta contribuirá a garantizar la integridad de los procesos del sistema de pago en la UCI.	4,5
3. La solución propuesta contribuye a la centralización de los datos del sistema de pago en la UCI.	4,4

DISEÑO, IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

4. La solución propuesta contribuye a mejorar la consulta de la información histórica del proceso de pago.	4,5
5. La solución propuesta contribuye a mejorar la creación de reportes en el proceso de pago.	3,9

Conclusiones Parciales

- ✓ Se obtuvo como resultado la implementación del módulo de gestión de Sistema de Pago de la Universidad de Ciencias Informáticas.
- ✓ Se realizaron las pruebas de caja blanca y de caja negra, arrojando resultados satisfactorios con respecto a el cumplimiento de los requisitos funcionales y a la implementación de la propuesta de solución.

Conclusiones Generales

- ✓ Durante el estudio de la presente investigación se realizó un análisis relacionado con los sistemas de planificación de recursos empresariales que de alguna manera remuneran al trabajador, arrojando como resultado la elaboración del marco teórico conceptual lo que permitió definir una propuesta de solución.
- ✓ La generación de los distintos artefactos de negocio, análisis y diseño propuestos por la metodología posibilitaron en gran medida un correcto entendimiento y realización del diseño de la propuesta de solución.
- ✓ Con la utilización de estándares de codificación en la implementación del sistema se obtuvo un código estructurado y normalizado, logrando un lenguaje común y comprensible, lo cual facilitará el estudio del código haciendo que sea más fácil a la hora de realizar cambios o dar soporte.
- ✓ Como resultado de la disciplina de Implementación se logró construir el módulo funcional para la gestión del sistema de pago de la UCI.
- ✓ La validación de la investigación mediante el uso del método de criterio de experto arrojó que la propuesta de solución contribuye a mejorar el proceso de gestión del sistema de pago de la UCI.

RECOMENDACIONES

Recomendaciones

Los objetivos generales de este trabajo fueron alcanzados, pero durante su desarrollo, han surgido ideas que sería recomendable tener en cuenta para su futuro perfeccionamiento:

- Continuar el desarrollo de este módulo, adicionándole nuevas funcionalidades y servicios que puedan satisfacer necesidades futuras de los clientes.
- Consumir los servicios que te permitan obtener los datos de los trabajadores.
- Presentar los resultados de la investigación en eventos científicos.

Bibliografía

- Unit4 Business World. 2018.** Software ERP Unit4 Business World. [Online] mayo 22, 2018. [Cited: abril 23, 2018.] [//www.unit4.com/es/productos/software-erp-gestion-recursos-humanos/human-resources-and-payroll..](http://www.unit4.com/es/productos/software-erp-gestion-recursos-humanos/human-resources-and-payroll..)
- ADAM. 2018.** ADAM - Nómina y Recursos Humanos. [Online] 2018. [Cited: marzo 4, 2018.] <http://www.b1consulting.net/ADAM%20Payroll.htm>.
- Agile software development methods: Review and analysis.* **Pekka Abrahamsson, Outi Salo, Jussi Ronkainen. 2017.** 2017.
- Asamblea Nacional del Poder Popular. 2013.** *Código del Trabajo.* 2013.
- . **2008.** *Ley No. 105/08.* La Habana : s.n., 2008.
- Carlos Tinchet Varela, Angel Luis Selva. 2014.** 9, 2014, Vol. 1.
- CEIGE. 2015.** *Informe del estudio realizado a Cedrux.* Centro de Informatización de Entidades. Universidad de las Ciencias Informáticas(UCI) : s.n., 2015.
- Chidamber and Kemerer's metrics suite: a measurement theory perspective.* **Martin Hitz, Behzad Montazeri. 1996.** 4, 1996, Vol. 22.
- Extreme Programming and Scrum.* **Faiza Anwer, Syed Shah Muhammad Shah. 2017.** 2, 2017, Vol. 8.
- García Valdés, M and Suárez Marín, M. 2013.** *El método Delphi para la consulta a expertos en la investigación científica.* 2013.
- International Standard ISO/IEC 14764 IEEE. 2006.* Universidad de las Ciencias Informáticas : Second Edition(2006), 2006, Software Engineering-Software Life Cycle Processes-Maintenance. Std 14764-2006.
- Iruela, Juan. 2018.** Canal Informática y TICS. [Online] 2018. [Cited: junio 6, 2018.] <https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/>.
- Isabel Ramos. 2007.** Técnicas cuantitativas para la gestión en la Ingeniería del Software. 2007.
- JasperSoft. 2014.** JasperReports. [Online] 6.0.0, febrero 10, 2014. [Cited: mayo 2, 2018.] <http://community.jaspersoft.com/project/jasperreports-library>.
- Larman, Craig. 2003.** *UML y patrones, Introducción al análisis y diseño orientado a objetos.* 2003.
- Metodología de desarrollo para la Actividad Productiva de la UCI. 2015.** *Metodología de desarrollo para la Actividad Productiva de la UCI.* 2015.
- Microsoft. 2016.** Revisiones de código y estándares de codificación. [Online] 2016. [Cited: mayo 1, 2018.] <https://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.
- Ministerio de Trabajo y Seguridad Social. 2009.** *Decreto 283 Reglamento Ley Seguridad Social.* La Habana : s.n., 2009.

- . **2009.** *RESOLUCIÓN No. 116/2009.* La Habana : s.n., 2009.
- ODOO. 2016.** [Online] enero 20, 2016. [Cited: abril 23, 2018.] <http://www.odoo.com>.
- Paradigm, Visual. 2012.** What is Visual Paradigm? [Online] mayo 7, 2012. [Cited: abril 23, 2018.] <http://www.visual-paradigm.com>.
- Patrones de diseño.* **Jean Carlo Murillo Alpízar, Carolina Oconitrillo Rodriguez, Leylin Esquivel Bolaños. 2017.** 2017.
- PgAdmin. 2016.** PgAdmin. [Online] 2016. [Cited: mayo 30, 2018.] www.pgadmin.org.
- PostgreSQL. 2016.** PostgreSQL. [Online] 2016. [Cited: mayo 30, 2018.] www.postgresql.org.
- Pressman, Roger. 2002.** *Ingeniería del software. Un enfoque práctico.* 2002. Quinta Edición.
- . **2002.** *Ingeniería del Software. Un enfoque práctico.* s.l. : Quinta Edición, 2002.
- Pressman, Roger S. 2010.** *Ingeniería del Software. Un Enfoque Práctico.* 2010.
- Project Management Assurance in Agile Projects: Research in Progress.* **Dawn Owens, Jeffrey W. Merhout. 2018.** 2018.
- PyCharm. 2018.** PyCharm.: *PyCharm: Python IDE for Professional Developers by JetBrains.* [Online] 2018. [Cited: junio 6, 2018.] <https://www.jetbrains.com/pycharm/>.
- Python. 2018.** Welcome to Python.org. [Online] 2018. [Cited: junio 6, 2018.] <https://www.python.org/>.
- Roger S. Pressman, Jose Maria Troya. 1988.** *Ingeniería del software.* 1988.
- Salaries and Piece Rates.* **Lazear, Edward P. 2014.** 2014.
- Sommerville, Ian. 2002.** *Ingeniería del Software.* Universidad de las Ciencias Informáticas : s.n., 2002.
- Universidad de las Ciencias Informáticas. 2017.** *Catálogo de Claves de Ausencias Ordenado por Código.* La Habana : s.n., 2017.
- Win2PDF. 2015.** *Utilización del método Delphi en la pronosticación: una experiencia inicial.* 2015.
- XML. 2016.** XML. [Online] febrero 6, 2016. [Cited: abril 23, 2018.] <http://www.xml.com>.