

Universidad de las Ciencias Informáticas “Facultad 4”



Título: “Análisis y diseño del componente Nómina del subsistema Capital Humano del sistema CedruX”

Trabajo de Diploma para optar por el título de
Ingeniero en ciencias Informáticas

Autor(es): Daniellis Palau Pelegrín
Fidel Jiménez Sanzano

Tutor(es): Ing. Susana Bermúdez Rodríguez

Co-tutor: Ing. Lisett Pérez Quintero

Ciudad de la Habana, junio 2009.

DATOS DE CONTACTO

- Graduada en el 2008 de la carrera Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas.
- Ha trabajado en la Unidad de Compatibilización y Desarrollo de Software para la Defensa (UCID).
- Actualmente se desempeña como analista del proyecto ERP en una de sus líneas de desarrollo.

Correo electrónico: sbermudez@uci.cu

AGRADECIMIENTOS

Quiero agradecer en primer lugar a mi mamá, por su preocupación, su espíritu de sacrificio y entrega, por darme siempre lo mejor y sobre todo por quererme tanto.

Al resto de mi familia, a mi tía Maye por su cariño y cuidado, mis abuelos, mis hermanos Raúl por ayudarme tanto durante toda mi vida y a Damián por cuidar siempre de mi.

A mis amigas Ariagna, Elizabeth, Yanara por estar siempre ahí cuando mas lo necesitaba. A los nuevos amigos Betty, Misle, Oda, Arnol y Yankiel por su comprensión y ayuda.

A mi tutora Susana, le agradezco mucho su infinita paciencia y por enseñarme que siempre se pueden hacer mejor las cosas.

Y a todos aquellos que han compartido conmigo todos estos años.

Muchas gracias

Nelly

Quisiera agradecerle a nuestro comandante por brindarme la oportunidad de formarme como ingeniero al estudiar en esta universidad que es un sueño hecho realidad.

A mi familia, a mi papá por enseñarme a soñar con lo imposible y a mi mamá por enseñarme a luchar por realizarlo.

A mi hermano que para que se sienta orgulloso de mí me empuja a ser mejor cada día.

A mis abuelos por mostrarme que el mejor libro de historia es el cuento de un abuelo.

A mi tutora Susana por todo lo que aprendí gracias a ella durante la tesis.

A mi amigo y compañero de trabajo Rainer, aquellos días fueron duros en Metadatos pero valieron la pena.

A mis compañeros de proyecto que gracias a ellos pudimos entregar una primera versión.

En general quisiera agradecer a todos lo que de una forma u otra han contribuido a mi formación como Ingeniero en Ciencias Informáticas

Fidel

DEDICATORIA

A la memoria de mi abuelo Ruben a quien después de tantos años de ausencia sigo recordando.

A mi mamá.

A Lala mi segunda madre.

Nelly

Al comandante impulsor de esta idea.

A mi familia que siempre a estado dándome ánimo por seguir adelante.

A mis tíos y tías por la ayuda brindada.

A mis amigos de la universidad de todos estos años en especial a Jose Armando, Lugo, Toste, Jose Ramón, Félix y otros.

A mis amigas de estos años Magdanis, Daylén, Sucel, y a las de los grupos por los cuales pasé.

A mis compañeros de cuarto en especial a Lara, John

Fidel

RESUMEN

En el área de Capital Humano de las entidades cubanas se realizan varios procesos entre ellos la Confección y cálculo de la nómina, que permite realizar el pago de los trabajadores.

En la actualidad se ha comprobado que el uso de las tecnologías en los procesos de negocios provoca un aumento de la productividad haciendo más manejables la información y los cálculos relacionados con ésta. La informatización del proceso Confección y cálculo de la nómina forma parte de tales avances con el objetivo de lograr una gestión eficiente del mismo.

Con el propósito de contribuir al procesamiento de la nómina en las entidades, el presente trabajo realiza un análisis de los principales requisitos de manera tal que sea adaptable a todas las formas de pago existentes. También se presenta el diseño del sistema que satisface los requisitos antes acordados y otros agregados con el objetivo de que la solución dada sea lo más configurable y estable posible.

En este trabajo con la aplicación de las técnicas de captura de requisitos se modelan los subprocesos que describen el cálculo de la nómina obteniéndose los diagramas correspondientes al negocio, también se describen los requisitos validados con los clientes, los diagramas de clases y de componentes para el modelado del diseño de la solución propuesta. La realización de un sistema que cumpla con lo antes planteado y además integrado a la solución de ERP para nuestro país se hace una necesidad para mejorar la dinámica de este proceso en las entidades.

PALABRAS CLAVES

Capital humano, nómina, procesamiento nómina, concepto de pago, patrones, requisitos

TABLA DE CONTENIDOS

AGRADECIMIENTOS	1
DEDICATORIA	2
RESUMEN	3
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 Introducción	5
1.2 Marco Teórico.....	5
1.2.1 Capital Humano	5
1.2.2 Estimulación Moral y Material.....	5
1.2.3 Concepto de pago.....	6
1.2.4 Impuesto	6
1.2.5 Nómina.....	6
1.2.6 Salario.....	6
1.2.7 Submayores.....	7
1.2.8 Retenciones	7
1.3 Proceso de Confección y Cálculo de la Nómina.....	7
1.3.1 Mapa de relaciones internas	8
1.3.2 Mapa de relaciones externas	11
1.4 Soluciones existentes	13
1.4.1 Sistemas similares en el área internacional.....	13
1.4.2 Sistemas similares en el área nacional.....	20
1.5 Modelo de desarrollo orientado a componentes.....	23
1.6 Lenguaje de Modelado	24
1.6.1 Notación para el Modelado de Procesos del Negocio	24
1.6.2 Lenguaje de Modelado Unificado	25
1.6.3 Comparación entre UML y BPMN	26
1.7 Herramientas de Modelado.....	27
1.7.1 Visual Paradigm for UML 6.1 Enterprise Edition.....	27
1.8 Propuesta de lenguajes de programación a utilizar para el desarrollo del sistema.....	28
1.8.1 PHP 5.....	29
1.9 Sistemas gestores de bases de datos	29
1.9.1 PostgreSQL 8.3.....	30
1.10 Servidor Web	31
1.10.1 Servidor Web Apache 2.0.....	31
1.11 Zend Estudio para Eclipse.....	32
1.12 Conclusiones parciales del capítulo.....	32
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	33

2.1	Introducción	33
2.2	Técnicas y métodos empleados en la captura de requisitos	33
2.2.1	Entrevistas	33
2.2.1	Prototipos.....	34
2.2.2	Glosario.....	34
2.2.3	Sistemas Existentes	34
2.2.4	Arqueología de documentos.....	35
2.3	Patrones empleados.....	35
2.3.1	Patrones CRUD	35
2.4	Modelo conceptual.....	36
2.5	Requisitos funcionales	39
2.6	Especificaciones de requisitos	43
2.6.1	Requisito funcional Gestionar impuesto y contribuciones	43
2.6.2	Requisito funcional Gestionar concepto de pago.....	47
2.6.3	Requisito funcional Gestionar tipo de nómina.....	50
2.6.4	Requisito funcional Gestionar período de pago	54
2.6.5	Requisito funcional Gestionar tipos de ajustes	58
2.6.6	Requisito funcional Administrar nómina.....	60
2.6.7	Requisito funcional Revisar nómina.....	69
2.6.8	Requisito funcional Comprobante de operaciones.....	76
2.6.9	Requisito funcional Cierre contable	78
2.6.10	Requisito funcional Configuración.....	78
2.6.11	Requisito funcional Gestionar frecuencia de pago	79
2.6.12	Requisito funcional Reportes	82
2.7	Conclusiones parciales	85
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....		86
3.1	Introducción	86
3.2	Patrón de arquitectura	86
3.2.1	Modelo vista controlador (MVC)	86
3.3	Patrones de diseño.....	87
3.3.1	Patrones Grasp (Patrones generales de software para asignar responsabilidades)	87
3.3.2	Patrones Gof (Gang of Four).....	89
3.3.2.1	Patrones estructurales	89
3.3.2.2	Patrones de comportamiento.....	90
3.4	Diagramas de clases del diseño	92
3.5	Diagrama de componente.....	105
3.6	Conclusiones parciales	107
CAPÍTULO 4: VALIDACION DE LOS RESULTADOS.....		108
4.1	Introducción	108
4.2	Criterios para la evaluación y aceptación de los requisitos	108
4.3	Técnicas de validación de requisitos.....	110
4.3.1	Técnica de Prototipado	110

4.3.2	Técnica de Diseño de Casos de Prueba	111
4.3.2.1	Diseño de caso de prueba del requisito Crear nómina.....	111
4.3.2.2	Diseño de caso de prueba del requisito Agregar trabajadores a la nómina.....	116
4.3.2.3	Diseño de caso de prueba del requisito Procesar nómina	118
4.4	Métricas orientadas a clases para evaluar el diseño	119
4.4.1	Tamaño de clase (TC).....	119
4.4.2	Árbol de profundidad de herencia (APH)	122
4.4.3	Relaciones entre Clases (RC)	123
4.5	Conclusiones parciales	127
CONCLUSIONES		128
RECOMENDACIONES		130
BIBLIOGRAFÍA.....		131
ANEXOS.....		134
GLOSARIO		269

INTRODUCCIÓN

El mundo actual evoluciona a una velocidad vertiginosa y el uso de las tecnologías de la información y las comunicaciones (TIC) se ha vuelto indispensable para su permanente desarrollo. Las TIC se han desarrollado con el fin de mejorar la calidad de vida de las personas en los diferentes entornos y están presentes en la educación, la salud, la cultura, la economía y en el resto de las esferas que constituyen la sociedad.

Como parte de las tecnologías de la información y las comunicaciones *la informatización de los procesos* ha sido un motor propulsor para potenciar el cumplimiento de las funciones a todos los niveles, con un máximo de racionalidad y control en las diferentes áreas de la misma. En las empresas es esencialmente importante, pues organizan las actividades guiando a las entidades hacia el aumento de la productividad con un elevado nivel de eficiencia.

La conveniencia de la utilización de las TIC está siendo aprovechada por el gobierno cubano, con el fin de crear una sociedad con un alto nivel cultural, permitiendo elevar la economía del país con la implantación de sistemas informáticos en las diferentes entidades existentes, muchos de los cuales han sido importados. Una de las principales metas es desarrollar la Industria del Software y convertirla en uno de los principales renglones económicos del país, no solamente por los beneficios en el desarrollo de sistemas para uso interno, sino también con el fin de insertarnos en el mercado de software a nivel mundial por su perspectiva económica.

En estos últimos tiempos la dirección del país ha ido comprendiendo la necesidad de desarrollar sistemas informáticos propios y para ello se está llevando a cabo una estrategia para lograr mayor eficiencia económica, ahorro de recursos y sustitución de importaciones de sistemas con el objetivo de lograr el fortalecimiento de las entidades. Como parte de las iniciativas que se están llevando a cabo se tomó la decisión del desarrollo de una solución informática para la gestión integral de entidades que proveerá de una herramienta para el control y gestión de recursos financieros, materiales y de capital humano. Un sistema que proporcionará mayor integración de la información y modernización de los procesos de negocio que le permitirá a las entidades empresariales y presupuestadas mayor eficiencia y productividad.

INTRODUCCIÓN

Este sistema informático esta siendo desarrollado por estudiantes y profesionales de la Universidad de las Ciencias Informáticas (UCI), apoyados por los diferentes ministerios del país. En la UCI se tiene ya la experiencia de desarrollar sistemas de grandes dimensiones aunque no con la misma envergadura de un sistema de estas características que representa un reto para el equipo de desarrollo implicado en su creación por la complejidad de su negocio y la interrelación que existe entre sus procesos. Este dará la posibilidad de poder mantener un mejor control sobre las diferentes entidades que existen en el país, posibilitando que cumpla con todas las funcionalidades requeridas

Uno de los subsistemas más abarcadores que forman parte del sistema CEDRUX es el de Capital Humano. Desde hace ya varios años el factor humano en las organizaciones ha ido adquiriendo una importancia vital en las empresas e instituciones de todo tipo, ya que independientemente de la tecnología que pueda disponerse, el hombre decide en el éxito o no del cumplimiento de su misión por lo que una gestión eficiente de los recursos humanos es fundamental. La gestión del Capital Humano abarca un grupo de procesos que interactúan entre si como son: la organización del trabajo, la integración y selección, la evolución del desempeño, la capacitación y desarrollo, la seguridad y salud del trabajo, la estimulación moral y material, el autocontrol y la comunicación institucional.

Actualmente en las entidades de nuestro país no existe un sistema que interrelacione la gestión de incidencias que se lleva en el departamento de Recursos Humanos y el procesamiento de la nómina de los trabajadores que se realiza en Contabilidad. Esto trae como consecuencia que al no estar integrados la información se tiene duplicada, crece el margen de contaminación de la información sobre todo por errores de captura y se crea un escenario favorable para pagos indebidos. Esto implica también que la información que se obtiene no es oportuna lo que imposibilita un control eficiente de la misma. Dificultando el cumplimiento de las fechas establecidas en el convenio colectivo de trabajo para el pago del personal. De esta forma al proceso de negocio no se le puede dar un adecuado seguimiento y se dificulta el trabajo a las personas encargadas de la toma de decisiones afectando la elaboración de estrategias e iniciativas organizacionales.

Para ello se propone realizar el análisis y diseño de componente Nómina que facilite el cálculo para el pago de los trabajadores, que sea capaz gestionar de manera integrada la información recibida del área

INTRODUCCIÓN

de Recursos Humanos, que permita un acceso rápido y eficiente a la información que se procesa en la misma. Además de permitir realizar el procesamiento de la nómina de manera eficiente, facilitando controlar el pago de los trabajadores posibilitando así un mayor trabajo de análisis para el enfrentamiento a ilícitos económicos. Se pretende poner en práctica una correcta estrategia de captura de requisitos, a través de frecuentes intercambios con los clientes para el entendimiento de los procesos del negocio, constituyendo la base para realizar el diseño del sistema garantizando la eliminación de posibles errores a la hora de implementar el modelado propuesto.

Basándose en lo explicado, se puede deducir, que el **problema** a resolver en este trabajo es: ¿Como contribuir al procesamiento de la nómina, para gestionar de manera eficiente los procesos que en él se llevan a cabo?

Con el objetivo de lograr una alternativa de solución, el presente trabajo propone minimizar el problema existente, partiendo de la modelación de un sistema informático que permita automatizar los procesos del sistema de nómina del subsistema Capital Humano, razón por la cual el **objeto de estudio** queda enmarcado en los procesos de Capital Humano, y delimitando el **campo de acción** el procesamiento de la nómina en las entidades cubanas.

El **objetivo general** trazado para darle solución al problema establecido es: Realizar el análisis y diseño del componente Nómina del subsistema Capital Humano que permita desarrollar de manera más eficiente el cálculo de la nómina de los trabajadores.

Con el propósito de darle cumplimiento al mismo se derivan los siguientes **objetivos específicos**:

- Analizar como se realiza en la actualidad el cálculo de la nómina en las entidades cubanas.
- Realizar el modelo de procesos de negocio y el modelo conceptual.
- Especificar los requisitos de software.
- Validar la propuesta mediante métricas y técnicas.
- Realizar el diseño del sistema de nómina.

INTRODUCCIÓN

Idea a defender

Realizando el análisis y diseño de un sistema que sea capaz de realizar el cálculo de la nómina interrelacionando la gestión de incidencias con el procesamiento de la misma se reduciría la posibilidad de errores, garantizando un mayor control en las operaciones de pago realizadas a los trabajadores.

Como finalidad para dar confirmación a la idea a defender se proponen realizar las siguientes **tareas de investigación:**

- Conceptualizar la información que se maneja en el subsistema de Capital Humano relacionado con el módulo de Nómina.
- Estudiar la documentación de los procesos de Nómina y desarrollar un análisis detallado de los mismos.
- Revisar y seleccionar la bibliografía acerca del tema de Ingeniería de Software para la comprensión y desarrollo exitoso de la investigación.
- Estudiar aplicaciones o soluciones similares.
- Identificar y definir los procesos de negocio.
- Realizar el modelo conceptual.
- Realizar el levantamiento de requerimientos del software.
- Validar los requisitos de software mediante métricas y técnicas.
- Realizar los diagramas de clases de la etapa de diseño.
- Realizar el diagrama de componente de la etapa de diseño.

Este trabajo está estructurado en cuatro capítulos que contienen toda la información referente a la investigación realizada y al análisis y diseño del sistema a desarrollar, la información está distribuida de la siguiente manera:

Capítulo 1: Se enuncian los principales conceptos relacionados con el negocio a modelar; además se identifican y modelan los procesos del negocio. Se realiza un estudio de algunas soluciones de software existentes relacionadas con los sistemas contables y de recursos humanos tanto a nivel nacional como

INTRODUCCIÓN

internacional y por último se hace referencia a la metodología de desarrollo de software y herramienta seleccionada para realizar este trabajo.

Capítulo 2: Se realiza un estudio preliminar del sistema. Se identifican los conceptos del negocio y se realiza el modelo conceptual. Se especifican las técnicas empleadas en la captura de requisitos. Además de la identificación y descripción de los requisitos funcionales que debe de presentar la solución a construir.

Capítulo 3: Se plantean los detalles relacionados con el diseño del sistema que se propone, utilizando para su modelado los diagramas de clases con la aplicación de los patrones correspondientes y del diagrama de componentes; realizando posteriormente una descripción de los mismos.

Capítulo 4: Se realiza la validación de los requisitos funcionales mediante métricas y técnicas definidas que permitirán validar que los requisitos estén descritos correctamente. Se validara la modelación del diseño con la utilización de métricas que permitirán conocer si el diseño realizado es sencillo de implementar.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se tratarán los principales conceptos relacionados con el negocio a modelar tales como nómina, concepto de pago, procesamiento de la nómina, submayores, impuestos y retenciones; siendo estos de vital importancia para su comprensión. Se hace referencia a las principales características de algunos sistemas similares que existen en la actualidad en el área nacional así como internacional. Se brinda información acerca del proceso Confección y cálculo de nómina y por último se referencian las principales metodologías, lenguajes de modelado, tecnologías y herramientas propuestas para el desarrollo de esta aplicación.

1.2 Marco Teórico

1.2.1 Capital Humano

Conjunto de conocimientos, experiencias, habilidades, sentimientos, actitudes, motivaciones, valores y capacidad para hacer, portados por los trabajadores para crear más riquezas con eficiencia. (1)

1.2.2 Estimulación Moral y Material

Estimulación material: Sistema de acciones que interactúan y se integran con la estimulación moral, para motivar a los trabajadores en el logro de la eficiencia y eficacia y en la consecución de los objetivos estratégicos de la organización. El pago con arreglo al trabajo, por cantidad y calidad, es el elemento principal de la estimulación material. (1)

Estimulación moral: Sistema de acciones que se realizan para propiciar el desarrollo de la moral socialista en el trabajo y el sentido de pertenencia; reconocer y promover el aporte laboral de los trabajadores en la consecución de los objetivos estratégicos y la elevación de la cultura de la organización, así como la satisfacción individual y colectiva de los trabajadores. (1)

1.2.3 Concepto de pago

El concepto de pago es la caracterización del pago en términos de su naturaleza y del tratamiento de los parámetros de tiempo, importe, acumulación de vacaciones, aplicación de impuestos y procedimientos de contabilización. De modo que deben definirse tantos conceptos de pago como combinaciones de estas componentes se identifiquen. (2)

1.2.4 Impuesto

En este concepto se recogen los impuestos y contribuciones que debe aportar la entidad al presupuesto del estado, la diferencia está en que existen dos tipos los empresariales con base el total de lo pagado a los trabajadores y los salariales que son un aporte del trabajador tienen como base el salario devengado de este.

1.2.5 Nómina

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

Según la Resolución No. 13-2007 del Ministerio De Finanzas y Precios de nuestro país el objetivo de la nómina es relacionar a todos los trabajadores de la entidad que perciban salarios y que les correspondan haberes por concepto de: sueldos, jornales, primas, vinculación, vacaciones, licencias y subsidios, obteniéndose la conformidad del cobro efectuado mediante la firma en este documento, siempre y cuando no se ejecute por Tarjetas Magnéticas. (3)

1.2.6 Salario

Salario: Parte del producto nacional que se distribuye a los trabajadores de forma individual, atendiendo a la cantidad y calidad del trabajo aportado, según las condiciones económicas de cada momento histórico. Comprende lo percibido por el trabajador, por rendimiento, unidad de tiempo, pagos adicionales, trabajo extraordinario, laborar en día de conmemoración nacional y feriados, y vacaciones anuales pagadas. (1)

Salario básico: Remuneración que comprende la tarifa de la escala salarial, más los pagos adicionales establecidos legalmente, y se utiliza en los casos previstos en la legislación. (1)

Salario escala: Cantidad de dinero, en moneda nacional de curso legal, que corresponde a cada grupo de la escala de complejidad. (1)

1.2.7 Submayores

Es un registro donde se analizan las subcuentas y las cuentas que lo requieran. En ellos se reflejan todos los movimientos que tienen las cuentas desde su apertura.

1.2.8 Retenciones

Son los adeudos contraídos por el trabajador que se descuentan periódicamente, según se determine mensual, quincenal o semanal, del salario devengado por el trabajador.

1.3 Proceso de Confección y Cálculo de la Nómina

Uno de los procesos que se efectúa en el departamento de Recursos Humanos es el pago de los trabajadores en activo existentes en la entidad mediante la Confección y cálculo de la nómina. Dentro del mismo se desarrollan los subprocesos siguientes:

- Apertura de nómina
- Procesar Nómina
- Cierre de Periodo.

Se realiza basado en el resumen de incidencia de cada trabajador en el periodo de pago a analizar. Inicialmente se crean el Mayor de trabajadores, los Submayores correspondientes (submayor de vacaciones y submayor de retenciones), el Fichero histórico devengado y el Registro de salario y tiempo

de servicio. Luego se procede a realizar las nóminas correspondientes a Salarios, Subsidios, Vacaciones, entre otros, en el período que se establezca, con las informaciones obtenidas de la pre-nómina que contienen las incidencias procesadas por el área de personal; las retenciones de los trabajadores utilizando las notificaciones bancarias y los movimientos de nómina que permiten actualizar el maestro de trabajadores.

Cuando se realiza la nómina se tiene en cuenta si el trabajador tiene alguna retención verificándolo con el submayor de retenciones, en caso de tener se le descuenta y se actualiza este submayor al terminar el procesamiento de la nómina; si la nómina es de vacaciones se verifica la duración de las vacaciones que tomará el trabajador para sacar proporcionalmente el importe al tiempo de acuerdo a lo que tiene acumulado en el submayor de vacaciones y esta cantidad es lo que se le pagará, actualizando este submayor al concluir el procesamiento; si la nómina es de subsidio se debe tomar del fichero histórico devengado el acumulado del tiempo trabajado en los meses anteriores para el cálculo de la misma. Al terminar se actualizan los submayores de vacaciones con la provisión de vacaciones acumuladas en ese período y el de retenciones con las respectivas deducciones descontadas al trabajador; además de los submayores se actualiza el registro de salario y tiempo de servicio y se obtiene la distribución de moneda para el pedido del efectivo al banco.

La información de las nóminas se enviará mediante el comprobante de operaciones al subsistema de Contabilidad con el propósito de realizar la contabilización de las mismas y se obtendrá además la distribución de moneda para la solicitud del efectivo al banco.

Finalmente se procede a hacer el cierre del periodo contable para lo cual se comprueba la actualización de los submayores realizando la conciliación con el importe de las cuentas del departamento de Contabilidad, se verifica que todas las nóminas estén confirmadas y todos los comprobantes de operaciones estén confirmados.

1.3.1 Mapa de relaciones internas

En este mapa se representan los procesos de negocio y sus interacciones que ayudará a una mejor comprensión del negocio.

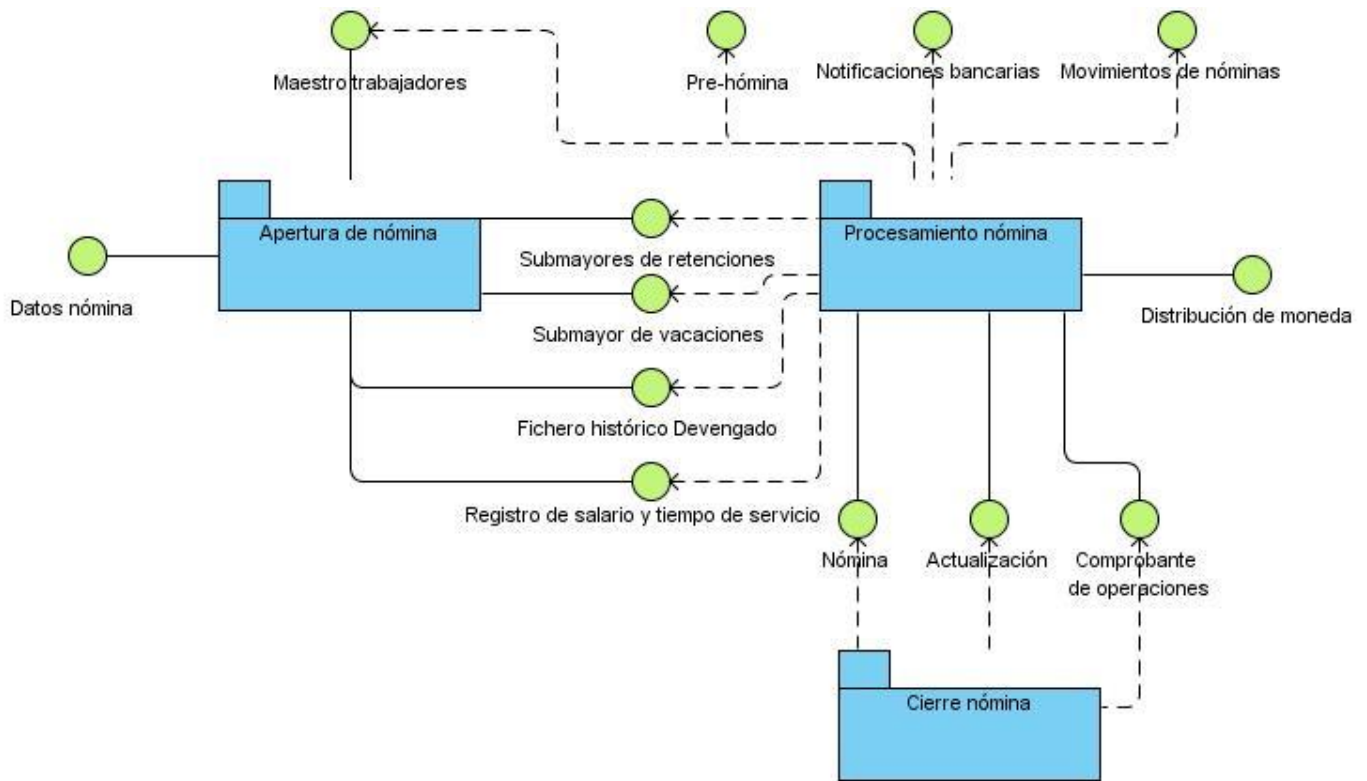


Figura 1 Mapa de relaciones internas.

La figura 1 muestra las relaciones internas del proceso Confección y cálculo de la nómina. Llámese relaciones internas a todos aquellos procesos o entidades, con los que el proceso intercambia información, la saeta discontinua indica que el flujo de información es de entrada al proceso, la ausencia de estas indica que flujo de información sale del proceso. A continuación se explican estas relaciones.

Apertura de nómina

El proceso de Apertura es el que registra los datos de los trabajadores en activo existentes en la entidad creando el Maestro y los Submayores correspondientes los cuales son el submayor de vacaciones, el submayor de retenciones, el fichero histórico devengado y el registro de salario y tiempo de servicio.

Procesar Nómina

Este proceso comprende todo lo relacionado con el Confección y cálculo de las nóminas correspondientes a Salarios, Subsidios, Vacaciones, entre otras, en el período que se establezca con las informaciones obtenidas de la prenómina que contiene las incidencias procesadas por el área de personal; las retenciones de los trabajadores utilizando las notificaciones bancarias y los movimientos de nómina que mantienen actualizado el maestro de trabajadores.

El salario básico del trabajador es el que le corresponde recibir por su trabajo y lo devenga durante el tiempo que permanezca laborando. El salario devengado es aquel que resulta de la realización de minoraciones al salario básico del trabajador, por ausencias, llegadas tarde, enfermedad, y otras causas.

A medida que el trabajador labora se va creando una provisión para retribuirle un período proporcional de descanso (a base de 1 mes de descanso por cada 11 meses de trabajo, o sea, aproximadamente el 9.09% del salario) que va actualizando el submayor de vacaciones una vez procesada la nómina.

Del importe de salario devengado por los trabajadores se pueden efectuar determinados descuentos por conceptos que están plenamente autorizados por la legislación por fallos de tribunales de justicia o mediante autorización expresa del trabajador. Las deducciones o retenciones realizadas a los salarios devengados por los trabajadores, hasta su pago a los beneficiarios o el Banco, se analiza por cada trabajador y por cada tipo de deducción.

Independientemente del pago del salario al trabajador, así como de los períodos de vacaciones, las empresas están en la obligación de contribuir al Presupuesto del Estado con el por ciento establecido (según el tipo de empresa) de todos los salarios pagados más el importe por las vacaciones acumuladas. Este aporte se entrega mensualmente al Presupuesto del Estado y constituye un gasto para la empresa.

Cuando se realiza la nómina se tiene en cuenta si el trabajador tiene alguna retención verificándolo con el submayor de retenciones en caso de tener se le descuenta y se actualiza este submayor al terminar el procesamiento de la nómina; si la nómina es de vacaciones se verifica la duración de las vacaciones que tomará el trabajador para sacar proporcionalmente el importe al tiempo de acuerdo a lo que tiene acumulado en el submayor de vacaciones y esta cantidad es lo que se le paga, actualizando este submayor al concluir el procesamiento; si la nómina es de subsidio se debe tomar del fichero histórico devengado el acumulado del tiempo trabajado en los meses anteriores para el cálculo de la misma. Al terminar se actualiza además de los submayores el registro de salario y tiempo de servicio y se obtiene la distribución de moneda para el pedido del efectivo al banco.

Finalmente se emite un comprobante de pago, por trabajador, en cada nómina, el que se adjunta al sobre del efectivo o se entrega al trabajador para su conocimiento cuando el pago se realiza por medio de tarjeta magnética.

Cierre de Período

El proceso de Cierre de Período se comprueba la actualización de los submayores realizando la conciliación con el importe de las cuentas en Contabilidad, se verifica que todas las nóminas estén confirmadas y todos los comprobantes de operaciones estén confirmados.

1.3.2 Mapa de relaciones externas

En este mapa se representan las relaciones que tiene el proceso de la Nómina con otros procesos de Capital Humano y otros subsistemas.

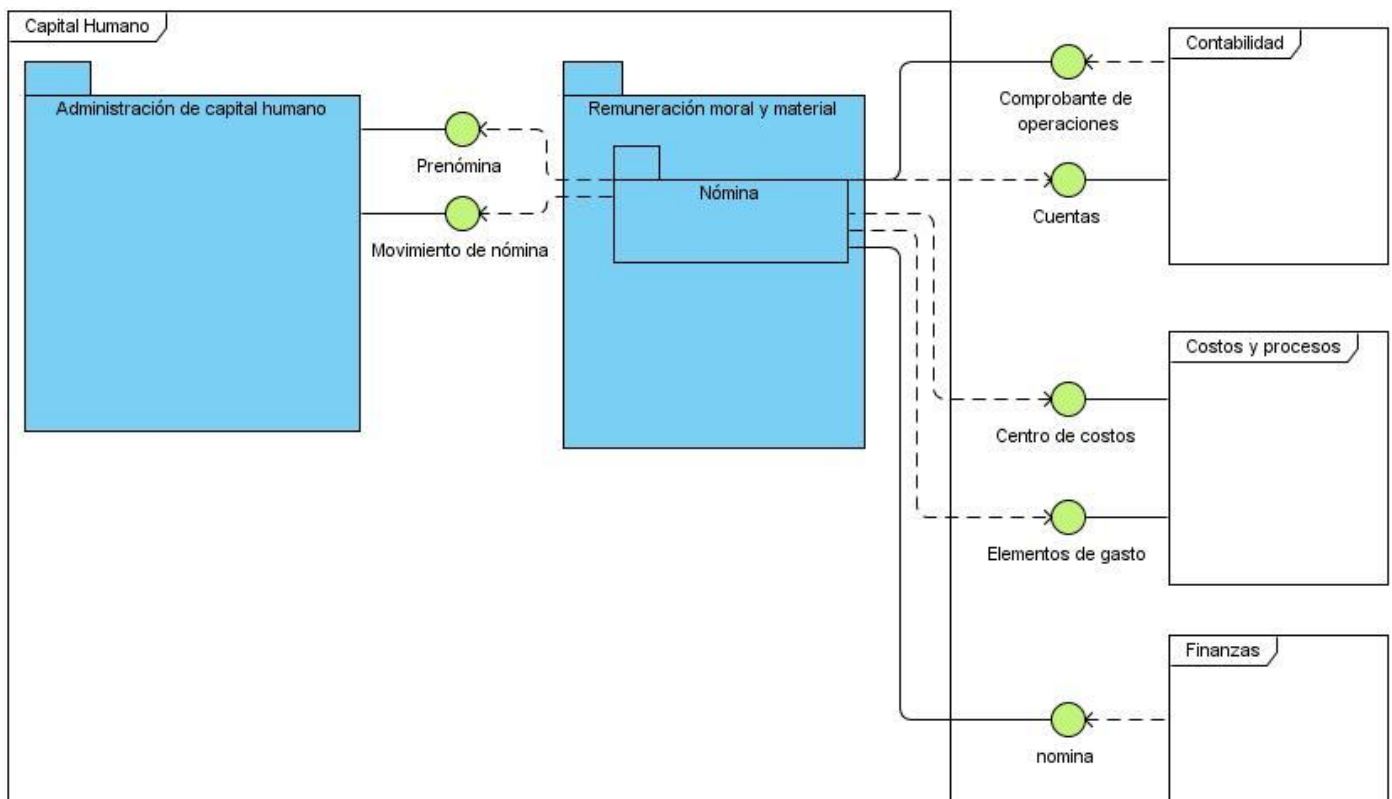


Figura 2 Mapa de relaciones externas.

La figura 2 muestra las relaciones externas del proceso Confección y cálculo de la nómina. Llámese relaciones externas a todos aquellos procesos o entidades que no pertenecen al módulo de Remuneración moral y material con los que el proceso intercambia información. A continuación se explican estas relaciones.

- Relación con el módulo Administración de capital humano

En el módulo de Administración de Capital Humano se realiza la pre-nómina y los movimientos de nómina que son utilizados por el módulo Remuneración moral y material. La pre-nómina es un documento donde se registran las incidencias de los trabajadores del periodo a analizar, esta es utilizada para realizar el cálculo de la nómina. Los movimientos de nómina son documentos que tienen como objetivo informar todo el movimiento de personal que se realice y produzca modificaciones en su estructura salarial, cargo o área de trabajo, constituyendo el documento que respalda el registro para mantener actualizados los datos que sirven de fuente para la preparación y pago de las nóminas.

- Relación con el subsistema Contabilidad

En vista de realizar la contabilización del pago de los trabajadores se debe tener una configuración de las cuentas en el componente Nómina, con este propósito se utiliza el nomenclador de cuenta definido en el subsistema de Contabilidad. Luego de terminado el cálculo de la nómina se informa a Contabilidad mediante un comprobante de operaciones las cuentas que han sido afectadas durante este procesamiento.

- Relación con el subsistema Costos y Proceso

En vista de realizar la contabilización de los costos en el cálculo de la nómina se utilizan los nomencladores de centro de costo y elemento de gasto definidos en el subsistema Costos y Procesos.

- Relación con el subsistema Finanzas

Luego de terminado el procesamiento de la nómina se debe brindar esta información al subsistema de Finanzas con el objetivo de que se realice la solicitud del efectivo al banco para el posterior pago de los trabajadores.

1.4 Soluciones existentes

En el mundo actual existen numerosos sistemas contables con el objetivo de dar una solución factible para la ejecución de los cálculos que se realizan durante el procesamiento de la nómina. En los siguientes epígrafes se realiza un análisis de algunos sistemas investigados sobre las funcionalidades que brindan, las plataformas en que están desarrollados, y quienes son los autores de los mismos.

1.4.1 Sistemas similares en el área internacional.

ASSETS NS

Assets S.A. es una compañía de desarrollo de software registrada en la República de Panamá con capital italiano, sus operaciones fundamentalmente se realizan en el área de América Latina con representaciones en Italia, Panamá, Cuba y en República Dominicana donde se halla la Casa Matriz.

ASSETS NS es un Sistema de Gestión Integral estándar y parametrizado que permite el control de los procesos de Compras, Ventas, Producción, Taller, Inventario, Finanzas, Contabilidad, Presupuesto, Activos Fijos, Útiles y Herramientas y Recursos Humanos. Como Sistema Integral todos sus módulos trabajan en estrecha relación, generando, automáticamente, al Módulo de Contabilidad los Comprobantes de Operaciones por cada una de las transacciones efectuadas, esto permite que se pueda trabajar bajo el principio de Contabilidad al Día.

El Módulo de Recursos Humanos Versión 3.1 desarrollada en Access 97 está concebido para calcular las nóminas y controlar los recursos laborales de una entidad. El sistema puede calcular y contabilizar nóminas de salario para cualquier tipo de pago (sueldo fijo, jornal, pago por rendimiento), incluyendo el pago de horas extras, interrupciones y condiciones laborales anormales. De igual forma, calcula y contabiliza las nóminas de vacaciones y subsidios y las nominillas de diferentes tipos (salario, vacaciones, subsidios, reintegros, estimulación, entre otras). Realiza deducciones automáticas de cualquiera de las nóminas que se calculen. Todos los procesos automáticos se realizan siguiendo los criterios y restricciones establecidos por la legislación laboral vigente (determinación del fondo de tiempo, cálculo de las vacaciones, salario promedio para el subsidio, etc.). La estructura salarial de los empleados permite considerar el pago de la idoneidad y de retribuciones complementarias móviles (régimen salarial del sector del turismo).

Una vez calculadas y pagadas las nóminas, el Módulo de Recursos Humanos le permite generar automáticamente los Comprobantes de Operaciones a la Contabilidad. Se generarán por separado los comprobantes de Salario, Vacaciones, Subsidios, Nominillas y Reintegros.

Las opciones que brinda este módulo para la nómina son las siguientes:

- Pagos por Concepto de Salario a sueldistas, tarifa horaria y pago a vinculados
- Pagos de Vacaciones
- Pagos de Subsidios
- Pagos a Interrumpidos
- Pagos de Horas Extras
- Pagos por Condiciones Laborales Anormales
- Nominillas
- Nómina de Retenciones
- Reintegros de Salario, Subsidio y Vacaciones
- Submayor de Vacaciones
- Registro de Salario y Tiempo de Servicio
- Submayor de Retenciones
- Nómina de Divisa
- Pre-nómina
- Nómina de Divisa
- Desglose de Efectivo para Pago de Nómina
- Solicitud de Fondos
- Análisis Estadísticos

Debido a las necesidades existentes en el país, y a que ASSETS NS es un software extranjero, lo cual significa gastar recursos monetarios en adquirirlo, este sistema no se puede utilizar como solución al

problema que se expone. Además para su instalación se requiere que sea en el sistema operativo Windows utilizando el Access 97.

SAGE MAS 500

Sage Software fue quien desarrollo el sistema Sage MAS 500, es una empresa que radica en California EEUU. Esta software ofrece servicios de software de automatización de los procesos de recursos humanos y la nómina, permiten automatizar las actividades y el cumplimiento de estas por los recursos humanos, además de permitir realizar un seguimiento de la asistencia, los empleados y de la nómina.

Abra nómina es el módulo que permite realizar el proceso de la nómina y la presentación de informes. Este está integrado con el sistema de recursos humanos y permite la gestión del plan de asistencia para actualizar automáticamente con el procesamiento de nóminas. La funcionalidad de Abra nómina permite producir cheques de pago en la demanda, y hacer cambios de última hora. Es adaptable a todas las organizaciones permitiendo incluir los ingresos ilimitados, impuestos, tipos de deducción, definidos por el usuario de múltiples grupos de pagar.

Abra nómina también ofrece una gama completa de capacidades de gestión de impuestos, incluyendo todos los estándares de las tablas de impuestos, la compatibilidad con el EFTPS (Sistema Electrónico de Pago de Impuestos Federales), y los medios electrónicos de información para todos los estados. Además, ofrece más de 100 informes estándar, así como el informe de consulta de Abra segura escritor y Crystal Reports opciones de presentación de informes que proporcionan acceso a información crítica para el negocio y garantizar el cumplimiento de las normas de presentación de informes del gobierno.

Abra nómina responde a sus necesidades actuales con un programa de apoyo superior al cliente, que incluye todas las actualizaciones trimestrales de impuestos, apoyo en línea, mejoras de productos, soporte técnico telefónico ilimitado, y muchos más. Altamente escalable y flexible para satisfacer las cambiantes necesidades de su negocio, Abra Nómina proporciona integración completa con Sage MAS 500 ERP, recursos humanos Abra, Abra Asistencia y Abra SEE. Abra Nómina ofrece un superior rendimiento de la inversión, en la mayoría de los casos el pago por sí mismo en un año o menos.

En resumen el módulo de Abra nómina ofrece:

- Simplificar la organización de procesamiento de la nómina.
- Gestión de requisitos de presentación de informes.
- Usar la completa funcionalidad de gestión tributaria que le ayudarán a permanecer compatible.
- Crear ingresos ilimitados y la deducción códigos.
- Crear automáticamente archivos ACH (Automated Clearing House) para procesar las transacciones de depósito directo.
- Asignar un número ilimitado de cuentas de depósito directo para sus empleados.
- Vista previa de impresión antes de los controles, controles de impresión de prueba, y simplificar aún más comprobar la impresión láser con MICR (Magnetic Ink Character Recognition) y opciones de la firma.
- Eliminar la duplicación de la entrada de datos.

Sage MAS 500 es un software extranjero no certificado, desarrollado para plataforma de software propietario. Para que funcione correctamente necesita de una integración con el resto de los subsistemas, es decir para que el funcionamiento sea el mas optimo posible debería de comprarse el sistema completo, siendo poco útil adquirir solamente algunos módulos, puesto que estos necesitarían de una interrelación con algún otro quedando funcionalidades sin poder utilizar. Debido a lo anteriormente expuesto este sistema no se puede utilizar como una solución posible al problema.

SAP

SAP AG (Sistemas, Aplicaciones y Productos) es una empresa alemana y uno de los proveedores de software empresarial más importantes del mundo. Uno de sus principales productos es SAP ERP, este proporciona a sus usuarios la tecnología necesaria para contar con una amplia visión de todas las actividades de su negocio. De esta manera, sus decisiones serán más acertadas, debido a que reflejarán la situación de la empresa en el momento oportuno y no deberán de basarse en estimaciones o información obsoleta.

SAP ERP está basado en una plataforma abierta que proporcionara un completo control sobre la operativa y estrategia empresarial. Mejora, al mismo tiempo, la productividad pues proporciona la flexibilidad necesaria para adaptar la estrategia corporativa a las necesidades empresariales cambiantes, basándose en información clara y en tiempo real. Entre las funcionalidades que SAP ERP ofrece, cabe destacar:

Análisis empresarial: Permite evaluar el rendimiento del negocio aprovechando las funcionalidades para el análisis de personal, operaciones y cadena de suministro. Todo esto a la mano de un estricto control contable y encaminado a llevar a cabo las estrategias empresariales.

Contabilidad financiera e interna: Permite gestionar las funciones financieras de la empresa mediante la automatización de la gestión de la cadena de suministro, la contabilidad financiera y la contabilidad interna. Esta funcionalidad la proporciona específicamente SAP ERP Financials.

Gestión del capital humano: Proporciona las herramientas necesarias para maximizar el potencial de rentabilidad de su personal mediante varias funcionalidades para la gestión de transacciones con los empleados y de su ciclo de vida. Esta funcionalidad la proporciona específicamente SAP ERP Human Capital Management.

Gestión de operaciones: Permite optimizar las operaciones con una funcionalidad integrada para gestionar procesos logísticos completos, a la vez que amplía las capacidades colaborativas en la gestión de la cadena de suministro, ciclo de vida del producto y el manejo de las relaciones con los proveedores. Esta funcionalidad la proporciona específicamente SAP ERP Operations.

Gestión de servicios corporativos: Permite optimizar los servicios centralizados y descentralizados de la gestión de bienes inmuebles, viajes de empresa e incentivos y comisiones. Esta funcionalidad la proporciona específicamente SAP ERP Corporate Services.

Autoservicios: Este es un portal orientado a los empleados y directivos para que éstos puedan crear, ver y modificar información clave en los procesos de gestión de operaciones y estrategia empresarial. El portal utiliza una amplia gama de tecnologías de interacción entre las que se encuentran navegadores,

dispositivos de voz y móviles que permiten que éstos accedan fácilmente al contenido empresarial, aplicaciones, servicios internos y externos.

El módulo Gestión del Capital Humano (SAP ERP Human Capital Management) comprende distintos procesos como la planificación de recursos necesarios, el reclutamiento y la selección, la gestión del personal, la liquidación de haberes, el desarrollo del personal y la capacitación.

Los principales subcomponentes son los siguientes:

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

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

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

SAP ERP aun estando certificado para su posible utilización en Cuba, no es una solución factible al problema que se expone, debido a que el país no cuenta con los recursos financieros suficientes como para adquirirlo y sus licencias tienen un alto valor monetario. Además de que es un software extranjero y desarrollado para plataforma de software propietario.

Seven2000

Seven2000 es un producto de SQL EVEN SEVEN C.A. que es una empresa radicada en Valencia, Estado Carabobo, Venezuela que brinda asesoría en las áreas de Desarrollo de Aplicaciones (Oracle Developer, Oracle Designer) así como en las áreas de Administración de Base de Datos y Entonación de Aplicaciones en distintas plataformas (Unix, Windows/NT, Linux, etc.). También tiene varios productos denominados Seven2000, que han sido instalados tanto en las empresas privadas como en las empresas públicas.

Seven2000 es un sistema integrado compuesto por módulos que pueden actuar separados o integrados. El módulo de Sistema de Finanzas es un sistema de manejo Financiero (Contabilidad, Cuentas por

Pagar, Cuentas por Cobrar, Bancos, Activo Fijo, Control Presupuestario). Multi-Empresa/ Multi-Moneda/ Moneda alterna.

El sistema de Comercialización es un sistema de manejo Comercial. Compras, Inventarios y Facturación. Integrados con los Auxiliares Contables. Multi-Empresa/Multi-Moneda/ Moneda alterna.

El Sistema de nómina y Recursos Humanos permite la total automatización de los procesos asociados al control de los pagos del personal y mantenimiento de historias de los trabajadores.

Algunas características de Seven2000:

- El subsistema de pagos de nóminas es un proceso interactivo orientado hacia el usuario final. La nómina se encuentra disponible en todo momento y su proceso es por excepciones, es decir, solo se calcularán a aquellas personas que han tenido cambios para la nómina que esta vigente.
- Permite procesar un número ilimitado de conceptos. La codificación de los conceptos es definida por el usuario.
- La definición de la forma de calcular estos conceptos es totalmente parametrizada y no requiere intervención de personal de informática (los cambios los realiza el usuario del sistema).
- Conceptos Fijos por Trabajador.
- Centro de Costo.
- Distribución de Moneda. Cartas de Depósitos Bancarios.
- Permite procesar múltiples contratos de nómina siendo la definición de estas determinada por el usuario (semanal, quincenal, mensual, confidencial, contratado, etc.).
- Permite manejar diferentes turnos de trabajo los cuales serán igualmente determinados por el usuario (fijos, rotativos, eventuales, planificada por mes/año).
- Permite la rotación automática por Grupos, para aquellas empresas de proceso ininterrumpido.
- Diseñado para evitar la acumulación de documentos dentro del Departamento de Personal, de forma tal que permite poder incluir variaciones de nómina a futuro (no se debe esperar a ultimo momento) o en el momento que ocurren y se procesaran dependiendo de la fecha.

Algunos Reportes que ofrece este:

- Resumen de nómina.
- Depósitos Bancarios.
- Resumen de Conceptos.
- Inventario Uniformes.
- Movimiento de Uniformes.
- Ficha de Personal.
- Control de Inasistencias.
- Reportes Min. Trabajo.
- Nomina Calculada.
- Generación de Archivos.
- Recibo de Pago (Sobres, Cheques, Cartas al Banco, Medios Magnéticos).
- Resumen de Nómina.

Seven2000 no es una solución factible al problema expuesto, debido a que es un software extranjero no certificado, desarrollado para plataforma de software propietario. Además no presenta una gestión de Recursos Humanos integrado a la nómina; uno de los objetivos que se desea lograr con la realización de este trabajo es que la nómina se realice como parte de la gestión de Recursos Humanos.

1.4.2 Sistemas similares en el área nacional.

RODAS XXI

El Sistema Integral Económico Administrativo RODAS XXI desarrollado por la empresa CITMATEL posibilita automatizar el funcionamiento de cualquier empresa o unidad presupuestada.

RODAS XXI es un sistema multi-empresa que cuenta actualmente con seis módulos: Finanzas, Contabilidad, Activos Fijos, Nóminas, Inventario y Facturación. Estos módulos pueden emplearse

integrados en su totalidad, formando cualquier subconjunto entre ellos, o cada uno de forma independiente.

En el módulo de nómina para la obtención de las nóminas en cada uno de los períodos de año el sistema divide las operaciones en dos pasos fundamentales, el cálculo de nóminas y su emisión. Como resultado del cálculo de cada nómina y su comprobante el sistema muestra como quedarían en caso de emitirse dicha nómina, en caso de detectar algún error u omisión en las incidencias agregadas a los trabajadores que les corresponda o en el comprobante calculado se pueden realizar las correcciones pertinentes calculándola nuevamente hasta que todo está correcto y una vez logrado esto puede procederse a emitirla lo que implica que en el sistema a nivel de módulo se registrará la nómina emitida en el período que se está trabajando así como su comprobante correspondiente. Una vez emitida la nómina el comprobante puede ser exportado al módulo de contabilidad facilitando en gran medida el trabajo del contador.

El sistema es capaz de guardar por trabajador los pagos y retenciones fijas que se realizan a cada uno de ellos por lo que para la confección de las nóminas cada mes sólo es necesario actualizar las incidencias que correspondan y todo el trabajo posterior de cálculo es realizado de forma automática.

EL submayor de vacaciones, el de retenciones y el de decreto ley 91 son generados automáticamente por el sistema al igual que los salarios devengados y las retenciones por trabajador. También son generados de forma automática los modelos SNC 4-2-25 y SNC 4-1-25, el reporte de bajas, el resumen de otros pagos y de salarios devengados. Siempre que se desee se pueden ver los reportes correspondientes a cada una de las nóminas emitidas, la nómina en sí, su comprobante, los sobres para pago, el desglose de efectivo, las retenciones realizadas.

Este módulo permite además visualizar información correspondiente a períodos anteriores, tan sólo con cambiar de período contable a periodos anteriores ya cerrados, aunque en dichos periodos no podrá realizar ninguna operación.

El sistema contable RODAS XXI aunque tiene funcionalidades muy útiles como lo es la de guardar por trabajador los pagos y retenciones que se le realizan y solo actualizarlas cada mes para la posterior elaboración de la nómina, no es una solución factible al problema expuesto, aun siendo un producto nacional. Esto se debe a que esta desarrollado para plataforma de software propietario y no presenta una gestión de recursos humanos integrada a la nómina.

VERSAT SARASOLA

El sistema **VERSAT-Sarasola** está desarrollado por la Empresa de Tecnologías de la Información y el Conocimiento (TEICO) del Ministerio del Azúcar en una plataforma de Delfi con servidor SQL Server, esta constituido por 10 módulos o subsistemas que incluyen configuración y seguridad, contabilidad general y de gastos, costos y procesos, finanzas y caja, activos fijos, planificación y presupuestos, control de inventarios, pago de salario (nómina), facturación y generador de reportes. **Es un paquete para la gestión económica financiera se** distingue por ser el primer sistema de contabilidad cubano certificado, según las nuevas normativas establecidas por los Ministerios de Finanzas y Precios y de la Informática y las Comunicaciones, para este tipo de Software.

El mismo, está diseñado para ser empleado en cualquier tipo de entidad empresarial o presupuestada. Permite llevar a cabo el control y registro contable e 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, incluyéndose el inventario de una entidad.

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. Logra establecer un proceso de interacción usuario-sistema y posee una gran rapidez y agilidad a partir de la configuración del proceso de contabilización de los documentos primarios y de las propias posibilidades de trabajo contenidas en cada subsistema.

El subsistema Nómina de Salarios es lo suficientemente configurable a los diversos sistemas de pago, estimulaciones y demás regulaciones laborales que existen en el país. Posee además todo lo relacionado con el descuento, liquidación y submayor de retenciones, así como vacaciones acumuladas y SNC- 2-25. Se puede configurar el sistema definiendo de antemano todas las variantes de impuestos, bonificaciones, condiciones anormales, retenciones, descuentos, penalizaciones, etc. con las denominaciones y los parámetros numéricos que se requieran. Mediante los conceptos de pago puede definir todas las variantes de cálculos y procedimientos contables que se presenten en su entidad y designarlas con las denominaciones más naturales.

El sistema contable VERSAT Sarasola presenta funciones útiles como lo es la de poder configurar el sistema definiendo de antemano todas las variantes de impuestos, pagos adicionales y retenciones. Aun

así no es una solución viable debido a que está desarrollado para software propietario y no presenta una gestión de recursos humanos integrado a la nómina. Y al igual que otros de los sistemas analizados anteriormente se deben comprar las licencias del mismo, viéndose en la necesidad de comprar el sistema en su totalidad para que funcione correctamente, debido a que necesita de una integración con el resto de los subsistemas, siendo poco útil adquirir solamente algunos módulos, puesto que estos requerirían de una interrelación con algún otro quedando funcionalidades sin poder utilizar.

1.5 Modelo de desarrollo orientado a componentes

Debido a la creciente demanda de software, y a que estos se han convertido en un factor muy importante para la evolución de la sociedad actual, se vio la necesidad de crear procedimientos que sirviesen para guiar y organizar el desarrollo de software, basándose en métodos comunes; estos procedimientos son conocidos como metodologías. Estas definen los roles que van a intervenir, los artefactos que se van a obtener y las actividades que se van a realizar durante todo el proceso de desarrollo del software logrando mayor productividad del mismo, teniendo estos un alto nivel de calidad así como las cualidades tanto físicas como funcionales requeridas.

Un proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo" (4)

El Modelo de desarrollo orientado a componentes es la metodología definida por el proyecto ERP-Cuba, para la realización de este trabajo. Este es un modelo de desarrollo orientado a las necesidades y artefactos generados durante el proceso de desarrollo del proyecto ERP-Cuba. Es una combinación de diferentes metodologías de las cuales se ha tomado lo que sería más conveniente para llevar a término el proyecto. Entre las características que posee se encuentran que se modela el negocio mediante procesos; la ingeniería de requisitos es mucho más clara que en otras metodologías; es orientada a componentes, posibilitando la independencia de funciones del sistema a la hora de mantener o modificar el sistema funcional; como lenguaje de modelado para el negocio se utiliza BPMN y para el diseño UML.

Este modelo de desarrollo permitirá la generación de artefactos de vital importancia en el análisis y el diseño como son: Modelo de proceso de negocio, Descripción de procesos de negocio, Modelo

conceptual, Prototipo de interfaz de usuario, Especificación de requisitos, Casos de prueba, Diagrama de clases y Descripción del diseño de clases.

1.6 Lenguaje de Modelado

La creciente complejidad de los sistemas informáticos representa un reto importante para los ingenieros y arquitectos del software. De la preocupación inicial sobre la definición de la estructura y calidad del código final, se ha pasado a dedicar cada vez más tiempo, atención y esfuerzo al diseño y modelado del sistema. Los modelos proporcionan un mayor nivel de abstracción, permitiendo trabajar con sistemas mayores y más complejos, y facilitando el proceso de codificación e implementación del sistema de forma distribuida y en distintas plataformas.

Un modelo es una descripción de (parte de) un sistema, descrito en un lenguaje bien definido. Un lenguaje bien definido es un lenguaje con una sintaxis y semántica precisa, y que puede ser interpretado y manipulado por un ordenador. (5)

1.6.1 Notación para el Modelado de Procesos del Negocio

La Notación para el Modelado de Procesos del Negocio (BPMN: Business Process Modeling Notation) es un estándar de modelado de procesos de negocio, en donde se presentan gráficamente las diferentes etapas del proceso del mismo. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes.

Es muy utilizado a nivel mundial y considerado un estándar avanzado. BPMN ayuda a definir, documentar y hacer mucho más eficiente los procesos del negocio. Su principal objetivo es proveer una notación estándar facilitando el entendimiento de las colaboraciones y transacciones de negocio para todos los involucrados e interesados del negocio. Entre estos interesados están los analistas de negocio (quienes definen y redefinen los procesos), los desarrolladores técnicos (responsables de implementar los procesos) y los gerentes y administradores del negocio (quienes monitorean y gestionan los procesos). Tiene como finalidad servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación.

En BPMN el modelado es realizado mediante diagramas muy simples con un conjunto muy pequeño de elementos gráficos. Con esto se busca que para los usuarios del negocio y los desarrolladores técnicos sea fácil entender el flujo y el proceso. Las cuatro categorías básicas de elementos son:

- Objetos de flujo: Eventos, Actividades, Rombos de control de flujo (Gateways)

- Objetos de conexión: Flujo de Secuencia, Flujo de Mensaje, Asociación
- Swimlanes (Carriles de piscina): Pool, Lane
- Artefactos: Objetos de Datos, Grupo, Anotación

Con estas cuatro categorías de elemento se puede realizar un diagrama simple de procesos de negocio (BPD: Business Process Diagram).

1.6.2 Lenguaje de Modelado Unificado

El Lenguaje de Modelado Unificado (UML: Unified Modeling Language), es un lenguaje gráfico para especificar, construir y documentar los artefactos que modelan un sistema. Fue diseñado para ser un lenguaje de modelado de propósito general, por lo que puede utilizarse para especificar la mayoría de los sistemas basados en objetos o en componentes, y para modelar aplicaciones de muy diversos dominios de aplicación (telecomunicaciones, comercio, sanidad, etc.) y plataformas de objetos distribuidos (como por ejemplo J2EE, .NET o CORBA). Al ser un lenguaje de modelado de propósito general proporciona una gran flexibilidad y expresividad a la hora de modelar sistemas.

UML ayuda al usuario a entender la realidad de la tecnología y la posibilidad de que reflexione antes de invertir y gastar grandes cantidades en proyectos que no estén seguros en su desarrollo, reduciendo el coste y el tiempo empleado en la construcción de las piezas que constituirán el modelo. (5)

Cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

Modelo estático (estructural):

- Diagrama de despliegue.
- Diagrama de componentes.
- Diagrama de clases.
- Diagrama de objetos.

Modelo dinámico (comportamiento):

- Diagrama de estados.
- Diagrama de actividades.

- Diagrama de secuencia.
- Diagrama de colaboración.
- Diagrama de casos de uso.

Los diagramas a utilizar en la solución del problema serán el diagrama de clases y el de componentes.

A continuación se presentan algunas de las principales características del lenguaje UML.

- Es un lenguaje distribuido y adecuado a las necesidades de conectividad actual y futura, por lo que es ampliamente utilizado por la industria desde su adopción por OMG (Object Management Group). Reemplaza a decenas de notaciones empleadas con otros lenguajes.
- Permite el modelado de estructuras complejas.
- Las estructuras más importantes que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clase, componentes y nodos.
- Emplea operaciones abstractas como guía para variaciones futuras, añadiendo variables si es necesario. Otra característica que presenta es el comportamiento del sistema: casos de uso, diagramas de secuencia y de colaboraciones, que sirven para evaluar el estado de las máquinas.

1.6.3 Comparación entre UML y BPMN

El lenguaje UML toma un perfil orientado a objetos en el modelado de aplicaciones, mientras que BPMN toma un perfil orientado a procesos en el modelado de sistemas. Además BPMN tiene un enfoque específico a los procesos de negocio y UML se enfoca más al diseño del software; siendo ambas notaciones totalmente compatibles entre sí.

Las extensiones de UML para el modelado de negocio aportan elementos muy importantes ya que proporcionan algunas otras vistas de la arquitectura de negocio que son más difíciles de observar usando únicamente BPMN. Por ejemplo, la visualización de las responsabilidades de los trabajadores del negocio, la manipulación de las entidades del negocio y la comprensión de los estados asociados a las entidades del negocio.

Por las razones antes expuestas el modelado del negocio se realizara utilizando BPMN y para el diseño el Lenguaje de Modelado Unificado, aunque como también se ha visto UML puede ser utilizado en el modelado del negocio.

1.7 Herramientas de Modelado

El múltiple desarrollo de software ha influido grandemente en la aparición de novedosas herramientas automatizadas para ayudar a los diseñadores a realizar el proceso de análisis y diseño del software, haciendo posible que tenga una mejora en la calidad, fiabilidad, utilidad y rendimiento del mismo.

Una de las herramientas mas utilizadas en estos tiempos son las herramientas CASE (Computer Aided Software Engineering) o Ingeniería de Software Asistida por Computadora. Estas representan un conjunto de programas y ayudas que dan asistencia a los desarrolladores con el objetivo de automatizar los aspectos claves de todo el proceso de desarrollo de un sistema, en todas o algunas de sus fases.

1.7.1 Visual Paradigm for UML 6.1 Enterprise Edition

Visual Paradigm es una herramienta CASE profesional .Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite tanto la ingeniería directa como inversa, soportada en varios lenguajes de programación. Además de permitir el modelado de los procesos en UML y BPMN.

Permite generar:

- Diagramas de Casos de Uso
- Diagramas de Clases
- Diagramas de Secuencia
- Diagramas de Comunicación
- Diagramas de Estado
- Diagramas de Componentes
- Diagramas de Despliegue
- Diagramas de Objetos
- Diagramas de Interacción
- Diagramas de Entidad Relación

- Diagramas ORM
- Diagramas de Procesos del Negocio
- Diagramas de EJB
- Diagramas de visión general

De estos diagramas se utilizarán el diagrama de clases y de procesos del negocio en el análisis y diseño del presente trabajo.

Es una herramienta colaborativa porque soporta a varios usuarios trabajando en un mismo proyecto, genera la documentación del proyecto automáticamente en varios formatos como son Web o .Pdf, y permite control de versiones. Brinda la posibilidad de generar código a partir de los diagramas, para plataformas como .Net, Java y PHP, así como obtener diagramas a partir de código. Esta es precisamente una gran ventaja puesto que el sistema será desarrollado en PHP.

Visual Paradigm para UML es multiplataforma, lo cual le permite al usuario utilizar esta herramienta en varios sistemas operativos como Windows, Linux, Unix y otros; además se encuentra disponible en distintas versiones: Enterprise, Professional, Standard, Modeler, Personal y Community. Facilitando también las licencias especiales para fines académicos.

Entre sus características se encuentra que es muy fácil de instalar y actualizar. Posee compatibilidad entre sus ediciones y soporte de UML versión 2.1. Además soporta aplicaciones web. Debido a las funcionalidades y ventajas brindadas por el Visual Paradigm y por ser una herramienta multiplataforma que se integra fácilmente con varios IDEs (IDE: Integrated Development Environment [Ambiente Integrado de Desarrollo]) se decide utilizarlo como herramienta CASE para la definición del software.

1.8 Propuesta de lenguajes de programación a utilizar para el desarrollo del sistema

Desde el inicio de la red de redes Internet, surgieron diferentes demandas por los usuarios y dándose soluciones a estas mediante lenguajes estáticos. Con el transcurso del tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación para la web dinámica, que permitieran interactuar con los usuarios y utilizaran sistemas de

bases de datos. Estos son conocidos como lenguajes del lado del servidor, lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Entre los lenguajes de programación web más conocidos se encuentran PHP, Perl, Python y JAVA.

1.8.1 PHP 5

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. Es usado principalmente en interpretación del lado del servidor (server-side scripting). Las siglas PHP son un acrónimo recursivo que significa “PHP Hypertext Pre-processor”, aunque inicialmente se llamó Personal Home Page.

Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP.

A continuación se presentan algunas características del lenguaje de programación PHP.

- Este lenguaje posee la capacidad de expandir su potencial utilizando la enorme cantidad de módulos que tiene, además de una biblioteca nativa de funciones sumamente amplia e incluida.
- PHP 5 es un lenguaje libre por lo que se presenta como una alternativa de fácil acceso para todos y además multiplataforma siendo esto una de las principales características por lo que es muy usado en el mundo.
- Permite las técnicas de Programación Orientada a Objetos; no requiere definición de tipos de variables y tiene manejo de excepciones.
- Presenta la capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.

Para el desarrollo del sistema se utilizara el este lenguaje de programación, puesto que el conjunto de características antes expuestas lo hacen ideal para la posterior elaboración de la aplicación.

1.9 Sistemas gestores de bases de datos

Los sistemas gestores de base de datos constituyen un conjunto de programas que administran y gestionan la información contenida en una base de datos. Estos van a poder manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de los datos. Permitiendo almacenar enormes cantidades de datos de forma organizada y que se encontrarán disponibles para los usuarios que la requieran de forma rápida y a la vez protegida contra usuarios malintencionados. Entre los sistemas Gestores de Bases de Datos más usados en la actualidad se encuentran MySQL, Oracle, SQL Server, PostgreSQL, Interbase y otros.

1.9.1 PostgreSQL 8.3

Es un sistema de gestión de base de datos relacional orientada a objetos de software libre. Cuenta con una arquitectura probada que se ha ganado una sólida reputación de confiabilidad, corrección e integridad de los datos. Es altamente escalable, tanto en la enorme cantidad de datos que puede administrar como en el número de usuarios concurrentes que puede acomodar. Se distribuye bajo licencia BSD, lo cual permite el uso y distribución sin costo, tanto para aplicaciones comerciales como no comerciales.

PostgreSQL es la primera base de datos de código abierto en implementar Recorrido Sincronizado, que reduce el uso de E/S (entrada/salida) en aplicaciones de minería de datos.

Características principales de PostgreSQL

- Búsqueda en texto: una herramienta avanzada de búsqueda en texto, TSearch2, que ha sido incorporada en la distribución central, con mejor manejo y nuevos diccionarios e idiomas.
- Posee una alta concurrencia, es decir mediante un sistema denominado MVCC (Acceso concurrente multiversión) permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.
- Los disparadores (trigger): un disparador o trigger es una acción específica basada en algo ocuriente dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica.
- Presenta soporte SQL/XML de acuerdo al estándar ANSI, incluyendo exportación en formato XML.
- Presenta claves ajenas también denominadas Llaves ajenas o Claves Foráneas (*foreign keys*).
- Tiene un soporte de autenticación GSSAPI y SSPI nuevos tipos de datos: UUIDs, ENUMs y arreglos de tipos compuestos.

Para el desarrollo del sistema se utilizara PostgreSQL 8.3 en la creación de la base de datos, debido al conjunto de características antes expuestas.

1.10 Servidor Web

Un servidor web básicamente, sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante HTTP (hypertext transfer protocol).

1.10.1 Servidor Web Apache 2.0

Un servidor web Apache es un servidor HTTP (hypertext transfer protocol) de código abierto, es decir es un programa que se ejecuta continuamente desde una computadora (que será el llamado servidor) e interpreta las peticiones HTTP que recibe por parte de un cliente (un navegador de Internet) y las satisface. Su nombre se debe a la tribu Apache que fue la última en caer en manos de lo que luego sería el gobierno de EE.UU; para indicar la firmeza y robustez del servidor Apache.

Entre las características que presenta Apache 2.0 se encuentra que corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal. Los módulos de Apache pueden escribirse para que se comporten como filtros que actúan sobre el flujo de contenidos tal y como salen del servidor o tal y como son recibidos por el servidor. Posee mensajes de error en diferentes idiomas, tiene un soporte mejorado para las plataformas que no son tipo Unix y cuenta con la infraestructura necesaria para servir distintos protocolos.

Apache trabaja con gran cantidad de Perl, PHP y otros lenguajes de script. También trabaja con Java y páginas jsp, teniendo todo el soporte que se necesita para tener páginas dinámicas. Este es un servidor altamente configurable de diseño modular, siendo muy sencillo ampliar sus capacidades. Permite la creación de ficheros de log a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor. Apache es una tecnología gratuita de código fuente abierto lo que permite que se pueda ver todo que tiene dentro sin que sea un secreto.

Debido a que el servidor web Apache, es flexible, robusto y teniendo en cuenta las características antes expuestas, se utilizara como servidor web en el posterior desarrollo del sistema.

1.11 Zend Estudio para Eclipse

Es un entorno integrado de desarrollo (IDE) orientado a desarrollar aplicaciones web, en lenguaje PHP. El programa completo está escrito en Java, lo que ha permitido que este disponible para diferentes plataformas.

Zend Studio es el único entorno de desarrollo integrado disponible para desarrolladores profesionales que abarca todas las capacidades de edición de código necesario para desarrollar modernas aplicaciones empresariales. Incluye un poderoso editor de código con soporte para todos los formatos web. Estas características se combinan para proporcionar simplicidad y productividad sin precedentes para los desarrolladores.

Proporciona una completa depuración de las pruebas y el apoyo que le ayuda a diagnosticar y solucionar problemas de forma rápida y sencilla. Con un fuerte apoyo para la depuración, la elaboración de perfiles, el código de cobertura y PHPUnit; Zend Estudio tiene todas las herramientas que un desarrollador necesita para asegurarse de que el código es correcto y empezar a resolver problemas.

Debido a lo anteriormente expuesto se utilizara como entorno de desarrollo para la implementación de la solución propuesta.

1.12 Conclusiones parciales del capítulo.

En el presente capítulo se abordaron los conceptos principales a tratar, para un mejor entendimiento del negocio a modelar. Se realizó un análisis de sistemas que presentan características similares al que se desea realizar, teniendo en cuenta sus funcionalidades, explicando los inconvenientes que presentan y por lo que no pueden ser considerados como propuestas de soluciones. Se describe el proceso de Confección y cálculo de nómina; además se exponen las principales características de la metodología de desarrollo, el lenguaje de modelado y las herramientas a utilizar conformando la fundamentación teórica que sustenta el desarrollo del trabajo.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En el presente capítulo se realiza un estudio preliminar del sistema. Una vez identificado los procesos de negocio se procede a realizar el modelo conceptual con el objetivo identificar las entidades del negocio. Se especifican las técnicas y patrones utilizados en la captura de requisitos y se identifican y describen los requisitos funcionales que debe presentar la solución a construir. Se validan los requerimientos a través los prototipos de interfaz de usuario que representan una propuesta para el diseño de interfaz de la solución a construir.

2.2 Técnicas y métodos empleados en la captura de requisitos

El proceso de recopilar, analizar y verificar las necesidades del cliente o usuario para un sistema es llamado ingeniería de requerimientos. La meta de la ingeniería de requerimientos es entregar una especificación de requisitos de software correcta y completa, por ello durante esta etapa el equipo de desarrollo hace uso de técnicas que le permitan obtener, documentar y desarrollar todas las tareas de una forma más sencilla. Muchas de estas técnicas son muy útiles y fundamentales para capturar los requisitos del software de manera que permitan obtener la mayor cantidad de información posible del sistema a automatizar. A continuación se explican brevemente las empleadas en la captura de requisitos.

2.2.1 Entrevistas

Las entrevistas y cuestionarios se emplean para reunir información proveniente de personas o de grupos. Durante la entrevista, el analista conversa con el encuestado para obtener información deseada. 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. El éxito de esta técnica, depende de la habilidad del entrevistador y de su preparación para la misma. Para su realización 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 misma. Para realizar las entrevistas, conviene llevar preparado un cuestionario.

2.2.1 Prototipos

Los prototipos son simulaciones del posible producto, que luego son utilizados por el usuario final, permitiendo conseguir una importante retroalimentación en cuanto a si el sistema diseñado con base a los requerimientos recolectados le permite al usuario realizar su trabajo de manera eficiente y efectiva. El método de prototipo es muy útil en la captura de los requisitos porque permite al usuario hacerse una idea de la estructura de la interfaz del sistema pudiéndose incorporar ideas o sugerencias que este haga en función de que el producto final cumpla con sus expectativas.

Esta técnica tiene la desventaja de que el usuario piense que así es como quedara la versión final del sistema sin entender que lo que está viendo es solo un prototipo.

2.2.2 Glosario

La diversidad de personas que forman parte de un proyecto de software hace que sea necesario establecer un marco de terminología común. Esta necesidad se vuelve más patente en los sistemas de información web puesto que el equipo de desarrollo en ellas suele ser más interdisciplinario. Por esta razón son muchas las propuestas que abogan por desarrollar un glosario de términos.

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

2.2.3 Sistemas Existentes

Esta técnica consiste en analizar distintos sistemas ya desarrollados que estén relacionados con el sistema a ser construido. Por un lado, se pueden analizar las interfaces de usuario, observando el tipo

de información que se maneja y cómo es manejada. Además de poder analizar las funciones que presenta muchas de las cuales podrían ser muy útiles en sistema a desarrollar. Esto puede ser útil para descubrir información importante a tener en cuenta, información que tal vez el cliente/usuario haya fallado en comunicar.

Es recomendable que luego de haber analizado el sistema, este sea mostrado al cliente/usuario, ya que por su experiencia puede sugerir importantes ideas nuevas, permitiendo que el sistema final cumpla con las exigencias requeridas por el usuario.

2.2.4 Arqueología de documentos

La utilización de esta técnica depende de la información que se tenga almacenada en las entidades con respecto a cómo se realizan los procesos dentro de las mismas. Las entidades guardan información acerca de sus procesos, los modelos o informes necesarios para su desarrollo o para rendir cuenta a los organismos superiores.

Con esta técnica los analistas se tratan de determinar posibles requerimientos sobre la base de analizar la documentación utilizada por la empresa. Se debe recolectar cualquier formulario o documento que sea utilizado para registrar o enviar información. Este método por si solo no es efectivo, sirve más que nada como complemento de otras técnicas, y ayuda a obtener información que de otra manera sería sumamente difícil conseguir.

2.3 Patrones empleados

Los patrones representan una colaboración parametrizada que se puede emplear en múltiples ocasiones dentro de uno o más sistemas. Para ser un patrón, la colaboración tiene que ser utilizable en una amplia gama de situaciones, para justificar darle un nombre. Un patrón es una solución que demostradamente funciona en un cierto número de situaciones. (4)

2.3.1 Patrones CRUD

Acrónimo de Create-Read-Update-Delete (Crear-Leer-Modificar-Eliminar). Conocido como el padre de todos los patrones de capa de acceso. Describe que cada objeto debe ser creado en la base de datos

para que sea persistente. Una vez creado, la capa de acceso debe tener una forma de leerlo para poder actualizarlo o simplemente borrarlo.

Teóricamente el borrado de objetos debería quedar a cargo de la misma base de datos. Pero un recolector de objetos “basura” en una base de datos gigante afecta en gran medida el rendimiento. Por ello es que la tarea de borrado queda delegada al programador.

2.4 Modelo conceptual

El modelo conceptual no es una descripción de los componentes del software; representa los conceptos en el dominio del problema en el mundo real. (7) En el siguiente modelo conceptual se representan las entidades que conforman el proceso nómina del subsistema Capital Humano.

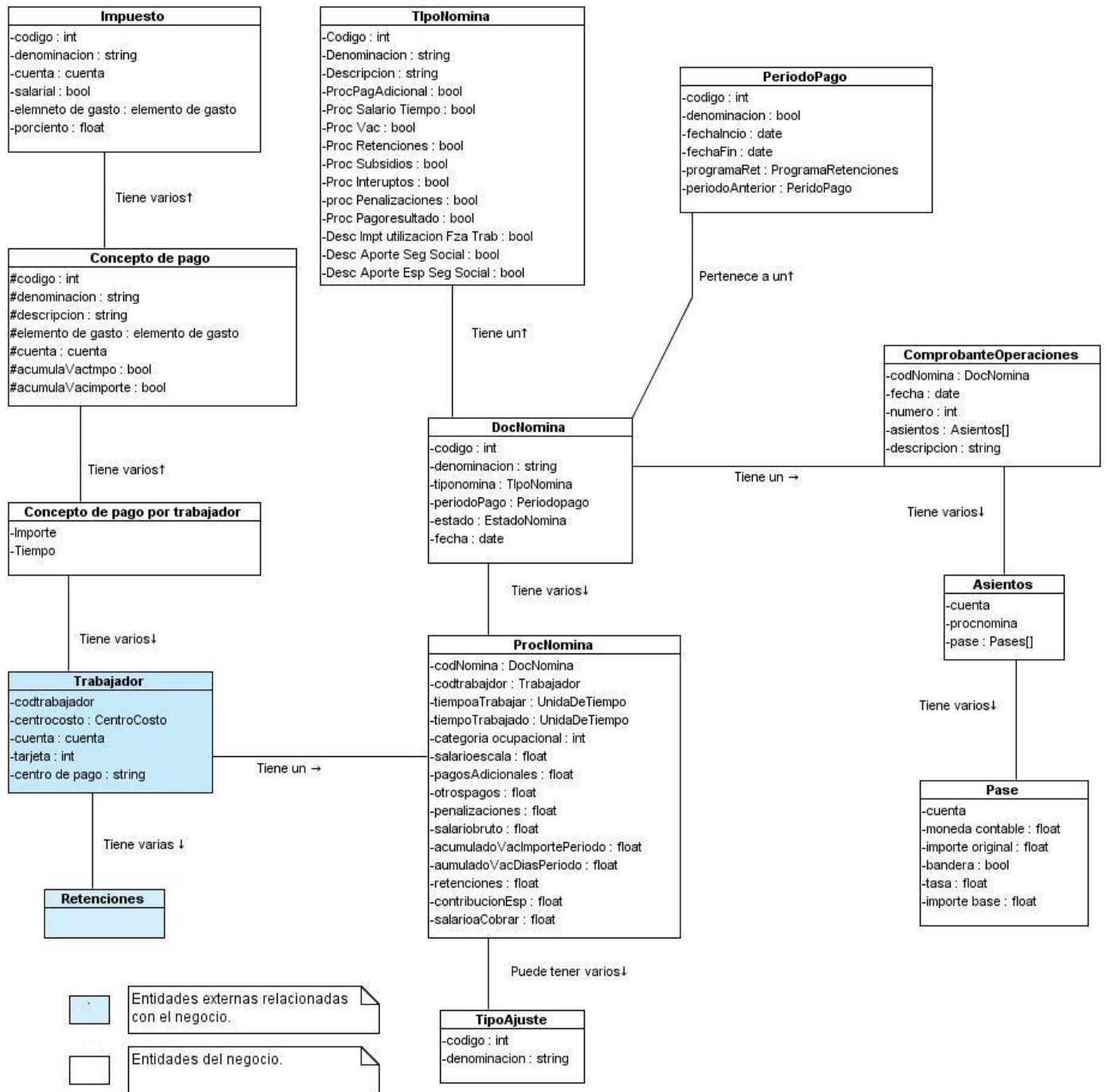


Figura 3 Modelo conceptual

La figura 3 muestra el modelo conceptual del componente Nómina del subsistema Capital humano en donde se representan todas las entidades que lo conforman.

Comprobante de operaciones

En este se registran las operaciones contables que se realizan en una nómina. En un comprobante de operaciones están registradas las operaciones en forma de asientos es decir un comprobante contiene varios asientos.

Asientos

En esta entidad se registran todos los pases u operaciones realizados sobre una cuenta. Un asiento es el resumen de todas las operaciones o pases realizados sobre una cuenta.

Pase

En este se registran las operaciones realizadas sobre una cuenta determinada por el asiento al que pertenece. Se registra además otros detalles de la operación como importe base, moneda contable.

Concepto de pago

Los conceptos de pago son por los cuales los trabajadores son remunerados y deben de ser contabilizados. Ejemplo Subsidios, vacaciones, salario, pagos adicionales y pago por resultado. Un concepto de pago puede tener varios impuestos asociados por los cuales la entidad debe pagar al estado. Además un trabajador puede tener vario conceptos de pago por los cuales se le pagará al procesar la nómina.

DocNómina

El documento de nómina (o nómina solamente) es el documento que se crea para procesar los pagos de los trabajadores. Sirve para definir el procesamiento que se le dará (tipo de nómina) a los conceptos de pago que se procesarán en el periodo de pago. Por cada trabajador en el DocNómina se tendrá un ProcNómina diferente.

Período de pago

En esta entidad se definen los distintos periodos de pago de la empresa. Un periodo de pago es un lapso de tiempo puede ser una semana, 15 días o un mes.

Impuestos

En este se definen los distintos tipos de Impuestos empresariales y salariales a aplicar en la empresa y por los cuales debe aportar al presupuesto del estado por cada concepto de pago definido en la entidad.

ProcNómina

En este se realiza el procesamiento de cada trabajador por cada nómina. Aquí se registra el resultado de los cálculos de los conceptos de pagos, además de las retenciones y otros descuentos. En caso de que se haya cometido algún error en el cálculo este se puede corregir mediante un ajuste de un tipo de los definidos en la entidad.

Tipo de nómina

En esta entidad se definen los tipos de nómina. El tipo de nómina determina el procesamiento que se le dará a las nóminas de cada tipo al ser calculadas.

Tipo de ajuste

En esta entidad se definen los tipos de ajustes. Los ajustes se utilizan para corregir los errores al reportarles los pagos a los trabajadores después de procesada la nómina.

2.5 Requisitos funcionales

Según la IEEE Standard Glossary of Software Engineering Terminology un requerimiento se define como condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.

Los requerimientos son una necesidad documentada sobre el contenido forma o funcionalidad de un producto. Estos establecen las funcionalidades y restricciones requeridas que debe de tener el sistema

para satisfacer las necesidades del usuarios/cliente al que está dirigido el software. Los requisitos se clasifican en funcionales y no funcionales. Los requerimientos funcionales definen las funciones que el sistema debe de tener para la obtención del producto final.

<p>RF- 1</p>	<p>Gestionar impuestos y contribuciones</p> <p>En el requisito se gestionan los tipos de impuestos que se tendrán en una entidad y que luego estos serán asociados a una determinada incidencia.</p> <p>1.1 Adicionar tipos de impuesto y contribuciones 1.2 Modificar tipos de impuesto y contribuciones 1.3 Eliminar tipos de impuesto y contribuciones 1.4 Mostrar tipos de impuesto y contribuciones</p>
<p>RF- 2</p>	<p>Gestionar conceptos de pago</p> <p>En el requisito se gestionan los conceptos de pago por los cuales se les pagará a los trabajadores de la entidad.</p> <p>2.1 Adicionar conceptos de pago 2.2 Modificar conceptos de pago 2.3 Eliminar conceptos de pago 2.4 Mostrar conceptos de pago 2.5 Definir impuesto y contribuciones por concepto</p>
<p>RF-3</p>	<p>Gestionar tipos de nómina</p> <p>En el requisito se gestionan los tipos de nómina por los cuales se determinan los pasos a seguir para procesar la nómina.</p> <p>3.1 Adicionar tipos de nómina 3.2 Modificar tipos de nómina 3.3 Eliminar tipos de nómina 3.4 Mostrar tipos de nómina</p>

RF-4	<p>Gestionar periodo de pago</p> <p>En el requisito se gestionan los períodos de pagos; un período puede ser un mes calendario, o una quincena, o una decena, o en fin, cualquier rango de fecha. Todo período de pago debe estar enteramente comprendido dentro de un período contable, aunque diferentes períodos contables pueden estructurarse de diferente forma en períodos de pago.</p> <p>4.1 Adicionar período de pago 4.2 Modificar período de pago 4.3 Eliminar período de pago 4.4 Mostrar período de pago 4.5 Generar periodos de pago</p>
RF-5	<p>Gestionar tipos de ajustes</p> <p>En el requisito se gestionan los tipos de ajustes para justificar una modificación realizada sobre un concepto de pago en una nómina dada.</p> <p>5.1 Adicionar tipo de ajuste 5.2 Modificar tipo de ajuste 5.3 Eliminar tipo de ajuste 5.4 Mostrar tipo de ajuste</p>
RF-6	<p>Administrar nómina</p> <p>En el requisito se permite administrar la nómina agrupando las funcionalidades de crear, modificar y eliminar procesar; además de permitir agregar y eliminar trabajadores a una nómina.</p> <p>6.1 Crear nómina 6.2 Modificar nómina 6.3 Eliminar nómina 6.4 Mostrar nóminas</p>

	<p>6.5 Agregar trabajadores a la nómina</p> <p>6.6 Eliminar trabajadores a la nómina</p> <p>6.7 Procesar nómina</p> <p>6.8 Filtrar por periodo</p>
RF-7	<p>Revisar nómina</p> <p>En el requisito se revisan las nóminas comprobando que estas estén procesadas correctamente.</p>
	<p>7.1 Mostrar nóminas</p> <p>7.2 Mostrar detalles del trabajador de una nómina</p> <p>7.3 Mostrar comprobante de operaciones de la nómina</p> <p>7.4 Ajustar nómina</p> <p>7.5 Revertir nómina</p> <p>7.6 Confirmar nómina</p>
RF-8	<p>Comprobante de operaciones</p> <p>En el requisito se permite revisar los detalles del comprobante de operaciones, que este va a contener todas las operaciones contables realizadas en una nómina; y si la misma esta correcta esta se envía a contabilidad.</p>
	<p>8.1 Detalles del comprobante de operaciones</p> <p>8.2 Emitir comprobante de operaciones</p>
RF-9	<p>Cierre</p> <p>En el requisito se realizan las validaciones necesarias para poder cerrar el período contable.</p>
	<p>9.1 Efectuar cierre contable</p>
RF-10	<p>Configuración</p> <p>En el requisito se realiza la configuración de las cuentas para la</p>

	contabilización general de la nómina.
	10.1 Establecer configuración de cuentas.
RF-11	<p>Recuperaciones</p> <p>En el requisito se obtienen un conjunto de listados para brindar información y tener versiones imprimibles.</p>
	<p>11.1 Listado para el pago por tarjeta</p> <p>11.2 Listado para el pago tradicional</p> <p>11.3 Distribución de moneda</p> <p>11.4 Comprobante de operaciones</p>
RF-12	<p>Gestionar frecuencia de pagos</p> <p>En el requisito se permitirá definir las posibles frecuencias con que se pague en la entidad en un mes calendario. Ejemplo si en la entidad se paga de forma quincenal la frecuencia sería de 2 veces por mes.</p>
	<p>12.1 Adicionar frecuencia de pago</p> <p>12.2 Modificar frecuencia de pago</p> <p>12.3 Eliminar frecuencia de pago</p> <p>12.4 Mostrar frecuencia de pago</p>

2.6 Especificaciones de requisitos

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

2.6.1 Requisito funcional Gestionar impuesto y contribuciones

Especificación del requisito Adicionar impuesto y contribuciones

Conceptos tratados	Conceptos	Atributos
	Impuesto y contribuciones.	Código, denominación, porcentaje, cuenta de crédito, impuesto salarial, elemento del gasto y otra cuenta.
Precondiciones	Precondiciones	Pre-requisito
	Tiene que estar definidas las cuentas en contabilidad	Definir nomenclador de cuentas en contabilidad.
	Tiene que estar definidos los elementos de gasto en costos y procesos	Definir nomenclador de elemento de gastos en costos y procesos.
Descripción	<p>El sistema permite adicionar los siguientes datos del impuesto:</p> <p>Código: número definido por el usuario que identifica el tipo de impuesto.</p> <p>Denominación: nombre que el usuario le asigna al tipo de impuesto.</p> <p>Porcentaje: un número que representa el porcentaje.</p> <p>Cuenta de crédito: es la cuenta a la cual se va a acreditar el salario devengado que generalmente es la cuenta de Nóminas por Pagar.</p> <p>Impuesto salarial: marcado significa que es un impuesto salarial, sin marcar significa que no lo es.</p> <p>Elemento de Gasto: si se marca se está asumiendo que la contabilización del gasto se va a hacer a la combinación de Cuenta de Gastos y Centro de Costo que se asuma o se especifique en la Nómina.</p> <p>Otra cuenta: Si se marca otra cuenta se asume que ésta no es de gasto, y que por lo tanto no se requiere de Elemento del Gasto para la contabilización.</p> <p>Si el impuesto es un impuesto salarial se deben deshabilitar los campos: Elemento de Gasto. Otra cuenta.</p> <p>Se selecciona adicionar el tipo de impuesto.</p> <p>El sistema valida los datos.</p>	

	<p>En caso de error en la entrada de datos informar al usuario y permitir corregirlos.</p> <p>Si los datos son correctos informar al usuario que se ha adicionado satisfactoriamente el tipo de impuesto.</p>
Validaciones	<p>El sistema valida los datos según lo descrito en el Modelo Conceptual:</p> <p>Si el tipo de impuesto es un impuesto salarial se bloquea la opción de insertar estos atributos:</p> <p>Elemento del Gasto.</p> <p>Otra cuenta.</p>
Post-condiciones	Se ha adicionado un nuevo tipo de impuesto.
Post-requisito	No procede.

Prototipo de interfaz de usuario ver [Anexo 1](#)

Especificación del requisito Modificar impuesto y contribuciones

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

	Si los datos son correctos informar al usuario que se ha modificado satisfactoriamente el tipo de impuesto.
Validaciones	El sistema valida los datos. Se debe garantizar que los datos que no sean posibles modificar aparezcan deshabilitados. Ejemplo: el atributo código.
Post-condiciones	Se ha modificado el tipo de impuesto.
Post-requisito	No procede.

Prototipo de interfaz de usuario ver [Anexo 1](#)

Especificación del requisito Eliminar impuesto y contribuciones

Conceptos tratados	Conceptos	Atributos
	Impuesto y contribuciones	Código, denominación, por ciento, cuenta de crédito, impuesto salarial, elemento del gasto y otra cuenta.
Precondiciones	Precondiciones	Pre-requisito
	Que se haya adicionado un tipo de impuesto.	Adicionar tipo de impuesto.
Descripción	Se escoge el tipo de impuesto a eliminar. Se selecciona eliminar el tipo de impuesto. Se muestra un mensaje al usuario de confirmación. Ejemplo: ¿Desea eliminar el tipo de impuesto? Si se confirma se elimina el tipo de impuesto seleccionado.	
Validaciones	No se pueden eliminar tipos de impuestos respecto de los que se hayan registrado incidencias	
Post-condiciones	Se ha eliminado el tipo de impuesto.	
Post-requisito	No procede.	

Especificación del requisito Mostrar datos de impuesto y contribuciones

Conceptos tratados	Conceptos	Atributos
	Impuesto y contribuciones	Código, denominación, porciento, cuenta de crédito, impuesto salarial, elemento del gasto y otra cuenta.
Precondiciones	Precondiciones	Pre-requisito
	Que se haya adicionado un tipo de impuesto.	Adicionar tipo de impuesto.
Descripción	Se selecciona tipo de impuesto. Se muestra en una tabla los tipos de impuestos definidos.	
Validaciones	No procede.	
Post-condiciones	Se muestran los datos del tipo de impuesto.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario ver [Anexo 1](#)

2.6.2 Requisito funcional Gestionar concepto de pago

Especificación del requisito Adicionar concepto de pago

Conceptos tratados	Conceptos	Atributos
	Concepto de pago	Código, denominación, descripción, elemento de gasto, cuenta, acumula vacaciones tiempo, acumula vacaciones importe
	Impuesto	Código, denominación, por ciento.
Precondiciones	Precondiciones	Pre-requisito
	Tiene que estar definidas las cuentas en contabilidad	Definir nomenclador de cuentas en contabilidad.
	Tiene que estar definidos	Definir nomenclador de elemento de gastos en

	los elementos de gasto en costos y procesos	costos y procesos.
	Tiene que estar definidos los impuestos	Definir impuesto y contribuciones
Descripción	<p>Se selecciona la opción adicionar concepto de pago.</p> <p>El sistema permite llenar los datos:</p> <p>Código: número definido por el usuario que identifica el concepto de pago.</p> <p>Denominación: nombre que el usuario le asigna concepto de pago.</p> <p>Descripción: descripción del concepto de pago</p> <p>Elemento de gasto: elemento de gasto por el cual se contabilizará el concepto de pago.</p> <p>Cuenta: Cuenta de por el cual se contabilizará el concepto de pago.</p> <p>Acumula vacaciones tiempo: se marca se el concepto acumula o no tiempo de vacaciones.</p> <p>Acumula vacaciones importe: Se selecciona el tipo de acumulación del concepto, “No acumula” si no acumula importe, “Acumula” si acumula y “Acumula y Liquida” en caso de que se liquide en la misma nómina los acumulado por este concepto.</p> <p>Se marcan los impuestos a pagar por el concepto de pago que se esta definiendo.</p> <p>Se acepta.</p> <p>El sistema verifica los datos y en caso de haber algún error permite rectificar sino guarda la información.</p>	
Validaciones	<p>Se realizan las validaciones de datos correspondientes con el modelo conceptual.</p> <p>Se verifica que no se repita el código o la denominación.</p>	
Post-condiciones	Se ha creado un nuevo concepto de pago	
Post-requisito	No procede.	

Prototipo de interfaz de usuario ver [Anexo 2](#)

Especificación del requisito Modificar concepto de pago

Conceptos tratados	Conceptos	Atributos
	Concepto de pago	Código, denominación, descripción, elemento de gasto, cuenta, acumula vacaciones tiempo, acumula vacaciones importe
Precondiciones	Precondiciones	Pre-requisito
	Tiene que estar definidos los concepto de pago	Definir concepto de pago
Descripción	<p>Se selecciona la opción de modificar concepto de pago.</p> <p>Se llenan los datos exceptuando el código.</p> <p>Se acepta.</p> <p>El sistema verifica los datos y en caso de haber algún error permite rectificar sino guarda la información.</p>	
Validaciones	Se realizan las validaciones de datos correspondientes con el modelo conceptual.	
Post-condiciones	Se ha modificado el concepto de pago.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario ver [Anexo 2](#)

Especificación del requisito Eliminar concepto de pago

Conceptos tratados	Conceptos	Atributos
	Concepto de pago	Código, denominación, descripción, elemento de gasto, cuenta, acumula vacaciones tiempo, acumula vacaciones importe
Precondiciones	Precondiciones	Pre-requisito

	Tiene que estar definidos los concepto de pago	Definir concepto de pago
Descripción	Se selecciona el concepto de pago a eliminar. Se selecciona la opción eliminar. Se confirmar la eliminación.	
Validaciones	El sistema valida que no quede ninguna nómina sin procesar en al cual se haya utilizado este concepto.	
Post-condiciones	Se ha eliminado un concepto de pago.	
Post-requisito	No procede.	

Especificación del requisito Mostrar Concepto de pago

Conceptos tratados	Conceptos	Atributos
	Concepto de pago	Código, denominación, descripción, elemento de gasto, cuenta, acumula vacaciones tiempo, acumula vacaciones importe
Precondiciones	Precondiciones	Pre-requisito
	Tiene que estar definidos los concepto de pago	Definir concepto de pago
Descripción	Se selecciona concepto de pago. Se muestra en una tabla los conceptos de pagos definidos.	
Validaciones	No procede.	
Post-condiciones	Se muestran los datos del concepto de pago.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario ver [Anexo 2](#)

2.6.3 Requisito funcional Gestionar tipo de nómina

Especificación del Requisito Adicionar tipo de nómina

Conceptos tratados	Conceptos	Atributos
	Tipo de nómina	Código, denominación, descripción Salario escala, Pagos adicionales, Retenciones, Vacaciones, Subsidios, Pago por resultado, Penalizaciones, Contribución a la seguridad social, Impuesto por la utilización de la fuerza de trabajo, Contribución especial a la seguridad social
Precondiciones	Precondiciones	Pre-requisito
	No procede	No procede
Descripción	<p>Al seleccionar nuevo tipo de nómina el sistema permite escribir los datos del tipo de nómina.</p> <p>El sistema permite adicionar los siguientes datos:</p> <p>Código: número definido por el usuario que identifica el tipo de nómina.</p> <p>Denominación: nombre que el usuario le asigna al tipo de nómina.</p> <p>Descripción: descripción del tipo de nómina.</p> <p>Se selecciona la funciones del procesamiento a seguir para este tipo de nómina como por ejemplo:</p> <p>Salario escala: que en la nómina se le pagará el salario escala del trabajador de acuerdo al tiempo trabajado.</p> <p>Pagos adicionales: que se le pagaran los pagos adicionales al trabajador de acuerdo con los cálculos de cada pago adicional.</p> <p>Retenciones: que en la nómina se procesaran las retenciones de los trabajadores.</p> <p>Vacaciones: que se procesaran los reportes de vacaciones de los</p>	

	<p>trabajadores.</p> <p>Subsidios: que se procesaran los pagos por subsidios.</p> <p>Pago por resultado: que se procesaran los pagos por resultados de los trabajadores.</p> <p>Penalizaciones: que se descontaran las penalizaciones de los trabajadores</p> <p>Contribución a la seguridad social e Impuesto por la utilización de la fuerza de trabajo: se realizara el debido descuento y registro de obligaciones con el presupuesto del estado por la cantidad pagada de forma independiente uno del otro.</p> <p>Contribución especial a la seguridad social: de descontara el porciento correspondiente a los trabajadores afectado con esta contribución.</p> <p>Al aceptar se validan los datos, si hay algún error se notifica y se le permite corregirlo, sino se adiciona el tipo de nómina.</p>
Validaciones	<p>Se validan los datos según el modelo conceptual.</p> <p>Una nómina marcada procesar salario no puede estar marcada vacaciones o subsidios.</p>
Post-condiciones	Se ha adicionado un tipo de nómina.
Post-requisito	No procede.

Prototipo de interfaz de usuario ver [Anexo 3](#)

Especificación del Requisito Modificar tipo de nómina

Conceptos tratados	Conceptos	Atributos
	Tipo de nómina	Código, denominación, descripción
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir al menos un tipo de nómina.	Adicionar nómina
Descripción	Al seleccionar modificar tipo de nómina el sistema muestra permite	

	<p>modificar la denominación, o la descripción del tipo de nómina.</p> <p>También se permite modificar los detalles del procesamiento realizándose las mismas validaciones.</p> <p>Al aceptar se muestra un mensaje de confirmación.</p> <p>Si se acepta se validan los datos si hay algún error se notifica y se le permite corregirlo, sino se adiciona el tipo de nómina.</p> <p>Se modifica el tipo de nómina.</p>
Validaciones	El sistema valida que no quede ninguna nómina sin procesar en al cual se haya utilizado este tipo de nómina.
Post-condiciones	Se ha modificado un tipo de nómina.
Post-requisito	No procede.

Prototipo de interfaz de usuario ver [Anexo 3](#)

Especificación del Requisito Eliminar tipo de nómina

Conceptos tratados	Conceptos	Atributos
	Tipo de nómina	Código, denominación, descripción
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir al menos un tipo de nómina	Adicionar nómina
Descripción	<p>Al seleccionar eliminar tipo de nómina el sistema permite elegir que cual desea eliminar.</p> <p>Selecciona el que se quiere eliminar.</p> <p>Al aceptar se muestra un mensaje de confirmación</p> <p>Si se acepta se elimina el tipo de nómina seleccionado.</p>	
Validaciones	El sistema valida que no quede ninguna nómina sin procesar en al cual se haya utilizado este tipo de nómina.	
Post-condiciones	Se ha eliminado un tipo de nómina.	
Post-requisito	No procede.	

Especificación del Requisito Mostrar tipo de nómina

Conceptos tratados	Conceptos	Atributos
	Tipo de nómina	Código, denominación, descripción
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir al menos un tipo de nómina	Adicionar nómina
Descripción	Se selecciona tipo de nómina. Se muestra en una tabla los tipos de nómina definidos.	
Validaciones	No procede.	
Post-condiciones	Se muestran los datos del tipo de nómina.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario ver [Anexo 3](#)

2.6.4 Requisito funcional Gestionar período de pago

Especificación del requisito Adicionar período de pago

Conceptos tratados	Conceptos	Atributos
	Períodos de pago	Denominación, fecha inicio, fecha fin, programa, frecuencia de pago.
	Programa de retenciones	Denominación del programa de retenciones
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir programas de retenciones.	Gestionar programas de retenciones.
	Tiene que existir las frecuencias de pago.	Gestionar frecuencias de pago.

Descripción	<p>Se selecciona la opción adicionar un Período de pago.</p> <p>El sistema permite llenar:</p> <p>Denominación: la denominación del periodo de pago.</p> <p>Fecha inicio: Introducir la fecha inicio del período.</p> <p>Fecha fin: Introducir la fecha fin del período.</p> <p>Programa: seleccionar el programa de retenciones del período de pago.</p> <p>Frecuencia de pago: se selecciona la frecuencia de pago correspondiente a ese período.</p> <p>Se acepta para adicionar el Período de pago.</p> <p>Se muestra un mensaje “Se ha adicionado satisfactoriamente”.</p>
Validaciones	Se validan los datos según el modelo conceptual .
Post-condiciones	Se ha adicionado un Período de pago.
Post-requisito	No procede

Prototipo de interfaz de usuario ver [Anexo 4](#)

Especificación del requisito Modificar período de pago

Conceptos tratados	Conceptos	Atributos
	Período de pago	Denominación, fecha inicio, fecha fin, programa, frecuencia de pago.
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir al menos un período de pago.	Adicionar un período de pago.
Descripción	<p>Se selecciona la opción Modificar un Período de pago.</p> <p>Se realizan las modificaciones necesarias en los datos.</p> <p>Se acepta para efectuar la modificación al período de pago.</p> <p>Mostrar un mensaje de confirmación “Se ha modificado correctamente”.</p>	

	Si se confirma se guardan los cambios.
Validaciones	Se validan los datos según el modelo conceptual. No se puede modificar un período de pago del cual se haya registrado una nómina.
Post-condiciones	Se ha modificado el período de pago.
Post-requisito	No procede.

Prototipo de interfaz de usuario ver [Anexo 4](#)

Especificación del requisito Eliminar Período de pago

Conceptos tratados	Conceptos	Atributos
		Período de pago.
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir al menos un período de pago.	Adicionar un período de pago
Descripción	<p>Se selecciona la opción Eliminar un Período de pago.</p> <p>Se selecciona el período de pago a eliminar.</p> <p>Se muestra un mensaje “¿Está seguro que desea eliminar el período de pago?”</p> <p>Se elimina el período de pago.</p> <p>El sistema muestra un mensaje de confirmación” Se ha eliminado correctamente la retención”.</p>	
Validaciones	No se puede eliminar un período de pago del cual se haya registrado una nómina.	
Post-condiciones	Se ha eliminado un Período de pago, si procede.	

Especificación del Requisito Generar períodos de pago

Conceptos tratados	Conceptos	Atributos
	Período de pago.	Año, periodicidad,
Precondiciones	Precondiciones	Pre-requisito
	No procede	No procede
Descripción	<p>Se selecciona la opción generar período de pago.</p> <p>Se selecciona el año que se desea generar y la periodicidad de los períodos de pago que es de 1 mes o 15 días, se selecciona el programa o los programas de retenciones a aplicar depende de la periodicidad.</p> <p>Se indica generar los períodos.</p> <p>Se acepta la selección realizada.</p> <p>Se generan los períodos de pagos.</p>	
Validaciones	En ese año no se pueden haber definido periodo de pago anteriormente.	
Post-condiciones	Se generan los períodos de pago.	
Post-requisito	No procede.	

Especificación del Requisito Mostrar períodos de pago

Conceptos tratados	Conceptos	Atributos
	Período de pago	Denominación, fecha inicio, fecha fin, programa, frecuencia de pago.
Precondiciones	Precondiciones	Pre-requisito
	No procede	No procede
Descripción	<p>Se selecciona período de pago.</p> <p>Se muestra en una tabla los períodos de pagos definidos.</p>	
Validaciones	No procede.	
Post-condiciones	Se muestran los datos del período de pago.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario ver [Anexo 4](#)

2.6.5 Requisito funcional Gestionar tipos de ajustes

Especificación del requisito Adicionar tipo de ajuste

Conceptos tratados	Conceptos	Atributos
	Tipo de ajuste	Código, denominación, descripción
Precondiciones	Precondiciones	Pre-requisito
	No procede	No procede
Descripción	<p>Se selecciona la opción adicionar un tipo de ajuste. El sistema permite introducir los datos del tipo de ajuste. Código: número definido por el usuario que identifica el tipo de ajuste. Denominación: nombre que el usuario le asigna al tipo de ajuste. Descripción: descripción del tipo de ajuste.</p> <p>Se acepta para adicionar el tipo de ajuste.</p>	
Validaciones	Se validan los datos.	
Post-condiciones	Se ha adicionado un tipo de ajuste.	
Post-requisito	No procede	

Prototipo de interfaz de usuario ver [Anexo 5](#)

Especificación del Requisito Modificar tipo de ajuste

Conceptos tratados	Conceptos	Atributos
	Tipo de ajuste	Código, denominación, descripción
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir al menos un tipo de ajuste.	Adicionar tipo de ajuste

Descripción	<p>Se selecciona la opción modificar un tipo de ajuste.</p> <p>El sistema permite modificar la denominación y descripción para este tipo de ajuste.</p> <p>Se acepta para efectuar la modificación el tipo de ajuste.</p> <p>Mostrar un mensaje de confirmación.</p> <p>Si se confirma se guardan los cambios.</p>
Validaciones	Se validan los datos
Post-condiciones	Se ha modificado un tipo de ajuste.
Post-requisito	No procede

Prototipo de interfaz de usuario ver [Anexo 5](#)

Especificación del Requisito Eliminar tipo de ajuste

Conceptos tratados	Conceptos	Atributos
		Tipo de ajuste
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir al menos un tipo de ajuste.	Adicionar tipo de ajuste
Descripción	<p>Se selecciona la opción eliminar un tipo de ajuste.</p> <p>El sistema permite seleccionar el tipo de ajuste a eliminar.</p> <p>El sistema permite seleccionar la eliminación.</p> <p>El sistema muestra un mensaje de confirmación.</p> <p>Si se acepta se elimina la información.</p>	
Validaciones	No se puede eliminar un tipo de ajuste que tenga documentos de ajuste asociados.	
Post-condiciones	Se ha eliminado un tipo de ajuste.	
Post-requisito	No procede.	

Especificación del Requisito Mostrar tipo de ajuste

Conceptos tratados	Conceptos	Atributos
	Tipo de ajuste	Código, denominación, descripción.
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir al menos un tipo de ajuste.	Adicionar tipo de ajuste
Descripción	Se selecciona tipo de ajuste. Se muestra en una tabla los tipos de ajustes.	
Validaciones	No procede.	
Post-condiciones	Se muestran los datos del tipo de ajuste.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario ver [Anexo 5](#)

2.6.6 Requisito funcional Administrar nómina

Especificación del requisito Crear nómina

Conceptos tratados	Conceptos	Atributos
	Tipo de nómina	Código, denominación
	nómina	Código, denominación, fecha, período de pago, estado, tipo de nómina
	Período de pago	Código, denominación
Precondiciones	Precondiciones	Pre-requisito
	Tiene que estar definido el maestro de trabajadores con trabajadores.	Emitir movimiento de nómina de alta (Gestionar Trabajador)
	Tienen que estar definidos los períodos	Gestionar período de pago

	de pago.	
	Tiene que estar definidos los tipos de nómina.	Gestionar tipos de nómina
Descripción	<p>Al seleccionar la opción gestionar nóminas el sistema permite introducir los datos de la nómina que se necesitan para procesarla.</p> <p>Código: número definido por el usuario que identifica a la nómina.</p> <p>Denominación: nombre que el usuario le asigna a la nómina.</p> <p>Periodo de pago: Se selecciona el período al que pertenece la nómina y el Tipo de nómina: Se selecciona el tipo de nómina.</p> <p>Al aceptar se realizan las validaciones si hay errores se permiten corregirlos sino se crea el documento nómina.</p>	
Validaciones	Los datos se validan según el ERP-CH-Modelo conceptual v1.5	
Post-condiciones	Se ha creado una nómina.	
Post-requisito	No procede	

Prototipo de interfaz de usuario ver [Anexo 6](#)

Especificación del requisito Modificar nómina

Conceptos tratados	Conceptos	Atributos
	Tipo de nómina	Código, denominación
	nómina	Código, denominación, fecha, período de pago, estado, tipo de nómina
	Período de pago	Código, denominación
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir al menos una nómina.	Crear nómina
Descripción	Para modificar una nómina debe seleccionar la nómina y después opción modificar.	

	<p>El sistema muestra un formulario similar al de adicionar donde permite modificar los datos de la nómina excepto denominación el periodo de pago.</p> <p>Al aceptar se realizan las validaciones si hay errores se permiten corregirlos sino se crea el documento nómina.</p>
Validaciones	Los datos se validan según el Modelo conceptual.
Post-condiciones	Se ha modificado una nómina.
Post-requisito	No procede

Prototipo de interfaz de usuario ver [Anexo 6](#)

Especificación del requisito Eliminar nómina

Conceptos tratados	Conceptos	Atributos
	Tipo de nómina	Código, denominación
	nómina	Código, denominación, fecha, período de pago, estado, tipo de nómina
	Período de pago	Código, denominación
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir al menos una nómina.	Crear nómina
Descripción	<p>Si se desea eliminar una nómina se selecciona la nómina que se desea eliminar y se selecciona la opción eliminar.</p> <p>El sistema muestra un cuadro de dialogo solicitando la confirmación de la eliminación, si se acepta se realizan las validaciones y si se puede se elimina.</p>	
Validaciones	Una nómina no se puede eliminar si se ha confirmado.	
Post-condiciones	No procede	
Post-requisito	No procede	

Especificación del requisito Mostrar nóminas

Conceptos tratados	Conceptos	Atributos
	Tipo de nómina	Código, denominación
	nómina	Código, denominación, fecha, período de pago, estado, tipo de nómina
	Período de pago	Código, denominación
Precondiciones	Precondiciones	Pre-requisito
	No procede	No procede
Descripción	Se selecciona administrar nómina. Se muestra en una tabla la nómina creadas en el período seleccionado.	
Validaciones	No procede.	
Post-condiciones	Se muestran los datos de la nómina.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario ver [Anexo 6](#)

Especificación del requisito Agregar trabajadores a la nómina

Conceptos tratados	Conceptos	Atributos
	Trabajador	Expediente del trabajador, nombre y apellidos
	Nómina	Código entidad, Código, denominación, Fecha, Período de pago, Estado, Tipo de nómina. Expediente trabajador, Fondo de tiempo(solo para el cálculo)

		Horas trabajadas, Salario escala, Pagos adicionales Otros pagos, Descuentos(penalizaciones), Salario bruto, Importe acumulado de las vacaciones , Días acumulados de las vacaciones, Retenciones, Contribución especial, Salario neto
Precondiciones	Precondiciones	Pre-requisito
	Tiene que estar creada al menos una nómina a la que se le puedan añadir trabajadores.	Crear documento de nómina
	Tiene que estar definido el maestro de trabajadores con trabajadores.	Emitir movimiento de nómina de alta (Gestionar Trabajador)
Descripción	<p>Al seleccionar esta opción el sistema permite agregarle trabajadores a la nómina seleccionada.</p> <p>Se presenta una pantalla en la cual se permite filtrar por distintos criterios con el objetivo de obtener un trabajador o un grupo de trabajadores.</p> <p>Al aceptar se carga en el listado de trabajadores de la nómina los trabajadores resultantes de la búsqueda.</p> <p>Esto puede hacerse varias veces hasta que se tenga a todos los trabajadores de la nómina seleccionada.</p>	
Validaciones	No se pueden agregar trabajadores a una nómina que este procesada.	
Post-condiciones	Se le han agregado a una nómina los trabajadores.	
Post-requisito	No procede	

Prototipo de interfaz de usuario ver [Anexo 6](#)

Especificación del requisito Eliminar trabajadores a la nómina

Conceptos tratados	Conceptos	Atributos
	Trabajador	Expediente del trabajador, nombre y apellidos
	Nómina	Código entidad, Código, denominación, Fecha, Período de pago, Estado, Tipo de nómina. Expediente trabajador, Fondo de tiempo(solo para el cálculo) Horas trabajadas, Salario escala, Pagos adicionales Otros pagos, Descuentos(penalizaciones), Salario bruto, Importe acumulado de las vacaciones , Días acumulados de las vacaciones, Retenciones, Contribución especial, Salario neto
Precondiciones	Precondiciones	Pre-requisito
	Tiene que estar creada al menos una nómina a la que se le puedan añadir trabajadores.	Crear documento de nómina

	Tiene que estar definido el maestro de trabajadores con trabajadores.	Emitir movimiento de nómina de alta (Gestionar Trabajador)
	Tiene que haber definida una nómina con trabajadores.	Agregar trabajadores a la nómina
Descripción	<p>Al seleccionar esta opción el sistema permite eliminar trabajadores de la nómina seleccionada.</p> <p>Se presenta una pantalla un con el listado de trabajadores de la nómina seleccionada permitiéndole al usuario eliminar los trabajadores que no debería estar en esta nómina.</p> <p>Al aceptar eliminan del listado de trabajadores de la nómina los trabajadores seleccionados.</p> <p>Esto puede hacerse varias veces hasta que se tenga a todos los trabajadores de la nómina seleccionada.</p>	
Validaciones	No se pueden eliminar trabajadores a una nómina que este procesada.	
Post-condiciones	Se le han agregado a una nómina los trabajadores.	
Post-requisito	No procede	

Especificación del requisito Procesar nómina

Conceptos tratados	Conceptos	Atributos
	Tipo nómina	Código, denominación

	nómina	Código entidad, Código, denominación, Fecha, Período de pago, Estado, Tipo de nómina. Expediente trabajador, Fondo de tiempo(solo para el cálculo) Horas trabajadas, Salario escala, Pagos adicionales Otros pagos, Descuentos(penalizaciones), Salario bruto, Importe acumulado de las vacaciones , Días acumulados de las vacaciones, Retenciones, Contribución especial, Salario neto
	Período de pago	Código, denominación, cantidad de horas para sueldistas, cantidad de horas para jornaleros.
	Incidencias	Código, denominación, tipo incidencia
	Trabajador	Expediente trabajador, puesto de trabajo, sistema de pago
	Registro de incidencias	Código de la incidencia, CI(expediente laboral), importe,
Precondiciones	Precondiciones	Pre-requisito
	Tiene que estar definido el maestro de trabajadores con	Emitir movimiento de nómina de alta (Gestionar Trabajador)

	trabajadores.	
	Tiene que estar definidos las incidencias	Gestionar tipos de incidencias
Descripción	<p>Al seleccionar la opción procesar nómina el sistema muestra los documentos de nóminas que hay creadas, se elige cual se desea procesar y se selecciona la opción procesar.</p> <p>El procesamiento esta regido por el tipo de nómina en el cual se registran los pasos que de deben seguir al procesar.</p> <p>Si esta marcado Salario escala: se solicita el salario escala correspondiente al puesto de trabajo de cada trabajador relacionado, también se busca el tiempo trabajado y el que debió trabajar en el módulo de administración de capital humano.</p> <p>Si esta marcado Pagos adicionales: se busca por cada trabajador el importe a pagarle por cada pago adicional, proporcionando los datos necesarios para el cálculo de los mismos.</p> <p>Si esta marcado Vacaciones: se buscan los trabajadores con vacaciones reportadas, completando también el tiempo por el cual saldrá de descanso.</p> <p>Si esta marcado Subsidios: se buscan los trabajadores subsidiados, completando también el importe.</p> <p>Si esta marcado Retenciones: se procesaran las retenciones de los trabajadores descontando del salario a devengar según el orden de prioridad de cada retención.</p> <p>Si esta marcado Pago por resultado: se recogerán los reporte de pagos de los trabajadores procesándose en la nómina correspondiente.</p> <p>Si esta marcado Penalizaciones: se recogerán las penalizaciones activas a los trabajadores descontándosele en la nómina correspondiente.</p> <p>Si esta marcado Contribución a la seguridad social e Impuesto por la utilización de la fuerza de trabajo en la nómina se calcularan estas</p>	

	<p>obligaciones con el presupuesto reflejándose en el comprobante de operaciones correspondiente.</p> <p>Si esta marcado Contribución especial a la seguridad social se le descontara el porciento correspondiente al salario de los trabajadores afectados.</p> <p>Al final se tendrá al salario de los trabajadores.</p>
Validaciones	Los datos se validan según el Modelo conceptual.
Post-condiciones	Se ha procesado la nómina de los trabajadores.
Post-requisito	No procede

Especificación del requisito Filtrar por período

Conceptos tratados	Conceptos	Atributos
	nómina	Código, denominación, fecha, período de pago, estado, tipo de nómina
	Período de pago	Código, denominación
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir al menos una nómina.	Crear nómina
Descripción	El sistema permite filtrar las nóminas por periodo de pago mostrando en pantalla las nóminas que pertenezcan al periodo seleccionado.	
Validaciones	No procede	
Post-condiciones	Se muestran las nóminas según el periodo escogido.	
Post-requisito	No procede	

2.6.7 Requisito funcional Revisar nómina

Especificación del requisito Mostrar nóminas

Conceptos tratados	Conceptos	Atributos
	Nómina	Código, denominación,

		Fecha, Estado.
Precondiciones	Precondiciones	Pre-requisito
	No procede	No procede
Descripción	Se selecciona revisar nómina. Se muestra en una tabla las nóminas creadas.	
Validaciones	No procede.	
Post-condiciones	Se muestran los datos de la nómina.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario ver [Anexo 7](#)

Especificación del requisito Mostrar detalles del trabajador de una nómina

Conceptos tratados	Conceptos	Atributos
	Nómina	Código entidad, Código, denominación, Fecha, Período de pago, Estado, Tipo de nómina. Expediente trabajador, Fondo de tiempo, Horas trabajadas, Salario escala, Pagos adicionales Otros pagos, Penalizaciones, Salario bruto, Importe acumulado de las vacaciones ,

		Días acumulados de las vacaciones, Retenciones, Contribución especial, Salario neto
Precondiciones	Precondiciones	Pre-requisito
	Se tiene que haber procesado las nóminas	Procesar nómina
Descripción	Se selecciona el trabajador de una nómina. Se escoge la opción detalles. Se muestra un formulario con los parámetros del pago del trabajador.	
Validaciones	No procede	
Post-condiciones	Se han mostrado los detalles del pago de un trabajador.	
Post-requisito	No procede	

Prototipo de interfaz de usuario ver [Anexo 7](#)

Especificación del requisito Mostrar comprobante de operaciones de la nómina

Conceptos tratados	Conceptos	Atributos
	Nómina	Código, denominación.
	Comprobante de Operaciones.	Código nómina, número, descripción, fecha, asientos.
	Asientos	Cuenta, procesamiento nómina, pase
	Pases	Cuenta, moneda contable, importe original, bandera, tasa, importe base.
Precondiciones	Precondiciones	Pre-requisito
	Que se hayan procesado la nómina.	Procesar Nómina.

Descripción	<p>Durante el procesamiento de las nóminas es útil ver las operaciones de los comprobantes de las mismas que no son más que los pases a las diferentes cuentas que se afectan.</p> <p>Seleccionar del listado de nóminas procesadas, la que el usuario desee para ver las operaciones del comprobante.</p> <p>El sistema muestra un formulario donde se visualiza las operaciones del comprobante que corresponde a la nómina seleccionada.</p>
Validaciones	No procede.
Post-condiciones	Se ha mostrado el comprobante de operaciones de la nómina.
Post-requisito	No procede.

Especificación del requisito Confirmar nómina

Conceptos tratados	Conceptos	Atributos
	Nómina	<p>Código entidad,</p> <p>Código, denominación,</p> <p>Fecha,</p> <p>Período de pago,</p> <p>Estado,</p> <p>Tipo de nómina.</p> <p>Expediente trabajador,</p> <p>Fondo de tiempo,</p> <p>Horas trabajadas,</p> <p>Salario escala,</p> <p>Pagos adicionales</p> <p>Otros pagos,</p> <p>Penalizaciones,</p> <p>Salario bruto,</p> <p>Importe acumulado de las vacaciones ,</p> <p>Días acumulados de las vacaciones,</p>

		Retenciones, Contribución especial, Salario neto
Precondiciones	Precondiciones	Pre-requisito
	Se tiene que haber procesado las nóminas	Procesar nómina
Descripción	<p>Se selecciona la opción revisar nómina y el sistema muestra las nóminas que se han procesado y se selecciona la nómina que se desea revisar.</p> <p>Al seleccionar la nómina se muestran los trabajadores que están asociados a esta.</p> <p>Se selecciona el trabajador al que se le desea revisar la nómina y se presiona la opción detalles; ahí se revisara que los parámetros de pago del trabajador estén correctos.</p> <p>Si todo está bien se procede a confirmar la nómina.</p>	
Validaciones	No procede	
Post-condiciones	<p>La nómina ha cambiado de estado en caso de que sea confirmada será "Confirmada".</p> <p>Se ha confirmado la nómina.</p>	
Post-requisito	No procede	

Especificación del requisito Revertir nómina

Conceptos tratados	Conceptos	Atributos
	Nómina	Código entidad, Código, denominación, Fecha, Período de pago, Estado, Tipo de nómina. Expediente trabajador,

		Fondo de tiempo, Horas trabajadas, Salario escala, Pagos adicionales Otros pagos, Penalizaciones, Salario bruto, Importe acumulado de las vacaciones , Días acumulados de las vacaciones, Retenciones, Contribución especial, Salario neto
Precondiciones	Precondiciones	Pre-requisito
	Se tiene que haber procesado las nóminas	Procesar nómina
Descripción	<p>Se selecciona la opción revisar nómina y el sistema muestra las nóminas que se han procesado y se selecciona la nómina que se desea revisar. Al seleccionar la nómina se muestran los trabajadores que están asociados a esta.</p> <p>Se selecciona el trabajador al que se le desea revisar la nómina y se presiona la opción detalles; ahí se revisara que los parámetros de pago del trabajador estén correctos.</p> <p>Si se encuentra el mismo error en varios trabajadores la nómina se revierte para que vuelva a ser procesada.</p>	
Validaciones	No procede	
Post-condiciones	Si se revirtió debido a errores el estado de la nómina será "En edición". Se ha revertido la nómina.	
Post-requisito	No procede	

Especificación del requisito Ajustar nómina

Conceptos tratados	Conceptos	Atributos
	Nómina	Código entidad, Código, denominación, Fecha, Período de pago, Estado, Tipo de nómina. Expediente trabajador, Fondo de tiempo, Horas trabajadas, Salario escala, Pagos adicionales Otros pagos, Penalizaciones, Salario bruto, Importe acumulado de las vacaciones , Días acumulados de las vacaciones, Retenciones, Contribución especial, Salario neto
Precondiciones	Precondiciones	Pre-requisito
	Se tiene que haber procesado las nóminas	Procesar nómina
Descripción	Se selecciona la opción revisar nómina y el sistema muestra las nóminas que se han procesado y se selecciona la nómina que se desea revisar. Al seleccionar la nómina se muestran los trabajadores que están asociados a esta. Se selecciona el trabajador al que se le desea revisar la nómina y se presiona la opción detalles; ahí se revisara que los parámetros de pago del	

	trabajador estén correctos. Si existen datos incorrectos en la nómina del trabajador se da la posibilidad de que se corrija por medio de la opción ajuste.
Validaciones	No procede
Post-condiciones	Se ha ajustado la nómina.
Post-requisito	No procede

Prototipo de interfaz de usuario ver [Anexo 7](#)

2.6.8 Requisito funcional Comprobante de operaciones

Especificación del Requisito Emitir comprobante de operaciones

Conceptos tratados	Conceptos	Atributos
	Comprobante de operaciones	Código nómina, número, descripción, fecha, asientos.
	Asientos	Cuenta, procesamiento nómina, pase
	Pases	Cuenta, moneda contable, importe original, bandera, tasa, importe base.
Precondiciones	Precondiciones	Pre-requisito
	Que se hayan confirmado las nóminas.	Confirmar nómina.
Descripción	El sistema una vez confirmada la nómina o ajustes automáticamente emite un comprobante de operaciones con los pases de las cuentas. Este comprobante se envía a contabilidad.	
Validaciones	Validar que estén confirmadas las nóminas. Si no esta confirmada no emite ningún comprobante hasta que se confirme la misma.	
Post-condiciones	Se ha emitido el comprobante a Contabilidad.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario ver [Anexo 8](#)

Especificación del requisito Detalles del comprobante de operaciones

Conceptos tratados	Conceptos	Atributos
	Comprobante de Operaciones.	Código nómina, número, descripción, fecha, asientos.
	Asientos	Cuenta, procesamiento nómina, pase
	Pases	Cuenta, moneda contable, importe original, bandera, tasa, importe base.
Precondiciones	Precondiciones	Pre-requisito
	Que se hayan procesado la nómina y este en estado de edición o confirmación.	Procesar Nómina.
Descripción	<p>Durante el procesamiento de las nóminas es necesario revisar las operaciones de los comprobantes de las mismas que no son más que los pases a las diferentes cuentas que se afectan.</p> <p>Seleccionar del listado de nóminas procesadas, la que el usuario desee para ver las operaciones del comprobante.</p> <p>El sistema muestra un formulario donde se visualiza las operaciones del comprobante que corresponde a la nómina seleccionada.</p>	
Validaciones	No procede.	
Post-condiciones	Se han mostrado los detalles del comprobante de operaciones	
Post-requisito	No procede.	

Prototipo de interfaz de usuario ver [Anexo 8](#)

2.6.9 Requisito funcional Cierre contable

Especificación del Requisito Efectuar cierre contable

Conceptos tratados	Conceptos	Atributos
Precondiciones	Precondiciones	Pre-requisito
	Que se hayan confirmado las nóminas y ajustes.	Revisar nómina.
Descripción	<p>Se selecciona la opción cierre contable, el sistema realiza un grupo de validaciones.</p> <p>Si estas validaciones son correctas se cierra el periodo contable.</p> <p>Si no se informa de los errores.</p>	
Validaciones	<p>Se realiza la conciliación del submayor de vacaciones.</p> <p>Se realiza la conciliación del submayor de retenciones.</p> <p>Se verifica que todas las nóminas estén confirmadas.</p> <p>Se verifica que los comprobantes de operaciones estén confirmados.</p>	
Post-condiciones	Se ha cerrado el periodo contable del subsistema de capital humano.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario ver [Anexo 9](#)

2.6.10 Requisito funcional Configuración

Especificación del requisito Establecer configuración de cuentas

Conceptos tratados	Conceptos	Atributos
	Configuración	Cuenta de nómina por pagar

		Cuenta de salario basico Cuentas de conatabilizacion adicional
Precondiciones	Precondiciones	Pre-requisito
	Tiene que haber cuentas definidas en la entidad.	Nomenclador de cuentas (Contabilidad)
Descripción	Al seleccionar configuración se muestra la interfaz que permite escoger las cuentas: Cuenta de nómina por pagar, Cuenta de salario basico y las 3 cuentas necesarias para la contabilizacion adicional. Al aceptar se guarda la configuración seleccionada.	
Validaciones	No procede	
Post-condiciones	No procede	
Post-requisito	No procede	

Prototipo de interfaz de usuario ver [Anexo 10](#)

2.6.11 Requisito funcional Gestionar frecuencia de pago

Especificación del requisito Adicionar frecuencia de pago

Conceptos tratados	Conceptos	Atributos
		Frecuencia de pago
Precondiciones	Precondiciones	Pre-requisito
	No procede	No procede
Descripción	Se selecciona la opción adicionar frecuencia de pago. El sistema permite introducir los datos de la frecuencia de pago: Denominación: nombre que el usuario le asigna al tipo de frecuencia. Cantidad de veces por mes: cantidad de veces que según este tipo de	

	frecuencia se le paga en un mes a los trabajadores. Se acepta para adicionar la frecuencia de pago.
Validaciones	Se validan los datos.
Post-condiciones	Se ha adicionado una frecuencia de pago.
Post-requisito	No procede

Prototipo de interfaz de usuario ver [Anexo 11](#)

Especificación del requisito Modificar frecuencia de pago

Conceptos tratados	Conceptos	Atributos
	Frecuencia de pago	Denominación, cantidad de veces por mes
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir al menos una frecuencia de pago.	Adicionar frecuencia de pago.
Descripción	<p>Se selecciona la opción modificar frecuencia de pago.</p> <p>El sistema permite modificar la denominación y la cantidad de veces por mes.</p> <p>Se acepta para efectuar la modificación frecuencia de pago.</p> <p>Mostrar un mensaje de confirmación.</p> <p>Si se confirma se guardan los cambios.</p>	
Validaciones	Se validan los datos	
Post-condiciones	Se ha modificado una frecuencia de pago.	
Post-requisito	No procede	

Prototipo de interfaz de usuario ver [Anexo 11](#)

Especificación del requisito Eliminar frecuencia de pago

Conceptos tratados	Conceptos	Atributos
	Frecuencia de pago	Denominación, cantidad de veces por mes.
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir al menos una frecuencia de pago.	Adicionar frecuencia de pago.
Descripción	El sistema permite seleccionar la frecuencia de pago a eliminar. Se selecciona la opción eliminar. El sistema muestra un mensaje de confirmación. Se acepta.	
Validaciones	No procede.	
Post-condiciones	Se ha eliminado una frecuencia de pago.	
Post-requisito	No procede.	

Especificación del requisito Mostrar frecuencia de pago

Conceptos tratados	Conceptos	Atributos
	Frecuencia de pago	Denominación, cantidad de veces por mes
Precondiciones	Precondiciones	Pre-requisito
	No procede	No procede

Descripción	Se selecciona frecuencia de pago. Se muestra en una tabla las frecuencias de pago.
Validaciones	No procede.
Post-condiciones	Se muestran los datos de las frecuencias de pago.
Post-requisito	No procede.

Prototipo de interfaz de usuario ver [Anexo 11](#)

2.6.12 Requisito funcional Reportes

Especificación del requisito Listado para el pago por tarjeta magnética

Objetivo	Listado para el pago por tarjeta magnética.
Parámetros	Código entidad, Área de trabajo
FORMATO	
Encabezado	Reportes de Listado para el pago por tarjeta magnética
Encabezado de página	Nombre de la Entidad: _____ Código:_____ Fecha: _____
Pie de página	Realizado por: _____ Aprobado por: _____
Resumen	No procede.
Origen de los datos	Línea Estructura y composición: nombre de la entidad Área de trabajo
Ejemplo de formato	Ver Anexo 12
ELEMENTOS A MOSTRAR	

Especificación del requisito Listado para el pago tradicional

Objetivo	Con este reporte se obtiene el resultado del proceso del cálculo de la nomina, con el objetivo de que sea revisado por el encargado de nóminas.
Parámetros	Código entidad, Área de Trabajo
FORMATO	
Encabezado	Reportes de Listado para el pago tradicional
Encabezado de página	Nombre de la Entidad: _____ Periodo de la nomina: _____ Tipo de nomina: _____
Pie de página	Realizado por: _____ Aprobado por: _____
Resumen	No procede.
Origen de los datos	No procede.
Ejemplo de formato	Ver Anexo12
ELEMENTOS A MOSTRAR	

Especificación del requisito Distribución de moneda

Objetivo	Mostrar la cantidad de billetes y monedas a solicitar al banco para realizar el pago por caja a los trabajadores. Tomando el importe de cada trabajador se halla la cantidad de billetes y monedas que se necesita para pagarle a este trabajador y se hace un resumen de todos los trabajadores.
Parámetros	No procede
FORMATO	
Encabezado	DISTRIBUCIÓN DE MONEDA – NÓMINA DE SALARIO
Encabezado de página	Período: Mes/Año Fecha: DD/MM/AAAA
Pie de página	Realizado por: _____ Aprobado por: _____
Resumen	Por cada trabajador calcular la cantidad de billetes y monedas que hace falta para pagarle al trabajador. Y se suma para hallar la cantidad total.
Origen de los datos	No procede.

Ejemplo de formato	Ver Anexo 12
ELEMENTOS A MOSTRAR	

Especificación del requisito Comprobante de operaciones

Objetivo	Con este reporte se obtiene la visualización del comprobante de operaciones dada una nómina permitiendo que se imprima y se guarde.
Parámetros	Código entidad, nómina
FORMATO	
Encabezado	Comprobante de operaciones
Encabezado de página	Nombre de la entidad Tipo de la entidad (presupuestada ó empresarial) Periodo de la nómina Tipo de nómina Denominación de la nómina
Pie de página	Realizado por: _____ Aprobado por: _____
Resumen	No procede
Origen de los datos	Nombre de la entidad: id_entidad buscar en estructura y composición Tipo de entidad: id_entidad buscar en estructura y composición Periodo de la nómina: dado id_periodo de la nómina Nom_periodo Tipo de nómina: dado el id_tioptomina de la nomina nom_tioptomina. Denominación de la nómina: dado el id_nomina dat_nomina Estado: estado del comprobante de operaciones. Cuenta: cuenta de dat_asientos. Subcuenta: cuenta del dat_pase.

	Débito o crédito: es el importe de dat_pase que depende del campo bandera si es 0 es débito y si es 1 es crédito.
Ejemplo de formato	Ver Anexo 12
ELEMENTOS A MOSTRAR	

2.7 Conclusiones parciales

En el presente capítulo una vez identificados los procesos de negocio se realizó el modelo conceptual con el objetivo identificar las entidades del negocio. Se especificaron las técnicas y patrones utilizados en la captura de requisitos y se identificaron y describieron los requisitos funcionales satisfaciendo las necesidades reales de los usuarios y clientes.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

En el presente capítulo se muestra una representación grafica de diagramas del diseño correspondiente al sistema, entre los que están el diagrama de clase del diseño y el diagrama de componente, realizándose una breve descripción de los mismos. Especificándose además los patrones de diseño utilizados.

El diseño del software es un proceso iterativo mediante el cual los requisitos se traducen en un <<plano>> para construir el software. (8) El diseño permitirá modelar el sistema de forma tal que soporte todos sus requisitos.

3.2 Patrón de arquitectura

Los patrones de arquitectura están relacionados a la interacción de objetos dentro o entre niveles arquitectónicos. Resuelven problemas de adaptabilidad a requerimientos cambiantes, performance, modularidad y acoplamiento. Dando solución a llamadas entre objetos (similar a los patrones de diseño), decisiones y criterios arquitectónicos, empaquetado de funcionalidades.

3.2.1 Modelo vista controlador (MVC)

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

El 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.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

-
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo.

El Controlador es responsable de:

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

Las Vistas son responsables de:

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

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

3.3 Patrones de diseño

Los patrones de diseño nos ayudan a evitar que los cambios en el sistema se realicen de una forma específica asegurándonos de que se afecte lo menos posible. Cada patrón de diseño deja a cierto aspecto del sistema que varíe de forma independiente de otros aspectos, haciendo al sistema robusto a un tipo particular de cambio.

3.3.1 Patrones Grasp (Patrones generales de software para asignar responsabilidades)

Los patrones Grasp describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Booch y Rumbaugh definen la responsabilidad como "un contrato u obligación de un tipo o clase" (6)

Existen nueve patrones Grasp los cuales son: Experto, Creador, Alta Cohesión, Bajo Acoplamiento,

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Controlador, Polimorfismo, Fabricación Pura, Induración y No Hables con Extraños. De estos utilizaron Bajo Acoplamiento y Alta Cohesión con el fin de que contribuya a que el sistema sea más robusto y flexible.

Bajo Acoplamiento

Este patrón es un principio que asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades. El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema.

Consiste en tener las clases lo menos ligadas entre sí que se pueda, de tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

Mantener el bajo acoplamiento entre clases más que un patrón en una buena práctica del diseño dado que al realizar cambios no se afectan otros componentes, es más fácil de entender de manera aislada y de esta forma se puede reutilizar mejor las clases.

Alta Cohesión

En la perspectiva del diseño orientado a objetos, la cohesión (ó más exactamente, la cohesión funcional) es una medida de cuan relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. (7)

Este patrón nos dice que la información que almacena una clase debe de ser coherente y está en la mayor medida de lo posible relacionada con la clase.

El patrón Alta Cohesión -como tantas otras cosas en la tecnología de objeto- presenta semejanzas con el mundo real. Cuando alguien asume demasiadas responsabilidades -sobre todo las que debería delegar-, no será eficiente. Realizando un diseño donde las clases mantengan una alta cohesión se mejora la claridad y facilidad con que se entiende el diseño, se simplifica el mantenimiento y las

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

mejoras de funcionalidad, a menudo se genera un bajo acoplamiento, soporta mayor capacidad de reutilización.

3.3.2 Patrones Gof (Gang of Four)

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

Los patrones creacionales se encargan de la creación de los objetos ayudando a que el sistema sea independiente de la creación, composición y representación de los objetos.

Los patrones estructurales son los encargados de cómo las clases y objetos están compuestos para formar estructuras mas grandes. Los patrones estructurales usan la herencia para componer interfaces u objetos en tiempo de ejecución.

Los patrones comportamiento plantean algoritmos y la asignación de responsabilidades entre objetos. Estos patrones no solo describen clases y objetos sino también describen la comunicación entre ellos. Se seleccionaron los siguientes patrones Gof en la modelación del diseño:

3.3.2.1 Patrones estructurales

Composición

Este patrón permite Componer objetos en estructuras de árbol para representar jerarquías parte-todo. El patrón composición permite a los clientes darle a tratamiento a cada uno de los objetos y composiciones de objetos de manera uniforme.

La clave del patrón composición es una clase abstracta que representa a la vez la primitiva y sus contenedores. Para el sistema gráfico, esta clase es Gráfica. Gráfica tiene métodos como Dibujar, operación específica de los objetos gráficos. También declara que todas las operaciones de los objetos compuestos comparten, tales como las operaciones de acceso y gestión de sus hijos.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Este patrón se utiliza cuando se quiera representar jerarquías parte-todo de los objetos. También se desea que los clientes puedan ignorar la diferencia entre las composiciones de objetos y personas. Tratar a todos los clientes de los objetos en la estructura compuesta de manera uniforme.

Este se aplica en el procesamiento de las retenciones en el diagrama de clase administrar nómina cuando se tiene por tipo de retención un valor diferente en cada procesamiento. Una retención puede ser simple como en su estado inicial cuando es definida en el nomenclador o compuesta después que se le realiza el procesamiento donde se le agregan nuevos datos.

3.3.2.2 Patrones de comportamiento

Cadena de responsabilidad

La cadena de responsabilidad se encarga de evitar el acoplamiento del remitente de una petición a su receptor, dando más de un objeto la posibilidad de manejar la petición. La cadena recibe los objetos y pasa la solicitud a lo largo de la cadena hasta el objeto que la gestiona.

La idea de este patrón es dissociar los remitentes y receptores, dando múltiples objetos de una oportunidad para la tramitación de una solicitud. La petición se transmite a lo largo de una cadena de objetos hasta que uno de ellos se ocupa de ella. Este patrón se debe utilizar cuando se requiere que más de un objeto puede manejar una petición, y el manejador no se conoce a priori. Se desea remitir una petición a uno de varios objetos, sin especificar el receptor explícitamente.

Aplicar la cadena de responsabilidad trae beneficios como la reducción de acoplamiento, mayor flexibilidad en la asignación de responsabilidades a los objetos con el inconveniente de que como recepción no está garantizada pueden existir excepciones que nunca se manejen si la cadena está mal configurada.

Este es utilizado en la mayoría de los diagramas de clases siendo aplicado en el tratamiento de excepciones. Un ejemplo de su uso es cuando se produce un error al insertar en la base de datos, el cual es captado por las capas superiores, reenviando la excepción hasta la capa de aplicación donde traduce al lenguaje del usuario.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Estado

Este patrón permite a un objeto modificar su comportamiento cuando su estado interno cambia. El objeto aparecerá para cambiar su clase. El patrón Estado se debe utilizar cuando el comportamiento de un objeto depende de su estado, y debe cambiar su comportamiento en tiempo de ejecución en función de ese estado.

En operaciones grandes, de varias declaraciones condicionales que dependen sobre estado del objeto. Este estado suele ser representado por uno o más enunciados constantes. A menudo, varias operaciones de esta misma contendrán una estructura condicional. El patrón Estado pone cada rama de la condicional en una clase separada. Esto le permite tratar el estado del objeto como un objeto en su por derecho propio, que puede variar independientemente de otros objetos.

El uso de este patrón trae consigo que se localiza el estado específico de comportamiento y conducta para las particiones de los diferentes estados. Las transiciones de estado se hacen más explícitas y los objetos de estado pueden ser compartidos.

Este es aplicado en las validaciones realizadas utilizando la clase Validator, donde se llama un grupo de funciones según sea la validación que se quiere realizar teniéndose en tiempo de ejecución la construcción de un nuevo conjunto de operaciones.

Estrategia

Define una familia de algoritmos, encapsulando cada uno y los hace intercambiables. Estrategia permite al algoritmo variar independientemente de los clientes que la utilizan.

El patrón de estrategia debe utilizarse cuando muchas de las clases relacionados se diferencian sólo en su comportamiento. Estrategia proporciona una manera de configurar una clase con uno de los muchos comportamientos. Cuando se necesitan diferentes variantes de un algoritmo. Por ejemplo, puede definir algoritmos que reflejan los distintos espacio / tiempo de los compromisos. Estrategia puede utilizarse cuando se apliquen estas variantes como una jerarquía de clase de los algoritmos.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Un algoritmo que utiliza datos que los clientes no deben conocer. Usar estrategia para evitar la exposición compleja, algoritmo específico de estructura de datos.

Cuando una clase define muchos comportamientos, y éstos aparecen como múltiples condicionales las declaraciones en sus operaciones. En lugar de muchos condicionales, mueve las relaciones condicionales en su propia estrategia de clase. El patrón Estrategia tiene las siguientes ventajas como que da una familia de algoritmos relacionados, presenta una alternativa de subclasses, elimina las condicionales, proporciona diferentes implementaciones de la misma conducta.

Este patrón se evidencia en los diferentes diagramas donde la clase controladora utiliza métodos definidos en las clases del modelo según las operaciones a realizar.

3.4 Diagramas de clases del diseño

Un diagrama de clase del diseño es un diagrama estático que describe la estructura de un sistema mostrando sus clases, operaciones y las relaciones existentes entre ellos.

Contienen la siguiente información:

- Clases asociadas.
- Métodos
- Navegabilidad
- Dependencias

Tipos de clases

- Controladoras: Las clases controladoras o controller permiten la gestión de información entre la vista y modelo, en respuesta a acciones del usuario.
- Modelo:
- Negocio: Las clases del negocio definen la lógica del negocio de la aplicación.
- Dominio: Las clases del dominio son las responsables de acceder mediante consultas a la base de datos.
- Vista: Se encarga de mostrar la información recibida del modelo al usuario.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

A continuación se muestran los diagramas de clases del diseño; las descripciones de las clases se pueden ver en el [Anexo 22](#).

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

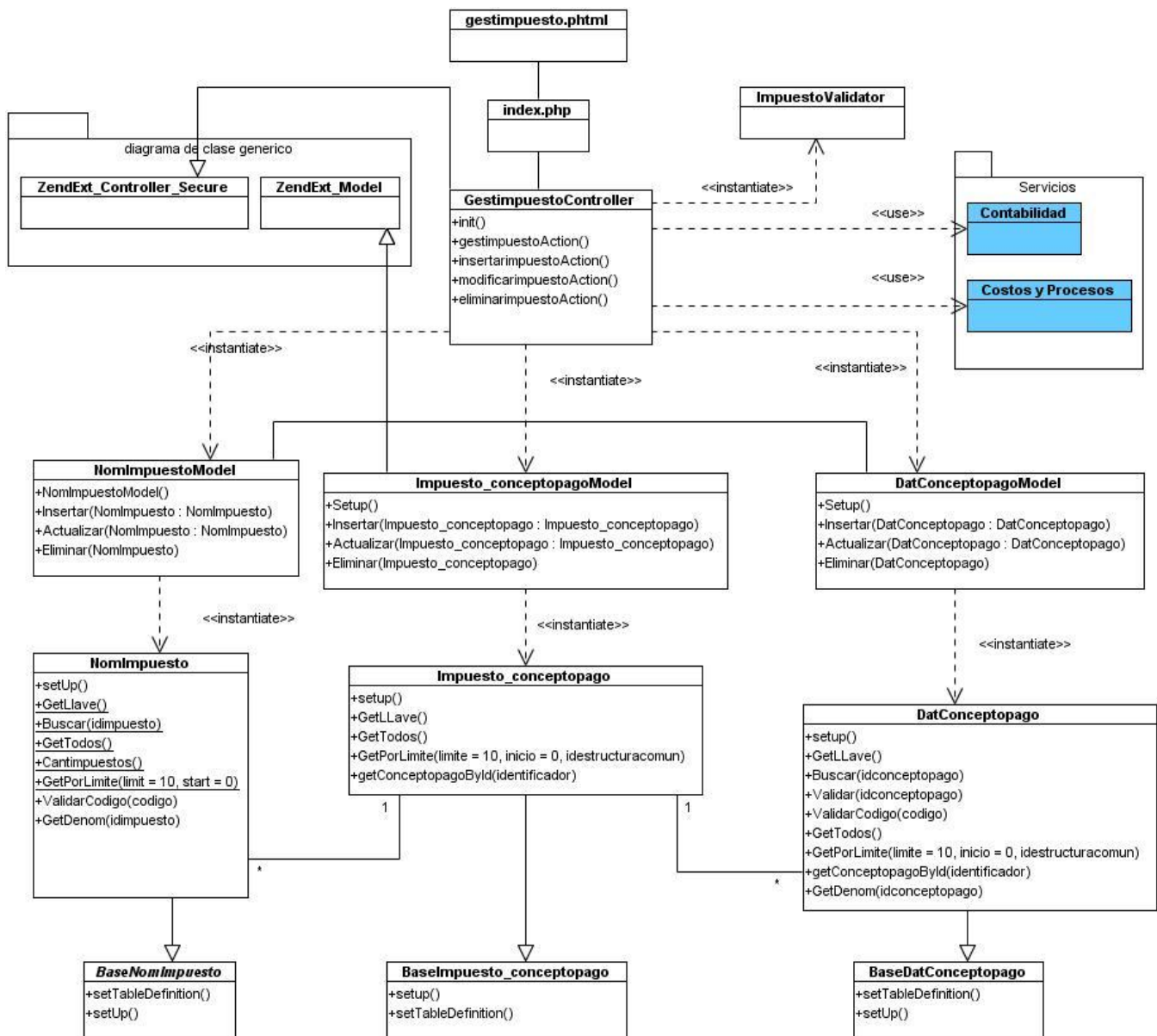


Figura 5 Diagrama de clase Gestionar impuesto y contribuciones.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

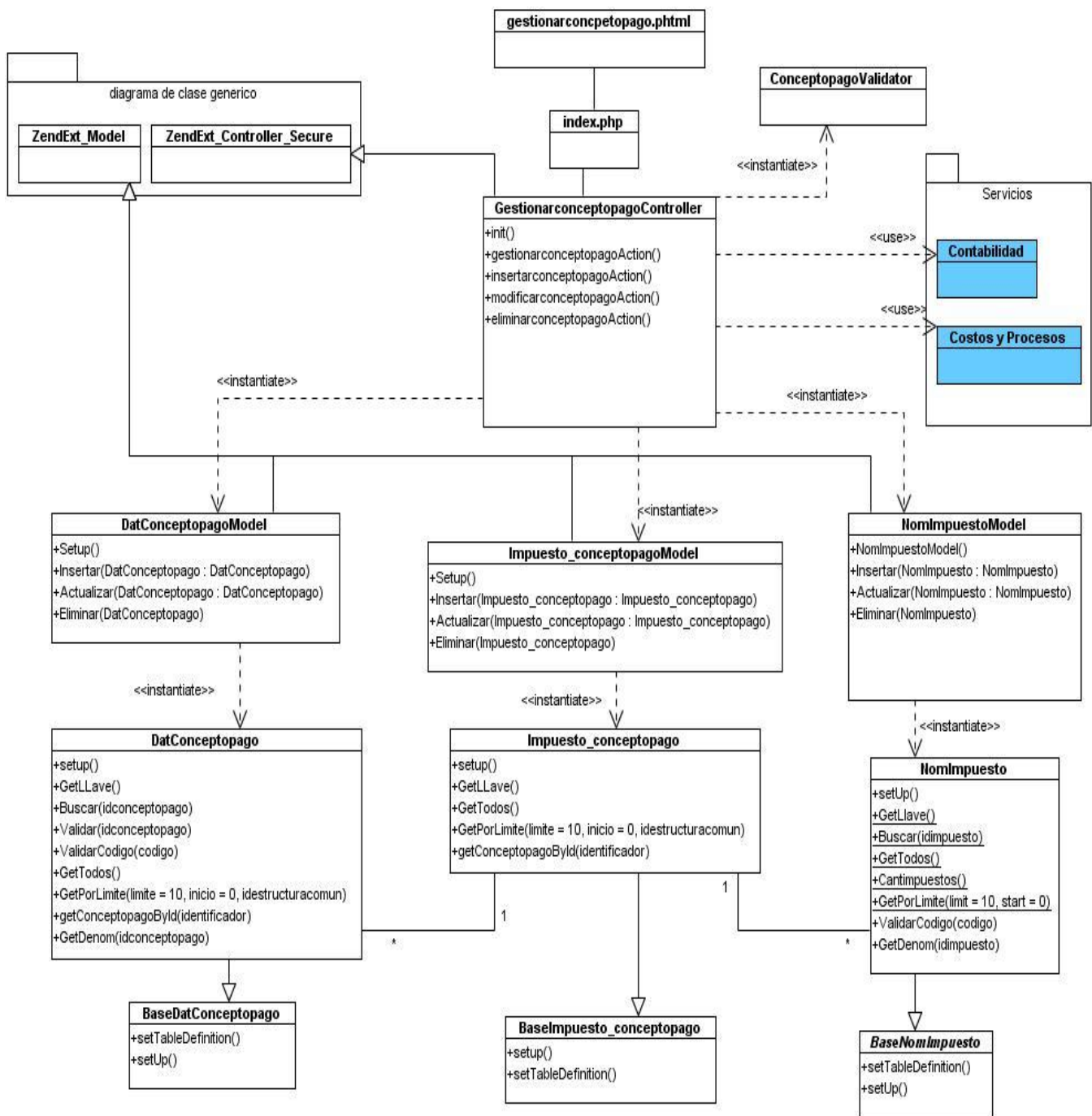


Figura 6 Diagrama de clase Gestionar concepto de pago.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

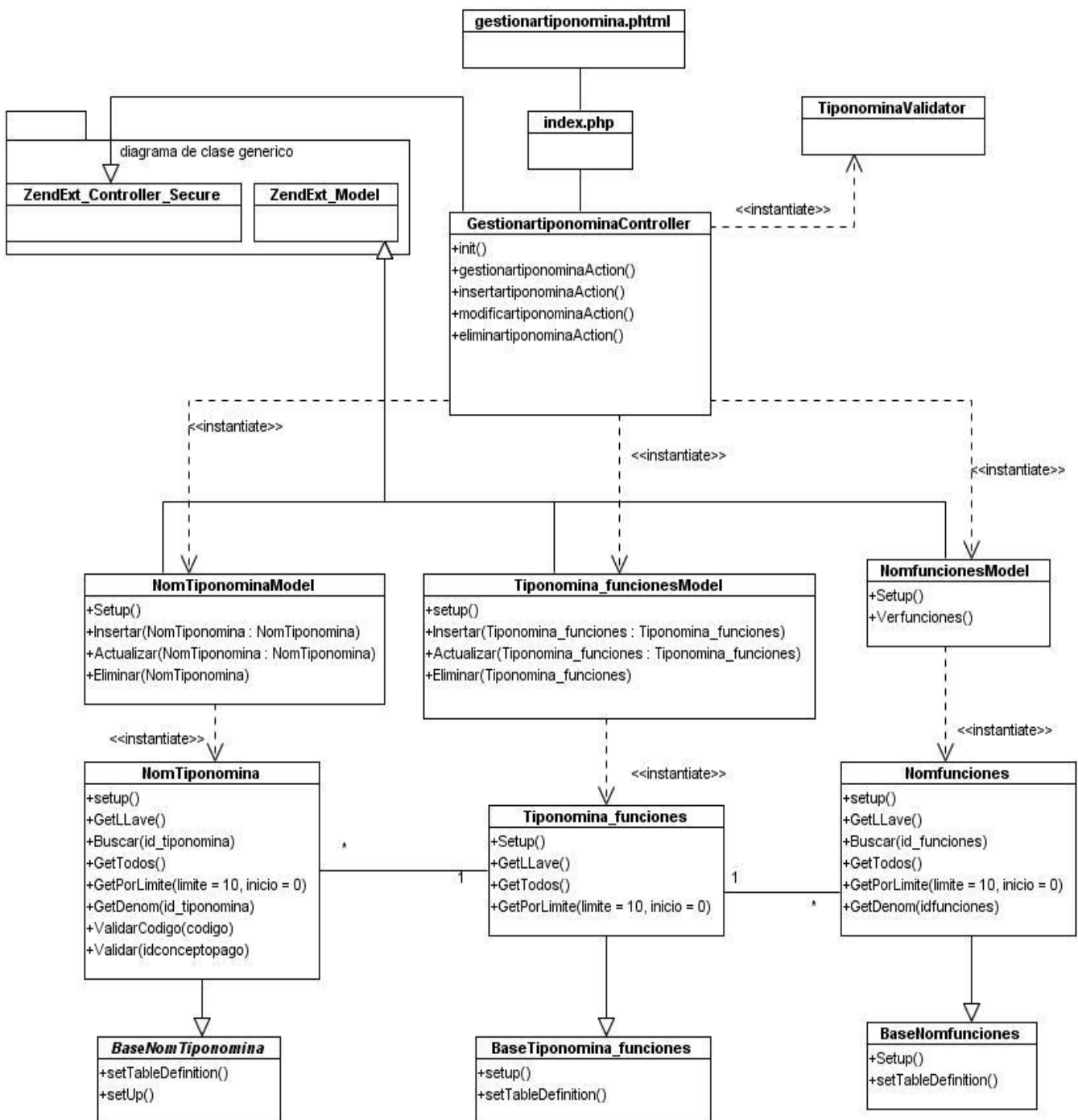


Figura 7 Diagrama de clase Gestionar tipo de nómina.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

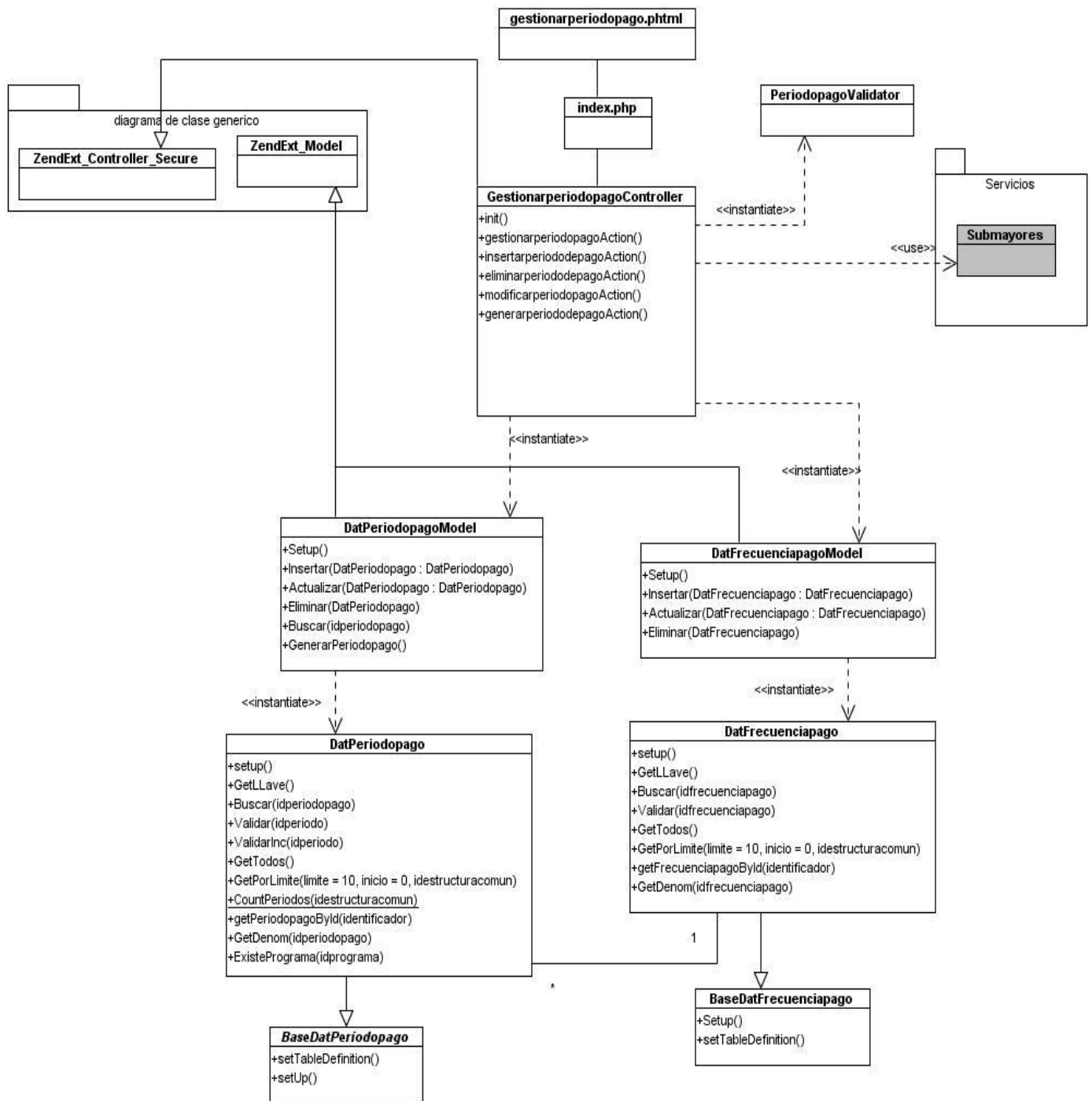


Figura 8 Diagrama de clase Gestionar periodo de pago.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

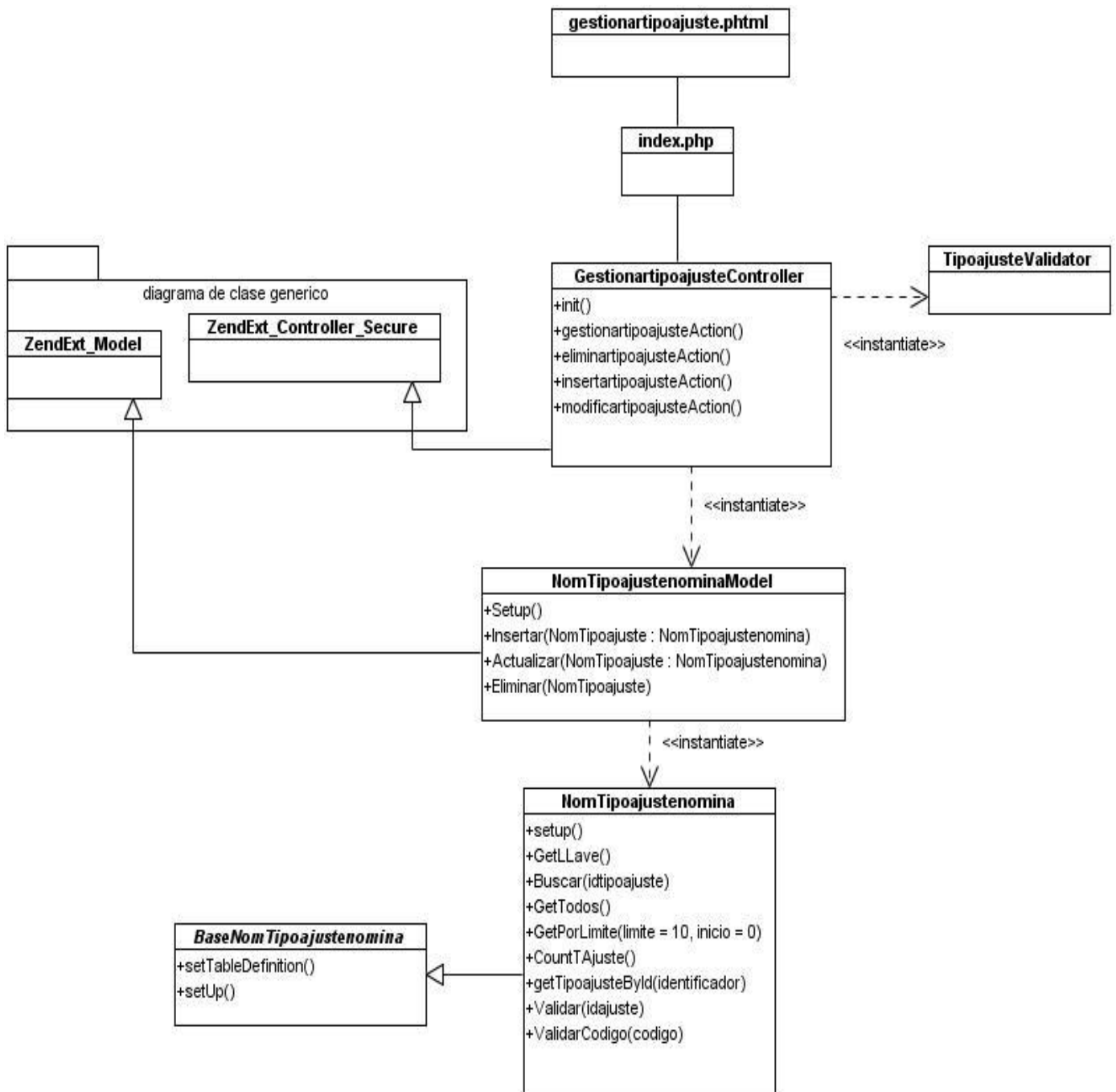


Figura 9 Diagrama de clase Gestionar tipos de ajustes.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

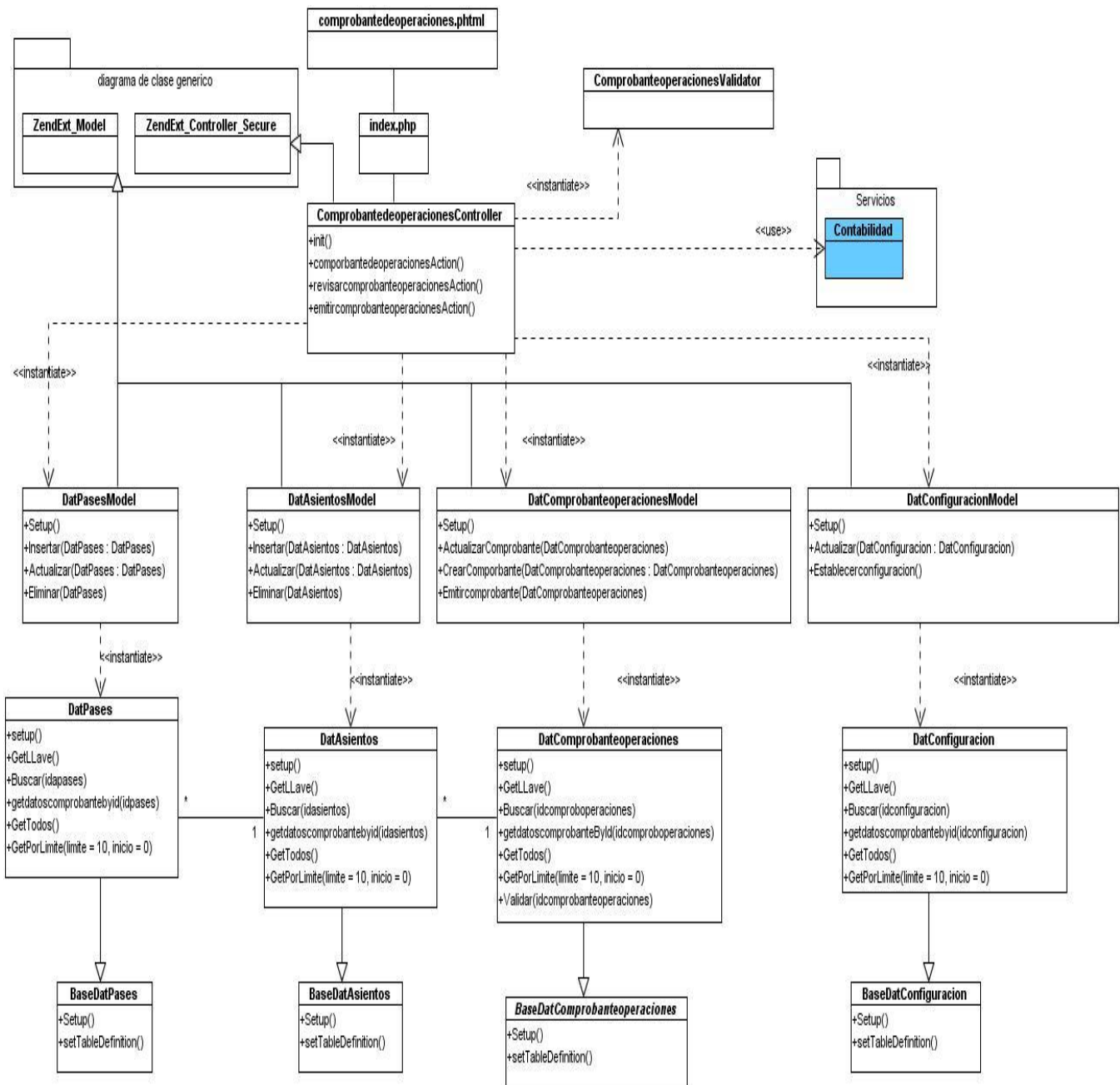


Figura 11 Diagrama de clase de Comprobante de operaciones

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

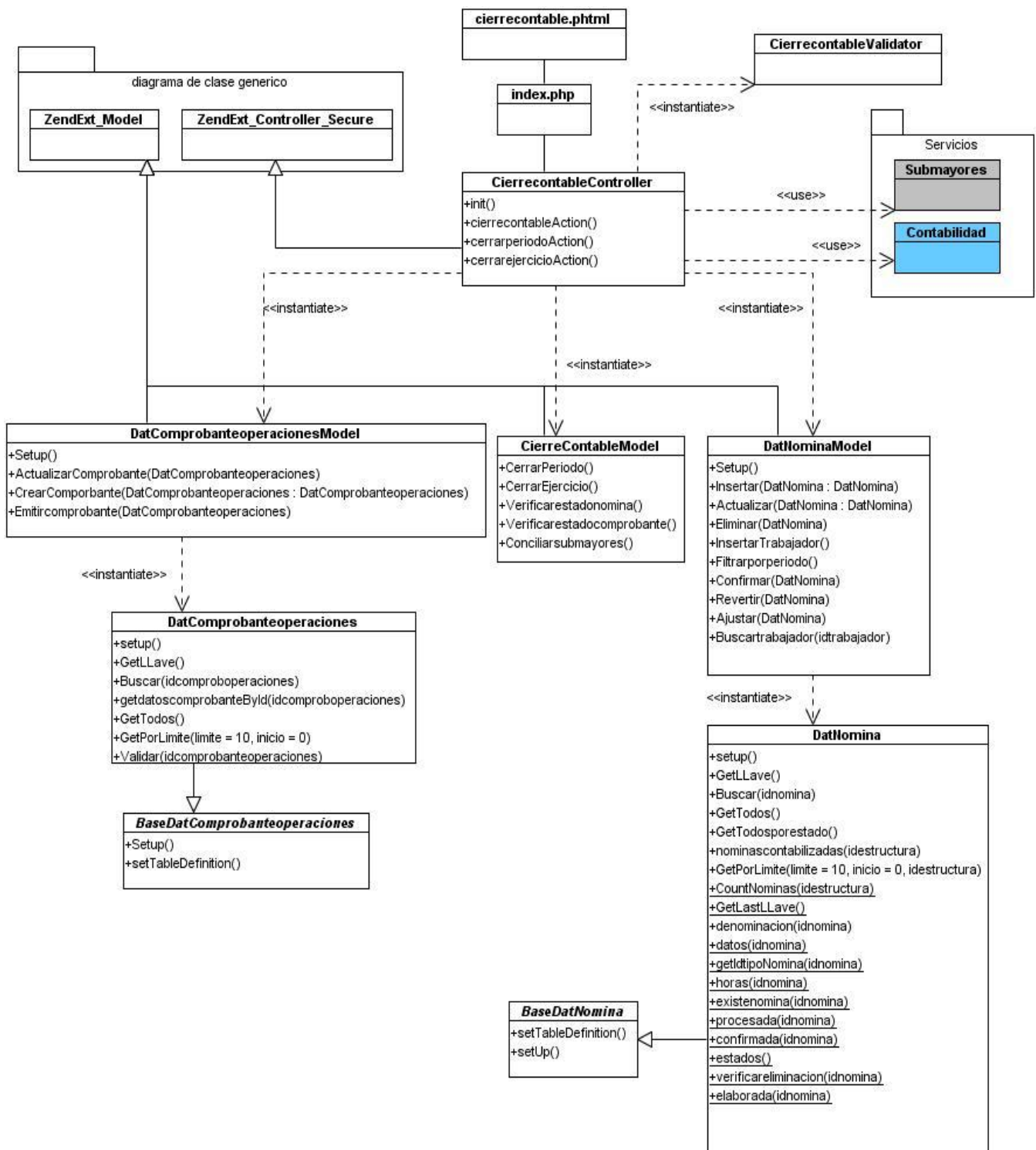


Figura 12 Diagrama de clase de Cierre.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

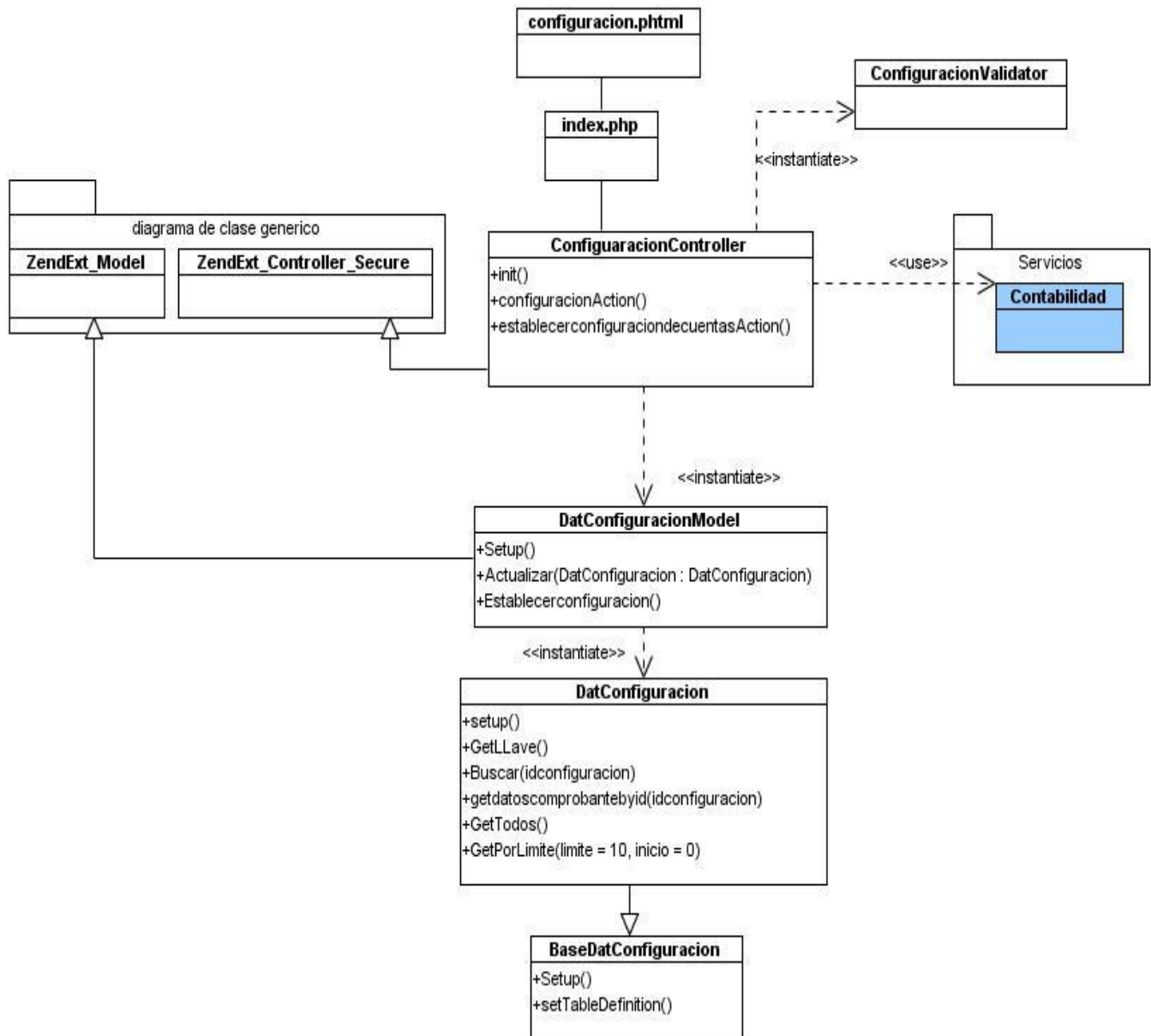


Figura 13 Diagrama de clase de Configuración

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

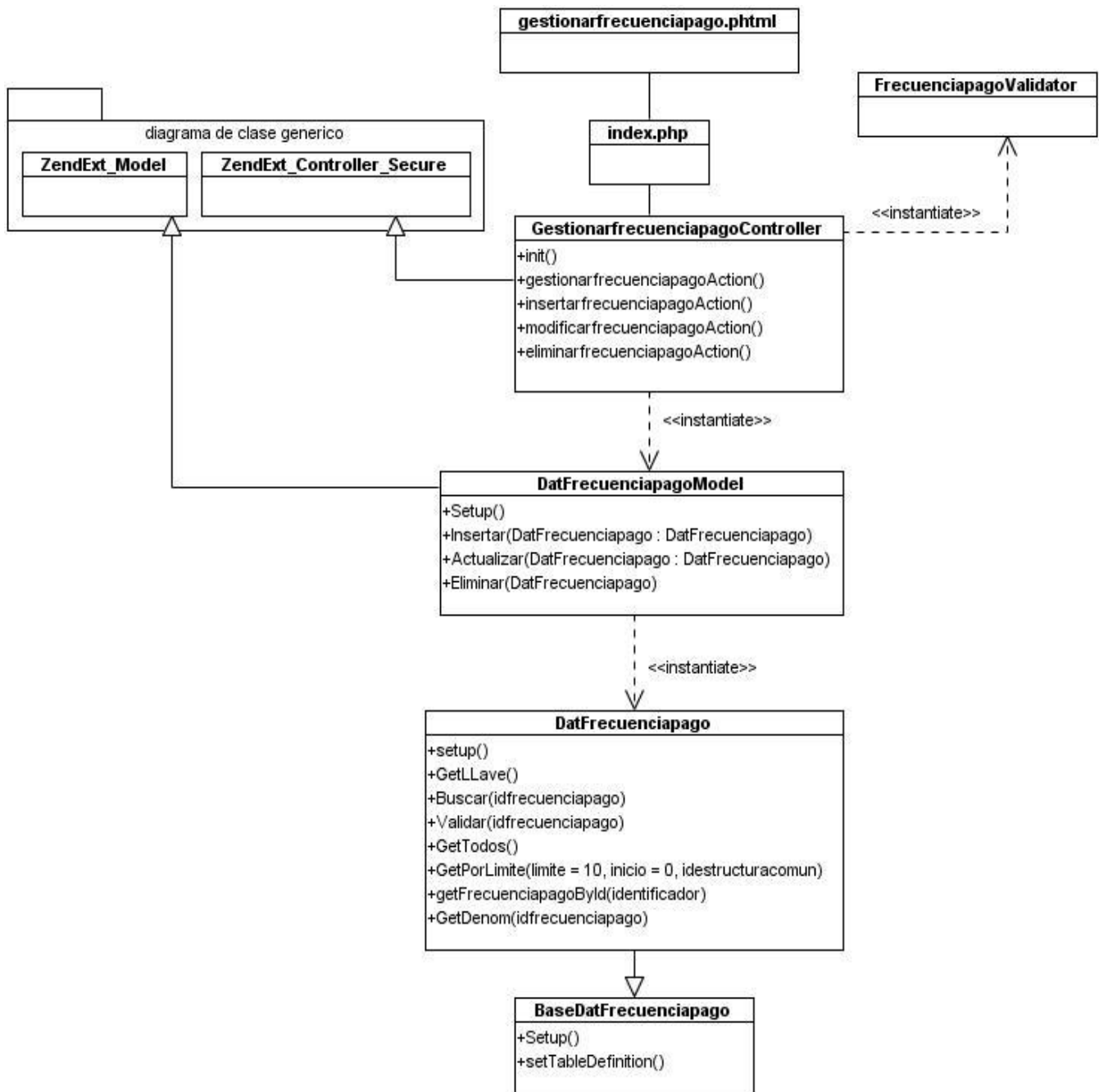


Figura 14 Diagrama de clase Gestionar frecuencia de pago.

3.5 Diagrama de componente

Un diagrama de componentes representa gráficamente cómo un sistema de software es dividido en componentes mostrando las dependencias existentes entre estos. El diagrama contiene componentes, interfaces, relaciones entre ellos y puede contener paquetes utilizados para agrupar elementos del modelo; mostrando las dependencias lógicas entre componentes software.

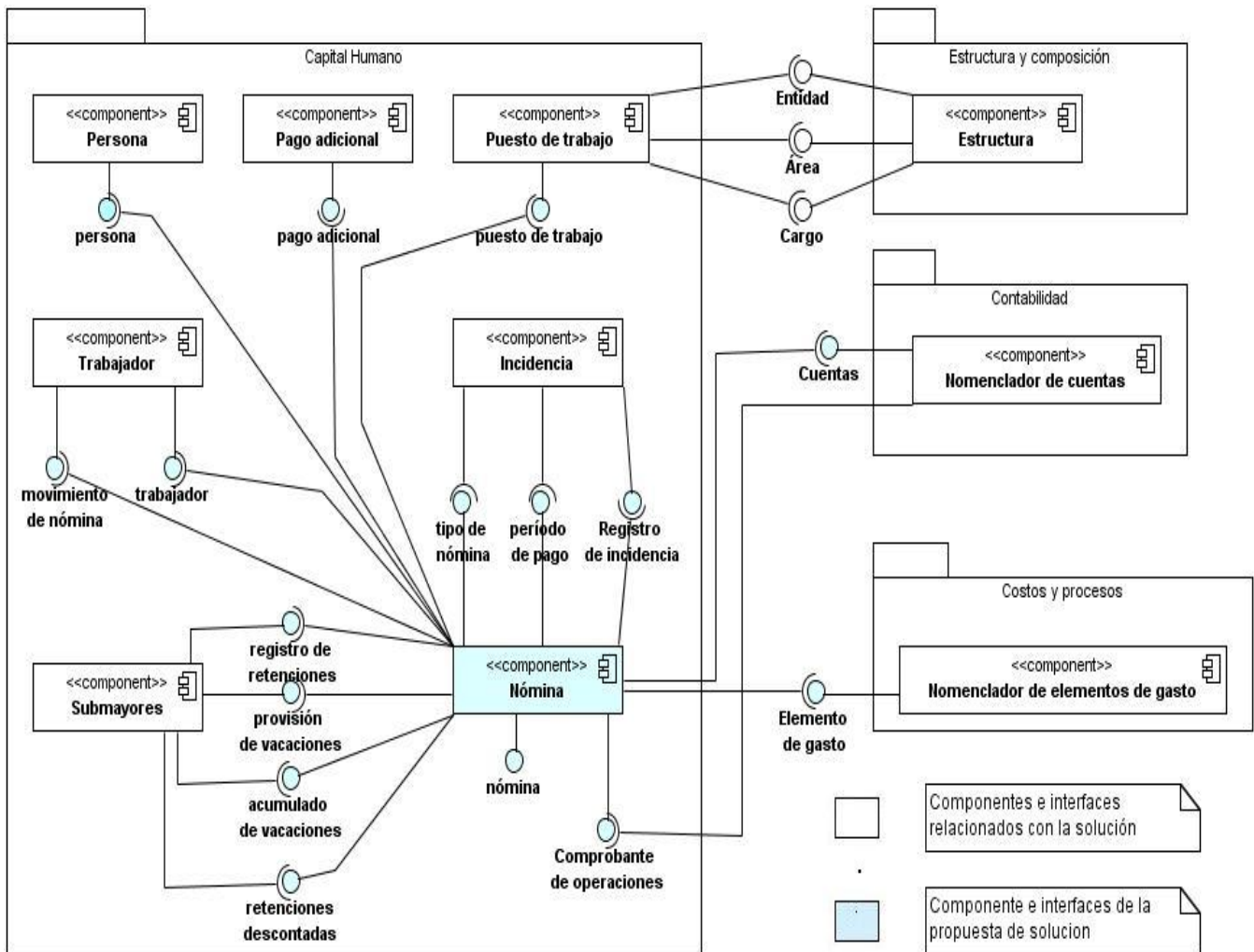


Figura 15 Diagrama de componente

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

En el diagrama de componente anterior se puede observar las dependencias que posee el componente Nómina con el resto de los representados, las cuales se describen brevemente a continuación.

- El componente Nómina tiene relación de dependencia con el componente Persona utilizando su interfaz que contiene los datos personales de los trabajadores.
- Se relaciona con el componente Pago adicional utilizando los datos necesarios para realizar el cálculo de los pagos adicionales de cada trabajador.
- Se relaciona con el componente Puesto de trabajo para obtener los pagos adicionales asociados a un puesto, el área, cargo y el salario escala.
- Se relaciona con el componente Trabajador mediante la interfaz movimiento de nómina para obtener el listado actualizado de los trabajadores en la entidad y con la interfaz trabajador para obtener los datos del personal.
- Se relaciona con el componente Incidencia proporcionándole los tipos de nómina para que este a su vez pueda definir los tipos de incidencias y los períodos de pagos con el objetivo que se realice el registro de las incidencias de los trabajadores, este registro utilizado posteriormente para el cálculo de la nómina.
- Se relaciona con el componente Submayores utilizando el registro de retenciones para realizar el descuento de las retenciones durante el procesamiento de la nómina; si la nómina es de vacaciones utiliza la interfaz provisión de vacaciones para realizar el cálculo de las vacaciones de los trabajadores. Luego de realizado el procesamiento actualiza el acumulado de las vacaciones y las retenciones descontadas por cada trabajador.
- Se relaciona con el componente Nomenclador de cuentas del subsistema Contabilidad mediante la interfaz cuentas para realizar el comprobante de operaciones, que luego de terminado se envía a este subsistema.
- Se relaciona con el componente Nomenclador elemento de gasto del subsistema Costos y procesos para contabilizar los conceptos de pagos y la realización del comprobante de operaciones.

3.6 Conclusiones parciales

En el capítulo se realizó la modelación del diseño representándose gráficamente los diagramas de clases que describen la estructura del sistema; los patrones aplicados en el diseño y el diagrama de componentes en donde se representaron las dependencias lógicas que tiene el componente nómina con otros componentes y subsistemas. Con el diseño propuesto se sientan las bases para la fase de implementación del componente nómina.

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

CAPÍTULO 4: VALIDACION DE LOS RESULTADOS

4.1 Introducción

En el presente capítulo se realizara una evaluación de los requisitos capturados, mediante la aplicación de métricas que permitirán validar que los requisitos se hayan descrito correctamente. Además de técnicas como el prototipado del sistema que también ayudaran a validar los requerimientos.

4.2 Criterios para la evaluación y aceptación de los requisitos

Para realizar la validación de los requisitos existe toda una lista de características que sugieren el uso de una o más métricas como son: correctitud, completitud, consistencia y muchas otras, pero solo se van a aplicar las métricas antes mencionadas.

Correctitud

La métrica se propone determinar si la especificación de requisitos contiene todos los requisitos necesarios para satisfacer las necesidades del negocio y los interesados. Para aplicarla se debe mantener un registro de las observaciones del negocio realizadas.

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

$$X = D/T$$

X – Correctitud.

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

T – Total de requisitos revisados.

$$X = D/T$$

$$X = 4/48$$

$$X = 0,08$$

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

Compleitud

La métrica pretende determinar si la especificación de requisitos es completa. Para aplicarla se debe mantener un registro de las observaciones de formato realizadas.

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

$$X = 1 - O/S$$

X – Compleitud

O – Total de observaciones de formato.

S – Total de secciones del documento.

$$X = 1 - O/S$$

$$X = 1 - O/6$$

$$X = 1$$

Consistencia

La métrica se propone determinar si la especificación de requisitos es consistente. Para aplicarla se debe mantener un registro de las observaciones de consistencia realizadas.

Observaciones de consistencia: Son las observaciones realizadas por contradicciones existentes entre los requisitos.

$$X = C/T$$

X – Consistencia

C – Total de observaciones de consistencia.

T – Total de requisitos revisados.

$$X = C/T$$

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

$$X = 1/48$$

$$X = 0,02$$

La utilización de las métricas anteriores arrojaron los siguientes resultados:

En la correctitud se obtuvo un valor de 0,08 y para aceptar la especificación de requisitos el valor debe de ser menor que 0,10 lo cual se cumple.

En la completitud se obtuvo un valor de 1 y para aceptar la especificación de requisitos el valor debe de ser mayor que 0,90 lo cual se cumple.

En la consistencia se obtuvo un valor de 0,02 y para aceptar la especificación de requisitos el valor debe de ser menor que 0,20 lo cual se cumple.

Con los resultados obtenidos se puede concluir que la especificación de los requisitos es correcta por lo que quedan validados los requerimientos del sistema.

4.3 Técnicas de validación de requisitos

La validación de los requisitos se realiza con el fin de examinarlos para asegurarse de que definen el sistema adecuado. El mismo es de gran importancia puesto que permite detectar los errores de forma temprana, con el fin de no conducir a resultados inesperados evitando provocar gastos excesivos y grandes pérdidas de tiempo.

Entre las técnicas para validar requisitos se encuentran:

- Prototipo de Interfaz de usuario
- Diseño de Casos de Prueba
- Lista de Chequeo

De estas técnicas solo aplicaron las dos primeras.

4.3.1 Técnica de Prototipado

Un prototipo de interfaz de usuario es una versión inicial de un sistema de software que se utiliza con el fin de demostrar los conceptos, probar las opciones de diseño y entender mejor el problema y su solución. Esta técnica permite que se pueda corregir la aparición de errores cuando aun el software se

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

encuentra en su fase inicial, pudiendo evitar gastos innecesarios y pérdidas en el tiempo. Además de ser una buena forma de interacción con el cliente, donde este pueda apreciar una primera muestra del producto, y dar ideas o recomendaciones para que el futuro software cumpla con sus necesidades.

Para la elaboración y construcción del prototipo de interfaz gráfica se hizo uso de la herramienta Visual Paradigm for UML 6.4 Enterprise Edition. La generación de los prototipos es una forma de validar los requerimientos funcionales que fueron capturados durante la etapa de Requerimientos. La principal ventaja que presentan los prototipos de interfaces de usuarios es la de reflejar la presentación e interacción de las necesidades del usuario final en un entorno amigable y fácil de entender.

Ver Anexos

4.3.2 Técnica de Diseño de Casos de Prueba

El propósito de un Caso de Prueba es especificar una forma de probar el sistema, incluyendo las entradas con las que se ha de probar, los resultados esperados y las condiciones bajo las que ha de probarse. Estos son un producto de desarrollo del software que ayuda a validar y verificar las expectativas de los clientes.

Los requerimientos son la fuente principal para obtener los Casos de Prueba pero no son el único medio. Y muchas veces son insuficientes para proporcionar una base completa que permita desarrollar las pruebas. Por lo que es necesario considerar otros elementos como riesgos, restricciones, tecnologías, cambios, fallos, etc. Con estos se validan los requerimientos funcionales del sistema.

Los Casos de Prueba incluyen la verificación del resultado de la interacción entre los actores y el sistema, que satisfacen las precondiciones, postcondiciones y la secuencia de acciones especificadas por el requisito. Estos Casos de Prueba especifican típicamente una prueba del sistema como “caja negra”, es decir, una prueba del comportamiento observable externamente del sistema. Los datos de prueba se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien.

A continuación se presentan algunos casos de prueba del Administrar nómina; para ver el resto de los casos de prueba de los otros requisitos del componente ver desde el [Anexo 13](#) hasta el [Anexo 21](#).

4.3.2.1 Diseño de caso de prueba del requisito Crear nómina.

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Administrar nómina.
- Se debe seleccionar la opción Administrar nómina.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Crear nómina.	Se crea una nómina.	EP 1.1: Crear nómina introduciendo los datos correctamente.	<ul style="list-style-type: none">– Se debe presionar el botón Adicionar.– Se debe introducir los datos de la nómina en el formulario.– Se debe presionar el botón Aceptar.

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

		<p>EP 1.2: Crear nómina dejando campos requeridos en blanco.</p>	<ul style="list-style-type: none">– Se debe presionar el botón Adicionar.– Se debe introducir los datos de la nómina en el formulario dejando al menos un campo requerido en blanco.– Se debe presionar el botón Aceptar.
		<p>EP 1.3: Crear nómina introduciendo errores en los datos.</p>	<ul style="list-style-type: none">– Se debe presionar el botón Adicionar.– Se debe introducir los datos de la nómina en el formulario introduciendo algunos datos inválidos.– Se debe presionar el botón Aceptar.
		<p>EP 1.4: Cancelar operación.</p>	<ul style="list-style-type: none">– Se debe presionar el botón Adicionar.– Se debe introducir los datos de la nómina o no en el formulario.– Se debe presionar el botón Cancelar.

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Denominación.	Campo de texto.	NO	Combinación de letras y números.
2	Tipo nómina.	Lista desplegable.	NO	Selección.
3	Período pago.	Lista desplegable.	NO	Selección.
4	Fecha.	Campo de fecha.	NO	Selección.
5	Código.	Campo de texto.	NO	Combinación de letras y números.

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Denominación	Tipo nómina	Período pago	Fecha	Código	Respuesta del sistema	Resultado de la prueba
EP 1.1	Crear nómina introduciendo los datos correctamente.	V(Enero)	V(Subsidio)	V(Enero)	V(12/12/2008)	V(1a)	Muestra un mensaje: "La nómina se creo satisfactoriamente."	
EP	Crear	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	Se muestra un	

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

1.2	nómina dejando campos requeridos en blanco.)		mensaje de error: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)".
		I(Vacío)	V(Subsidio)	V(Enero)	V(12/1 2/2008)	V(1a)	
		V(Enero)	I(Vacío)	V(Enero)	V(12/1 2/2008)	V(1a)	
		V(Enero)	V(Subsidio)	I(Vacío)	V(12/1 2/2008)	V(1a)	
		V(Enero)	V(Subsidio)	V(Enero)	I(Vacío)	V(1a)	
		V(Enero)	V(Subsidio)	V(Enero)	V(12/1 2/2008)	I(Vacío)	
EP 1.3	Crear nómina introduciendo errores en los datos.	I(E#\$\$)	V(Vacaciones)	V(Febrero)	V(12/0 5/2008)	V(1a)	Se muestra un mensaje de error: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)".

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

		V(Enero)	I(#\$\$)	V(Enero)	V(12/05/2008)	V(1a)	El sistema no permite la entrada de ningún dato.	
		V(Enero)	V(Subsidio)	I(#\$\$)	V(12/05/2008)	V(1a)		
		V(Enero)	V(Subsidio)	V(Enero)	I(#\$\$)	V(1a)		
		V(Enero)	V(Subsidio)	V(Enero)	V(12/05/2008)	I(#\$%)		
EP 1.4	Cancelar operación	NA	NA	NA	NA	NA	Se cancela la operación y se cierra la	

4.3.2.2 Diseño de caso de prueba del requisito Agregar trabajadores a la nómina

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Administrar nómina.
- Se debe seleccionar la opción Administrar nómina.
- Debe de existir al menos una nómina en el sistema.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
----------------------	---------------------	-----------------------	---------------------

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

1. Agregar trabajadores a la nómina.	Se adiciona un trabajador a una nómina.	EP 1.1: Agregar trabajadores a una nómina.	<ul style="list-style-type: none"> – Se debe presionar el botón Adicionar. – Se debe seleccionar los trabajadores a adicionar a la nómina. – Se debe presionar el botón Aceptar.
		EP 1.2: Cancelar operación.	<ul style="list-style-type: none"> – Se debe presionar el botón Adicionar. – Se debe seleccionar o no los trabajadores a adicionar a una nómina. – Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Trabajador a buscar.	Campo de selección.	NO	Selección.

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Trabajador a buscar	Respuesta del sistema	Resultado de la prueba
EP 1.1	Agregar trabajador a una nómina.	V(SI)	Muestra un mensaje: "Los trabajadores han sido adicionados a la nómina"	

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

EP 1.2	Cancelar operación	NA	Se cancela la operación y se cierra la ventana.	
--------	--------------------	----	---	--

4.3.2.3 Diseño de caso de prueba del requisito Procesar nómina

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Administrar nómina.
- Se debe seleccionar la opción Administrar nómina.
- Debe existir al menos una nómina.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Procesar nómina.	Se procesa una nómina.	EP 1.1: Procesar nómina correctamente.	<ul style="list-style-type: none"> – Se debe seleccionar la nómina a procesar. – Se debe presionar el botón Procesar. – Se debe presionar el botón Aceptar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Procesar nómina correctamente.	Muestra un mensaje: "La nómina se ha procesado satisfactoriamente."	

4.4 Métricas orientadas a clases para evaluar el diseño

4.4.1 Tamaño de clase (TC)

Para medir el tamaño de clase se tienen en cuenta los siguientes aspectos:

- Total de operaciones, ya sean las propias o las heredadas de las clases padres e interfaces que implementen.
- Cantidad de atributos, tanto los de ella, como lo de los padres.
- Promedio general de los dos anteriores para el sistema completo.

Para evaluar las métricas son necesarios los umbrales; con los cuales las clases se clasifican en tres grupos según su tamaño, estos se representan en la siguiente tabla.

Tabla 1 Valores de los umbrales para TC

Clasificación	Valores de los umbrales
Pequeño	≤ 20
Medio	> 20 y ≤ 30
Grande	> 30

En la tabla 2 se ilustran las clases del sistema aplicándole la métrica seleccionada.

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

Tabla 2 Tamaño de las clases

No	Nombre	Cantidad Atributos	Cantidad Operaciones	Tamaño
1	AdminisitrarnominaController	0	9	Pequeño
2	NomTiponominaModel	0	4	Pequeño
3	NomTiponomina	0	10	Pequeño
4	BaseNomTiponomina	0	2	Pequeño
5	NomEstadonominaModel	0	4	Pequeño
6	NomEstadonomina	0	7	Pequeño
7	BaseNomEstadonomina	0	2	Pequeño
8	DatProcesamientoincidenciumModel	0	4	Pequeño
9	DatProcesamientoincidencium	0	10	Pequeño
10	BaseDatProcesamientoincidencium	0	2	Pequeño
11	DatNominaModel	0	10	Pequeño
12	DatNomina	0	21	Medio
13	BaseDatNomina	0	2	Pequeño
14	DatProcesamientooretencionesModel	0	4	Pequeño
15	DatProcesamientooretenciones	0	11	Pequeño
16	BaseDatProcesamientooretenciones	0	2	Pequeño
17	DatProcesamientoonominaModel	0	7	Pequeño
18	DatProcesamientoonomina	0	18	Pequeño
19	BaseDatProcesamientoonomina	0	2	Pequeño
20	DatProcesamientoopagosadicionalesModel	0	4	Pequeño
21	DatProcesamientoopagosadicionales	0	10	Pequeño
22	BaseDatProcesamientoopagosadicionales	0	2	Pequeño
23	DatPeriodopagoModel	0	6	Pequeño
24	DatPeriodopago	0	13	Pequeño
25	BaseDatPeriodopago	0	2	Pequeño
26	DatConceptopagoModel	0	4	Pequeño
27	DatConceptopago	0	11	Pequeño
28	BaseDatConceptopago	0	2	Pequeño
29	CierrecontableController	0	4	Pequeño
30	DatComprobanteoperacionesModel	0	4	Pequeño
31	DatComprobanteoperaciones	0	9	Pequeño
32	BaseDatComprobanteoperaciones	0	2	Pequeño

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

33	CierrecontableModel	0	5	Pequeño
34	ComprobanteoperacionesController	0	4	Pequeño
35	DatPasesModel	0	4	Pequeño
36	DatPases	0	8	Pequeño
37	BaseDatPases	0	2	Pequeño
38	DatAsientosModel	0	4	Pequeño
39	DatAsientos	0	8	Pequeño
40	BaseDatAsientos	0	2	Pequeño
41	DatConfiguracionModel	0	3	Pequeño
42	DatConfiguracion	0	8	Pequeño
43	BaseDatConfiguracion	0	2	Pequeño
44	ConfiguracionController	0	3	Pequeño
45	GestionarconceptopagoController	0	5	Pequeño
46	NomImpuestoModel	0	5	Pequeño
47	NomImpuesto	0	10	Pequeño
48	BaseNomImpuesto	0	2	Pequeño
49	GestionarfrecuenciapagoController	0	5	Pequeño
50	DatfrecuenciapagoModel	0	4	Pequeño
51	Datfrecuenciapago	0	10	Pequeño
52	BaseDatfrecuenciapago	0	2	Pequeño
53	GestionarimpuestoController	0	5	Pequeño
54	GestionarperiodopagoController	0	6	Pequeño
55	GestionartipoajusteController	0	5	Pequeño
56	NomtipoajustenominaModel	0	4	Pequeño
57	Nomtipoajustenomina	0	11	Pequeño
58	BaseNomtipoajustenomina	0	2	Pequeño
59	GestionartiponominaController	0	5	Pequeño
60	NomTiponominaModel	0	4	Pequeño
61	NomTiponomina	0	10	Pequeño
62	BaseNomTiponomina	0	2	Pequeño
63	NomfuncionesModel	0	2	Pequeño
64	Nomfunciones	0	8	Pequeño
65	BaseNomfunciones	0	2	Pequeño
66	RevisarnominaController	0	5	Pequeño
67	Ajusteconcepto_procnominaModel	0	4	Pequeño
68	Ajusteconcepto_procnomina	0	7	Pequeño
69	BaseAjusteconcepto_procnomina	0	2	Pequeño

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

70	Concepto_procnominaModel	0	4	Pequeño
71	Concepto_procnomina	0	8	Pequeño
72	BaseConcepto_procnomina	0	2	Pequeño
73	Impuesto_conceptopagoModel	0	4	Pequeño
74	Impuesto_conceptopago	0	7	Pequeño
75	BaseImpuesto_conceptopago	0	2	Pequeño
76	Tiponomina_funcionesModel	0	4	Pequeño
77	Tiponomina_funciones	0	6	Pequeño
78	BaseTiponomina_funciones	0	2	Pequeño

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

La mayoría de las clases que conforman el sistema están dentro de la categoría de pequeñas, lo que demuestra que el sistema no es complejo. Los resultados obtenidos son positivos según esta métrica, como se puede ver en las siguientes tablas.

Tabla 3 Cantidad de clases por clasificación

Clasificación	Cantidad Clases
Pequeño	77
Medio	1
Grande	0

Tabla 4 Resultados de la Métrica TC

Cantidad Clases	Promedio Atributos	Promedio Operaciones
78	0	5.42

4.4.2 Árbol de profundidad de herencia (APH)

Esta métrica está definida por la máxima longitud que exista entre el nodo y la raíz del árbol. Donde el nodo es una clase hija que hereda de una clase, y así respectivamente hasta llegar a la raíz. A medida

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

que esa longitud va creciendo, entonces se van heredando más operaciones y atributos por las clases hijas. Se hace difícil predecir el comportamiento de las clases que se encuentran en los niveles más bajos del árbol. Esta tiene sus ventajas y desventajas. Si los valores de APH son grandes, entonces se garantiza que se reutilice gran cantidad de código; pero al mismo tiempo hace que el diseño sea más complejo. Esto provoca un mayor acoplamiento entre las clases.

En el modelado de los diagramas de clases del diseño no se hizo necesario hacer demasiado uso de la herencia. Aplicando esta métrica al diseño propuesto se obtienen resultados que demuestran su poca complejidad, el árbol de profundidad de herencia toma valor 1, por lo que existe bajo acoplamiento y es de fácil reparación.

4.4.3 Relaciones entre Clases (RC)

Esta métrica está dada por la cantidad de relaciones de uso que existe entre las distintas clases que forman el diseño propuesto. Se le aplica a las mismas clases que le fue aplicada la métrica TC. Los aspectos de calidad que se miden son: Acoplamiento, Complejidad de mantenimiento, Reutilización y Cantidad de pruebas.

Tabla 5 Cantidad de relaciones de uso entre las clases

No	Nombre	Relaciones de uso
1	AdminisitrarnominaController	4
2	NomTiponominaModel	0
3	NomTiponomina	0
4	BaseNomTiponomina	0
5	NomEstadonominaModel	0
6	NomEstadonomina	0
7	BaseNomEstadonomina	0
8	DatProcesamientoincidenciumModel	0
9	DatProcesamientoincidencium	0
10	BaseDatProcesamientoincidencium	0
11	DatNominaModel	0

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

12	DatNomina	0
13	BaseDatNomina	0
14	DatProcesamientoretencionesModel	0
15	DatProcesamientoretenciones	0
16	BaseDatProcesamientoretenciones	0
17	DatProcesamientonominaModel	0
18	DatProcesamientonomina	0
19	BaseDatProcesamientonomina	0
20	DatProcesamientopagosadicionalesModel	0
21	DatProcesamientopagosadicionales	0
22	BaseDatProcesamientopagosadicionales	0
23	DatPeriodopagoModel	0
24	DatPeriodopago	0
25	BaseDatPeriodopago	0
26	DatConceptopagoModel	0
27	DatConceptopago	0
28	BaseDatConceptopago	0
29	CierrecontableController	2
30	DatComprobanteoperacionesModel	0
31	DatComprobanteoperaciones	0
32	BaseDatComprobanteoperaciones	0
33	CierrecontableModel	0
34	ComprobanteoperacionesController	1
35	DatPasesModel	0
36	DatPases	0
37	BaseDatPases	0
38	DatAsientosModel	0
39	DatAsientos	0

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

40	BaseDatAsientos	0
41	DatConfiguracionModel	0
42	DatConfiguracion	0
43	BaseDatConfiguracion	0
44	ConfiguracionController	1
45	GestionarconceptopagoController	2
46	NomImpuestoModel	0
47	NomImpuesto	0
48	BaseNomImpuesto	0
49	GestionarfrecuenciapagoController	0
50	DatfrecuenciapagoModel	0
51	Datfrecuenciapago	0
52	BaseDatfrecuenciapago	0
53	GestionarimpuestoController	2
54	GestionarperiodopagoController	1
55	GestionartipoajusteController	0
56	NomtipoajustenominaModel	0
57	Nomtipoajustenomina	0
58	BaseNomtipoajustenomina	0
59	GestionartiponominaController	0
60	NomTiponominaModel	0
61	NomTiponomina	0
62	BaseNomTiponomina	0
63	NomfuncionesModel	0
64	Nomfunciones	0
65	BaseNomfunciones	0
66	RevisarnominaController	2
67	Ajusteconcepto_procnominaModel	0

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

68	Ajusteconcepto_procnomina	0
69	BaseAjusteconcepto_procnomina	0
70	Concepto_procnominaModel	0
71	Concepto_procnomina	0
72	BaseConcepto_procnomina	0
73	Impuesto_conceptopagoModel	0
74	Impuesto_conceptopago	0
75	BaseImpuesto_conceptopago	0
76	Tiponomina_funcionesModel	0
77	Tiponomina_funciones	0
78	BaseTiponomina_funciones	0

Para medir el acoplamiento según los resultados de esta métrica, algunos especialistas plantean los siguientes valores.

Tabla 6 Acoplamiento

Categoría	Relaciones de uso	Cantidad de Clases
Ninguno	0	70
Bajo	1	3
Medio	2	4
Alto	>2	1

Los demás parámetros de calidad que mide esta métrica dependen del valor promedio de las dependencias de uso de todas las clases, en este caso ese promedio es de 0.19.

Tabla 7 Cantidad de Pruebas y Complejidad de Mantenimiento

Categoría	Criterio	Cantidad de Clases
Baja	\leq Prom.	70
Media	$>$ Prom. Y ≤ 2 *Prom.	0

CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS

Alta	>2*Prom.	8
------	----------	---

Tabla 8 Reutilización

Categoría	Criterio	Cantidad de Clases
Baja	>2*Prom.	8
Media	>Prom. Y <=2*Prom.	0
Alta	<=Prom.	70

De manera general los resultados obtenidos de esta métrica son positivos. El acoplamiento existente entre las clases es bajo, el 89% no tiene ningún acoplamiento, el 4% es bajo, el 6% es medio, y solo el 1% tiene alto acoplamiento. El nivel de reutilización de las clases es alto, puesto que el 89% de las clases pueden ser reutilizadas. De igual forma la cantidad de pruebas y la complejidad de mantenimiento tienen que el 89% de las clases son fáciles de reparar y la cantidad de pruebas a realizar es relativamente corta.

4.5 Conclusiones parciales

En este capítulo se realizó la validación de los requerimientos funcionales del sistema, utilizando para ello métricas como la correctitud, completitud y consistencia, de las cuales se obtuvo un resultado favorable. Se aplicaron también técnicas como los prototipos de interfaz de usuario lográndose una mayor comunicación con el cliente durante la captura de requisitos y el diseño de casos de prueba logrando mayor claridad en las posibles respuestas del sistema. Las métricas usadas en la validación del diseño, demostraron que el mismo posee un bajo acoplamiento de las clases, una alta reutilización de las mismas y la herencia que existe entre ellas es de grado 1; lo que lo hace sencillo y fácil implementar, quedando demostrada la calidad del diseño.

CONCLUSIONES

En la realización de este trabajo se logró darle cumplimiento a los objetivos específicos a través del desarrollo claro y definido de las tareas propuestas modelándose el proceso de Confección y cálculo de la nómina, cumpliendo estrictamente con los requisitos solicitados por el cliente con las mejoras y ventajas propuestas.

Para llevar a cabo el modelamiento del proceso se realizó un estudio del estado del arte analizando algunas soluciones existentes en Cuba y el mundo con características similares a la solución propuesta para el procesamiento de la nómina, tomando ideas fundamentales para la confección de un producto que permitiera una mayor gama de ventajas y funcionalidades. Se describieron las características principales de las herramientas, modelo de desarrollo y los lenguajes de modelado utilizados que conforman la fundamentación teórica para el desarrollo del trabajo.

Con la ayuda de las técnicas utilizadas para comprender el negocio se logró realizar la modelación del mismo que constituyó la base fundamental para la identificación de los requisitos funcionales.

A partir del modelo de negocio que se obtuvo se identificaron los requisitos funcionales del sistema. Las técnicas empleadas en la captura de los requisitos funcionales permitieron una precisa identificación y especificación de los mismos. Las métricas aplicadas en la validación de los requerimientos arrojaron resultados satisfactorios demostrando la consistencia, completitud y correctitud de las especificaciones de requisitos realizadas que según las validaciones realizadas con los clientes y usuarios finales satisface las necesidades reales de estos.

Una vez especificados los requisitos funcionales se realizó la modelación del diseño obteniéndose los diagramas correspondientes. Se aplicaron los patrones de diseño y se validaron mediante métricas seleccionadas arrojando resultados satisfactorios.

La solución que se propone contribuye a garantizar que el proceso de Confección y cálculo de la nómina se gestione de forma eficiente facilitando el control y seguimiento del mismo. Esta garantiza rapidez en el

CONCLUSIONES

procesamiento, posibilitando realizar el pago en las fechas establecidas y facilitando grandemente el trabajo realizados por los especialistas. Favorece la elaboración de estrategias e iniciativas organizacionales contribuyendo de esta manera a la elevación de la productividad de las entidades cubanas

Finalmente se logró la validación de la idea a defender y se cumplieron los objetivos propuestos.

RECOMENDACIONES

Después de realizadas las conclusiones de este trabajo se recomienda:

- Terminar la implementación del sistema con las funcionalidades propuestas.
- Continuar el estudio del tema con el objetivo de incluir nuevas funcionalidades en el procesamiento de la nómina.
- Realizar el despliegue del componente propuesto como parte del subsistema de Capital Humano del sistema Cedrux.

BIBLIOGRAFÍA

1. **Cuba, Organismo Nacional de Normalización de la República de.** Norma Cubana 3000 . Ciudad de la Habana : s.n., 2007.
2. Ayuda del Versat Sarasola; Version 2.0 actualizacion 5.8.
3. **Precios, Ministerio de Finanzas y.** Resolución 13 . 2007.
4. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El Proceso Unificado de Desarrollo de software* . 2000.
5. **James Rumbaugh, Ivar Jacobson, Grady Boch.** *Lenguaje Unificado de Modelado. Manual de Referencia.* s.l. : Addison-Wesley, 2000.
6. **Larman, Craig.** *UML y patrones, Introducción al análisis y diseño orientado a objeto.* 1999.
7. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque practico* . 2005.
8. **Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides.** *Design Patterns.* 1995.
9. **Rubio, Ivonne.** *Seminario de valor competitivo.* 2007.
10. **Lidia Fuentes, José M. Troya y Antonio Vallecillo.** *Desarrollo de Software Basado en Componentes.*
11. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque práctico.* 1998.
12. **Navathe, Elmasri y.** *Fundamentos de Sistemas de Bases de Datos.* 2002.
13. **Hector Garcia-Molina, Jeff Ullman, Jennifer Widom.** *Database systems: the complete book.* 2008.
14. Sitio oficial de Visual Paradigm. [En línea] <http://www.visual-paradigm.com/>.
15. Sitio oficial de ADAM 5. [En línea] <http://www.alcofin.com.mx/index.html>.
16. Sitio oficial de Assets. [En línea] <http://assets.co.cu/index.asp>.
17. Sitio oficial de Condor. [En línea] <http://www.sics.cu/productos.aspx>.

18. Sitio oficial de Oracle E-Business Suite Human Resources Management System. [En línea]
http://www.oracle.com/applications/human_resources/intro.html.
19. Sitio oficial de Rodas XXI. [En línea] <http://www.rodasxxi.cu/index.php>.
20. Sitio oficial de Sage Mas 500. [En línea] <http://www.sagemas.com/products/sagemas500>.
21. Sap España. [En línea] <http://www.sap.com/spain/index.epx>.
22. Sap. [En línea] <http://www.sap.com/index.epx>.
23. Sitio Oficial de Seven2000. [En línea] <http://www.seven2k.com/index.html>.
24. Sitio oficial de PHP. [En línea] <http://www.php.net/>.
25. Sitio oficial de Extjs. [En línea] <http://extjs.com/>.
26. The apache software Foundation. [En línea] <http://www.apache.org/>.
27. Sitio web XAMPP. [En línea] <http://www.apachefriends.org/es/xampp.html>.
28. **comunicaciones, Agencia de control y supervision Ministerio de la informática y las.** Software Contables-Financieros. [En línea] http://www.acs-mic.cu/sw_certificados.htm.
29. Visión general de las nuevas funcionalidades de Apache 2.0 . [En línea] <http://www.apache2.com/>.
30. **Polo, Toribio Guerra.** El VERSAT-Sarasola: Sistema cubano de Gestión Contable-Financiero;. [En línea] <http://eleccodelcontador.blogspot.com/2008/10/el-versat-sarasola-sistema-cubano-de.html>.
31. **Vallecillo, Lidia Fuentes y Antonio.** Una Introducción a los Perfiles UML. [En línea]
<http://www.lcc.uma.es/~av/Publicaciones/04/UMLProfiles-Novatica04.pdf>.
32. Ingeniería de software y su relación con las herramientas CASE. [En línea]
http://caterina.udlap.mx/u_dl_a/tales/documentos/lis/rea_c_ji/capitulo_2.html .
33. Using BPMN to Model a BPEL Process. [En línea]
<http://www.bpmn.org/Documents/Mapping%20BPMN%20to%20BPEL%20Example.pdf> .
34. **Cuba, Asamblea Nacional del Poder Popular de la República de.** Ley 73. 1994.

35. —. Ley No. 49, Código de Trabajo. 1985.
36. Sprax Systems . [En línea] [Citado el: 10 de 5 de 2009.]
http://www.sparxsystems.com.ar/resources/tutorial/uml2_componentdiagram.html.
37. **Cuba, Banco Nacional de.** Instrucción de Procedimiento No. 257 . 1968.
38. **Precios, Ministerio de Finanzas Y.** Resolución No. 105. 2008.
39. —. Resolución No. 116. 2002.
40. —. Resolución No. 198. 2006.
41. **Social, Ministerio de Trabajo y Seguridad.** Resolución 27 . 2006.
42. **IEEE-STD-830.** *IEEE-STD-830: ESPECIFICACIONES DE LOS REQUISITOS DEL.* 1998 .
43. PHP Web Application Server. [En línea] <http://www.zend.com/en/>.
44. Zend Studio. [En línea] <http://www.zend.com/products/studio/>.

ANEXOS

Anexo 1

Prototipo del requisito Adicionar impuesto y contribuciones.

Adicionar tipo de impuesto ó contribución

Código:

Denominación:

Por ciento:

Cuenta:

Base salario devengado Empresarial

Elemento de gasto

Otra cuenta

Aceptar Aplicar Cancelar

Prototipo del requisito Modificar impuesto y contribuciones.

Solo lectura

Modificar tipo de impuesto ó contribución

Código:

Denominación:

Por ciento:

Cuenta:

Base salario devengado Empresarial

Elemento de gasto

Otra cuenta

Aceptar Aplicar Cancelar

Prototipo del requisito Mostrar impuesto y contribuciones.

The image shows a software window titled "Gestionar tipos de obligaciones". At the top right of the window are three standard window control buttons: minimize, maximize, and close. Below the title bar, there are three buttons: "Adicionar", "Modificar", and "Eliminar". Below these buttons is a table with six columns: "Código", "Denominación", "Por ciento", "Cuenta", "Elemento de gasto", and "Base Salario Devengado". The table is currently empty, showing only the header row.

Código	Denominación	Por ciento	Cuenta	Elemento de gasto	Base Salario Devengado

Anexo 2

Prototipo del requisito Adicionar concepto de pago.

Adicionar concepto de pago

Código:

Denominación:

Descripción:

Vacaciones

Acumula Tiempo:

Acumula Importe: ▼

Débitos

Elemento de gasto: ▼

Otra cuenta: ▼

Impuestos Empresariales

Código	Denominación	marcar

Aceptar Cancelar

Prototipo del requisito Modificar concepto de pago.

Solo lectura

Modificar concepto de pago

Código:

Denominación:

Descripción:

Vacaciones

Acumula Tiempo:

Acumula Importe:

Débitos

Elemento de gasto:

Otra cuenta:

Impuestos Empresariales

Código	Denominación	Porcentaje	Seleccionar
			<input type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>

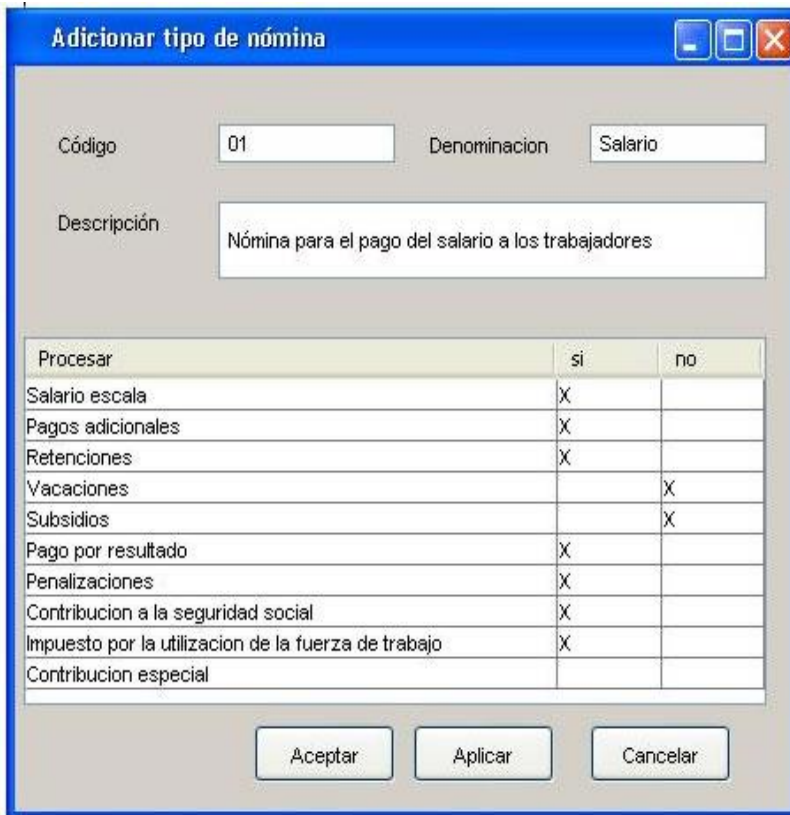
Aceptar Cancelar

Prototipo del requisito Mostrar concepto de pago.



Anexo 3

Prototipo del requisito Adicionar tipo de nómina.



Prototipo del requisito Modificar tipo de nómina.

Solo lectura

Modificar tipo de nómina

Código: Denominación:

Descripción:

Procesar	si	no
Salario escala	X	
Pagos adicionales	X	
Retenciones	X	
Vacaciones		X
Subsidios		X
Pago por resultado	X	
Penalizaciones	X	
Contribucion a la seguridad social	X	
Impuesto por la utilizacion de la fuerza de trabajo	X	
Contribucion especial		

Aceptar Aplicar Cancelar

Prototipo del requisito Mostrar tipo de nómina.

Tipos de nómina

Adicionar Modificar Eliminar

Código	Denominación	Descripción

Anexo 4

Prototipo del requisito Adicionar período de pago.

Adicionar período de pago

Fecha inicio:

Denominación:

Fecha fin:

Frecuencia de pago:

Programa a aplicar:

Período de pago anterior:

Aplicar Aceptar Cancelar

Detailed description: This is a software prototype for a dialog box titled 'Adicionar período de pago'. It features a blue title bar with standard window controls (minimize, maximize, close). The main area has a light gray background and contains six input fields arranged in two columns. The left column includes 'Fecha inicio', 'Fecha fin', and 'Programa a aplicar', each with a dropdown arrow. The right column includes 'Denominación', 'Frecuencia de pago', and 'Período de pago anterior', with the first being a text box and the others dropdowns. At the bottom, there are three buttons: 'Aplicar', 'Aceptar', and 'Cancelar'.

Prototipo del requisito Modificar período de pago.

Modificar período de pago

Fecha inicio:

Denominación:

Fecha fin:

Frecuencia de pago:

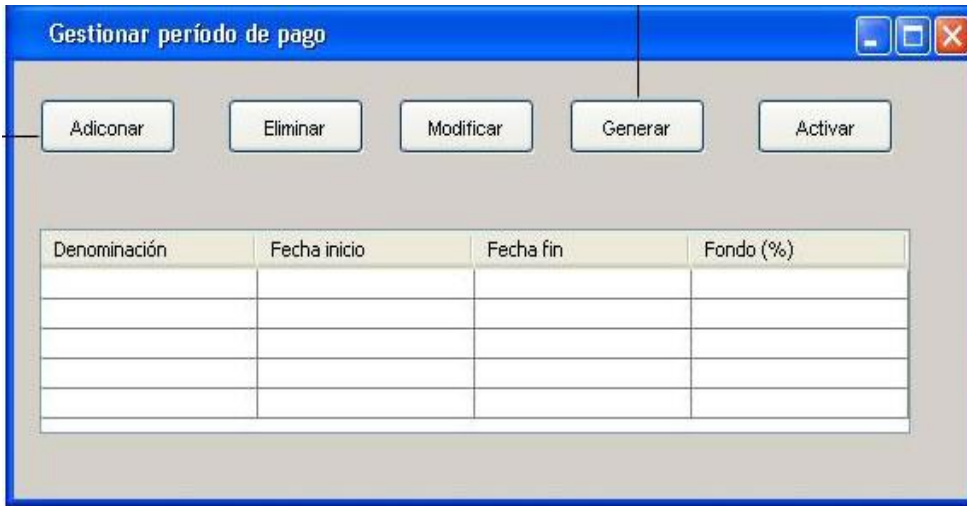
Programa a aplicar:

Período de pago anterior:

Aplicar Aceptar Cancelar

Detailed description: This is a software prototype for a dialog box titled 'Modificar período de pago'. It features a blue title bar with standard window controls (minimize, maximize, close). The main area has a light gray background and contains six input fields arranged in two columns. The left column includes 'Fecha inicio', 'Fecha fin', and 'Programa a aplicar', each with a dropdown arrow. The right column includes 'Denominación', 'Frecuencia de pago', and 'Período de pago anterior', with the first being a text box and the others dropdowns. At the bottom, there are three buttons: 'Aplicar', 'Aceptar', and 'Cancelar'.

Prototipo del requisito Mostrar período de pago.



Gestionar período de pago

Adicionar Eliminar Modificar Generar Activar

Denominación	Fecha inicio	Fecha fin	Fondo (%)

Anexo 5

Prototipo del requisito Adicionar tipo de ajuste.



Adicionar tipo de ajuste

Código

Denominación

Descripción

Cancelar Aplicar Aceptar

Prototipo del requisito Modificar tipo de ajuste.

The image shows a dialog box titled "Modificar tipo de ajuste" with a blue header bar containing standard window control buttons (minimize, maximize, close). A blue callout box labeled "Solo lectura" is positioned above the dialog, with a line pointing to a text input field for "Código". Below this are input fields for "Denominación" and "Descripción". At the bottom, there are three buttons: "Cancelar", "Aplicar", and "Aceptar".

Prototipo del requisito Mostrar tipo de ajuste.

The image shows a window titled "Gestionar tipos de ajustes" with a blue header bar and window control buttons. At the top, there are three buttons: "Adicionar", "Modificar", and "Eliminar". Below these is a table with three columns: "Codigo", "Denominacion", and "Descripcion". The table contains 10 empty rows.

Codigo	Denominacion	Descripcion

Anexo 6

Prototipo del requisito Crear nómina.

Adicionar nómina

Código: Denominación:

Fecha: Tipo de nómina:

Período de pago:

Aceptar Apicar Cancelar

Detailed description: This is a software prototype for a 'Adicionar nómina' (Add payroll) dialog box. It features a blue title bar with standard window controls. The main area has a light gray background and contains three rows of input fields. The first row has 'Código:' and 'Denominación:' labels followed by text input boxes. The second row has 'Fecha:' and 'Tipo de nómina:' labels; 'Fecha:' is followed by a date picker (green background), and 'Tipo de nómina:' is followed by a dropdown menu. The third row has 'Período de pago:' followed by a dropdown menu. At the bottom, there are three buttons: 'Aceptar', 'Apicar', and 'Cancelar'.

Prototipo del requisito Modificar nómina.

Solo lectura

Modificar nómina

Código: Denominación:

Fecha: Tipo de nómina:

Período de pago:

Aceptar Apicar Cancelar

Detailed description: This is a software prototype for a 'Modificar nómina' (Modify payroll) dialog box. It is identical in layout to the 'Adicionar nómina' dialog but includes a blue callout box labeled 'Solo lectura' (Read-only) pointing to the 'Código:' input field, which is rendered as a disabled gray box. The other fields and buttons are the same as in the previous prototype.

Prototipo del requisito Agregar trabajadores a la nómina.

Adicionar trabajador

Trabajadores de nómina

Nombre	Apellidos

<< >>

Trabajador a buscar

Nombre	Apellidos

Aceptar Cancelar

Prototipo del requisito Mostrar nóminas.

Administrar nómina

Adicionar Modificar Eliminar Procesar Período de pago

Código	Denominación	Fecha	Período de pago	Estado	Tipo de nómina

Trabajadores asociados a la nómina Adicionar Eliminar

Expediente	Nombre	Apellidos	Área

Anexo 7

Prototipo del requisito Mostrar nóminas.

El prototipo muestra una ventana de software con el título "Revisar nomina". En la parte superior, hay tres botones: "Revertir", "Confirmar" y "Mostrar comprobante". Debajo de los botones, hay una tabla con tres columnas: "Codigo", "Denominacion" y "Estado".

Codigo	Denominacion	Estado

Debajo de la tabla, hay un botón "Detalles". Al hacer clic en "Detalles", se muestra una segunda tabla con cuatro columnas: "Expediente", "Nombre", "Apellido" y "Salario".

Expediente	Nombre	Apellido	Salario

Prototipo del requisito Mostrar detalles del trabajador de una nómina.

Detalles

Ajuste

identificador	valor
-----Trabajador	
nombre	
Expediente	
Puesto de trabajo	
-----Salario	
Salario escala	
-----Tiempo	
Tiempo a trabajar	
Tiempo trabajado	
-----Pagos adicionales	
Pago 1	
Pago 2	
Pago 3	
-----Incidencias	
Incidencias 1	
Incidencias 2	
Incidencias 3	

Aceptar Cancelar

Prototipo del requisito Ajustar nómina.

Solo lectura

Realizar ajuste

Valor anterior:

Nuevo valor:

Tipo de ajuste:

Aceptar Cancelar

Anexo 8

Prototipo del requisito Revisar comprobante de operaciones.

Prototipo de la ventana 'Detalles comprobante'. La ventana tiene un título azul con el texto 'Detalles comprobante' y botones de control de ventana (minimizar, maximizar, cerrar). El contenido principal es una tabla con 5 columnas: 'Cuenta', 'Subcuenta', 'subcentro', 'debito' y 'Credito'. La tabla tiene 10 filas vacías. En la parte inferior derecha hay un botón 'Cerrar'.

Cuenta	Subcuenta	subcentro	debito	Credito

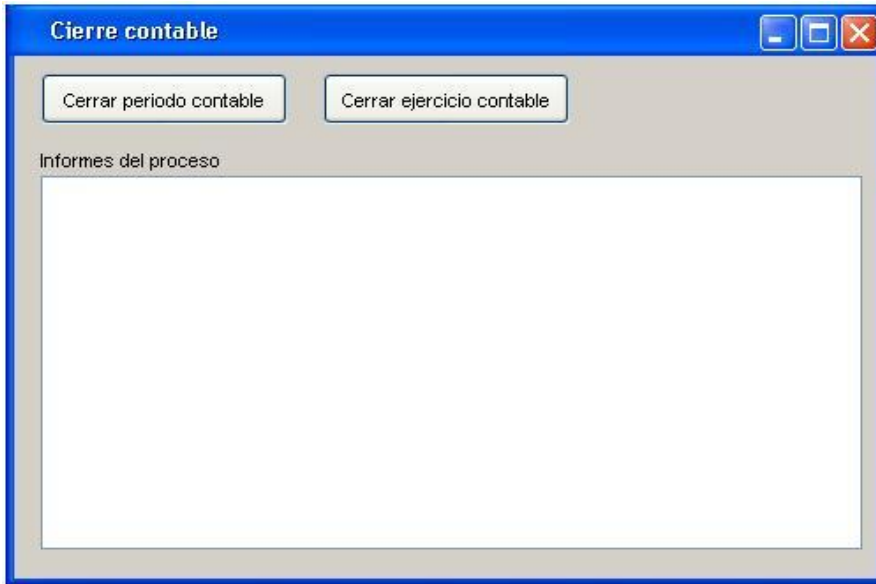
Prototipo del requisito Emitir comprobante de operaciones.

Prototipo de la ventana 'comprobante de operaciones'. La ventana tiene un título azul con el texto 'comprobante de operaciones' y botones de control de ventana (minimizar, maximizar, cerrar). En la parte superior hay dos botones: 'Detalles' y 'Emitir'. A la derecha de estos botones hay el texto 'Filtrar por período' y un menú desplegable. El contenido principal es una tabla con 4 columnas: 'Número', 'Fecha', 'Nómina' y 'Estado'. La tabla tiene 10 filas vacías.

Número	Fecha	Nómina	Estado

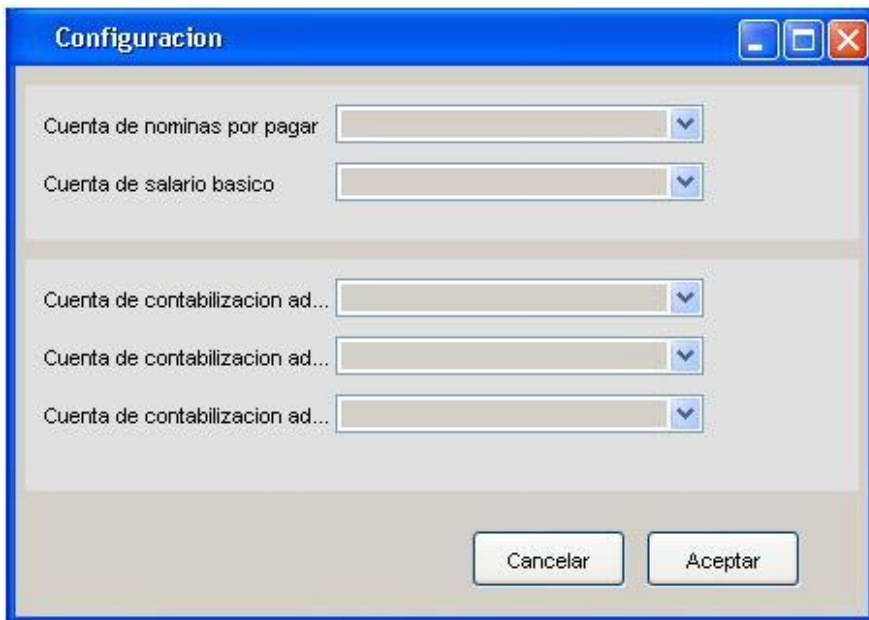
Anexo 9

Prototipo del requisito Efectuar cierre contable



Anexo 10

Prototipo del requisito Establecer configuración de cuentas.



Anexo 11

Prototipo del requisito Adicionar frecuencia de pago.

Adicionar frecuencia

Denominación

Cantidad por mes

Aceptar Aplicar Cancelar

Detailed description: This is a software prototype for a dialog box titled 'Adicionar frecuencia'. The window has a blue title bar with standard minimize, maximize, and close buttons. The main area is light gray and contains two text input fields. The first field is labeled 'Denominación' and the second is labeled 'Cantidad por mes'. At the bottom, there are three buttons: 'Aceptar', 'Aplicar', and 'Cancelar'.

Prototipo del requisito Modificar frecuencia de pago.

Modificar frecuencia

Denominación

Cantidad por mes

Aceptar Aplicar Cancelar

Detailed description: This is a software prototype for a dialog box titled 'Modificar frecuencia'. It has the same layout as the 'Adicionar frecuencia' dialog, with a blue title bar, two text input fields labeled 'Denominación' and 'Cantidad por mes', and three buttons at the bottom: 'Aceptar', 'Aplicar', and 'Cancelar'.

Prototipo del requisito Mostrar frecuencia de pago.

Denominacion	cantidad por mes

Anexo 12

Prototipo del requisito Listado para el pago por tarjeta magnética.

Reportes de Listado para el pago por tarjeta magnética

Nombre de la Entidad: _____ Código ONE: _____ Fecha: _____

Área de trabajo: _____

Expediente laboral	Nombre y apellidos	Tipo de nómina	Importe total	Número de tarjeta

Prototipo del requisito Listado para pago tradicional.

Reporte para detalles de nómina

Entidad:----- Período de la nómina:----- Tipo de nómina:-----

Código de la entidad -----	Área de trabajo -----	Elaborado por: -----
Nombre de la entidad -----		Revisado por: -----
Período de la nómina -----	Tipo de nómina -----	Contabilizado por: -----
Código de la nómina -----	Denominación -----	Fecha -----
		Cheque -----
		Fecha -----
		Importe -----
		Días -----
Expediente laboral -----	Nombre y apellido -----	Categoría ocupacional -----
	Salario escalatrabajadas -----	Horas adicionales -----
	Pagos adicionales -----	Otros pagos -----
	Penalizaciones -----	Salario bruto -----
	Retenciones -----	Contribución especial -----
		Salario acumulado de las -----
		vacaciones -----
		Salario acumulado de las -----
		vacaciones -----
		Firma -----

Realizado por: _____ Aprobado por: _____

Prototipo del requisito Distribución de moneda.

**Distribución de moneda -
Nómina de salario**

Período:----- "Fecha:
".date("d/m/Y")

Billetes							Monedas				Total
Cien	Cincuenta	Veinte	Diez	Cinco	Tres	Uno	Veinte centavos	Cinco centavos	Tres pesos	Un peso	
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Prototipo del requisito Comprobante de operaciones

Comprobante de operaciones

Nombre de la entidad: XXXXXX XXXXX **Tipo de la entidad:** presupuestada ó empresarial
Período de la nómina: XXXXXX XXXX **Tipo de nómina:** XXXXXX XXXXXXXX
Denominación de la nómina: XXX XXXX **Estado:** Confirmado ó enviado

Cuenta	Subcuenta	Subcentro	Débito	Cédito

Anexo 13

Diseño de caso de prueba del requisito Adicionar impuesto y contribuciones.

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Gestionar impuesto y contribuciones.
- Se debe seleccionar la opción Gestionar impuesto y contribuciones.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Adicionar impuesto y contribuciones.	Se adiciona un tipo de impuesto.	EP 1.1: Adicionar tipo de impuesto introduciendo los datos correctamente.	<ul style="list-style-type: none">– Se debe presionar el botón Adicionar.– Se debe introducir los datos del tipo de impuesto en el formulario.– Se debe presionar el botón Aceptar.

		<p>EP 1.2: Adicionar tipo de impuesto dejando campos requeridos en blanco.</p>	<ul style="list-style-type: none"> - Se debe presionar el botón Adicionar. - Se debe introducir los datos del tipo de impuesto en el formulario dejando al menos un campo requerido en blanco. - Se debe presionar el botón Aceptar.
--	--	--	---

		<p>EP 1.3: Adicionar tipo de impuesto introduciendo errores en los datos.</p>	<ul style="list-style-type: none"> - Se debe presionar el botón Adicionar. - Se debe introducir los datos del tipo de impuesto en el formulario introduciendo algunos datos inválidos. - Se debe presionar el botón Aceptar.
		<p>EP 1.4: Cancelar operación.</p>	<ul style="list-style-type: none"> - Se debe presionar el botón Adicionar. - Se debe introducir los datos del tipo de impuesto o no en el formulario. - Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Código.	Campo de texto.	NO	Combinación de letras y números.
2	Cuenta.	Lista desplegable.	NO	Selección.
3	Base salario devengado.	Campo de selección.	SI	Selección.
4	Empresarial.	Campo de selección.	SI	Selección.
5	Elemento de gasto.	Campo de selección.	SI	Selección.
6	Otra cuenta.	Campo de selección.	SI	Selección.
7	Denominación.	Campo de texto.	NO	Combinación de letras y números.
8	Porcentaje.	Campo de texto.	NO	Combinación de números.

1.1.2 Juegos de datos a probar

Id del escenario	Esce nario	Códi go	Cuent a	Base salario deveng ado	Empresar ial	Elemento de gasto	Otra cuenta	Denomi nación	Por cien to	Respue sta del sistema	Resu ltado de la prue ba
------------------	------------	---------	---------	-------------------------	--------------	-------------------	-------------	---------------	-------------	------------------------	--------------------------

EP 1.1	Adicionar tipo de impuesto introduciendo los datos correctamente.	V(1a)	V(sueldo)	V(SI)	V(Vacío)	V(elemento)	V(Vacío)	V(dada2)	V(23)	Muestra un mensaje: "El tipo de impuesto se creó satisfactoriamente."
EP 1.2	Adicionar tipo de impuesto dejando campos requeridos en blanco.	I(Vacío)	V(sueldo)	V(SI)	V(Vacío)	V(elemento)	V(Vacío)	V(dada2)	V(23)	Se muestra un mensaje de error: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)."
		V(1a)	I(Vacío)	V(SI)	V(Vacío)	V(elemento)	V(Vacío)	V(dada2)	V(23)	
		V(1a)	V(sueldo)	V(Vacío)	V(SI)	V(Vacío)	V(salario)	I(Vacío)	V(23)	
		V(1a)	V(sueldo)	V(Vacío)	V(SI)	V(Vacío)	V(salario)	V(dada2)	I(Vacío)	

EP 1.3	Adicionar tipo de impuesto introduciendo errores en los datos	I (#\$%)	V(sueldo)	V(SI)	V(Vacío)	V(elemento)	V(Vacío)	V(dada2)	V(23)	Se muestra un mensaje de error: "Por favor verifique nuevamente que	
		V(1a)	V(sueldo)	V(SI)	V(Vacío)	V(elemento)	V(Vacío)	I(@#%)	V(23)		
		V(1a)	V(sueldo)	V(SI)	V(Vacío)	V(elemento)	V(Vacío)	V(dada2)	I(@aas)		
	V(1a)	I (#\$%)	V(SI)	V(Vacío)	V(elemento)	V(Vacío)	V(dada2)	V(23)	El sistema no permite la entrada de ningún dato.		
	V(1a)	V(sueldo)	V(Vacío)	V(SI)	I(dsda@)	V(Vacío)	V(dada2)	V(23)			
V(1a)	V(sueldo)	V(Vacío)	V(SI)	V(Vacío)	I(dsda@)	V(dada2)	V(23)				
EP 1.4	Cancelar operación	NA	NA	NA	NA	NA	NA	NA	NA	Se cancela la operación	

Diseño de caso de prueba del requisito Modificar impuesto y contribuciones

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.

- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Gestionar impuesto y contribuciones.
- Se debe seleccionar la opción Gestionar impuesto y contribuciones.
- Debe de existir al menos un tipo de impuesto en el sistema.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Modificar impuesto y contribuciones.	Se modifica un tipo de impuesto.	EP 1.1: Modificar tipo de impuesto introduciendo los datos correctamente.	<ul style="list-style-type: none"> – Se debe presionar el botón Modificar. – Se debe introducir los datos del tipo de impuesto en el formulario. – Se debe presionar el botón Aceptar.
		EP 1.2: Modificar tipo de impuesto dejando campos requeridos en blanco.	<ul style="list-style-type: none"> – Se debe presionar el botón Modificar. – Se debe introducir los datos del tipo de impuesto en el formulario dejando al menos un campo requerido en blanco. – Se debe presionar el botón Aceptar.

		EP 1.3: Modificar tipo de impuesto introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se debe presionar el botón Modificar. - Se debe introducir los datos del tipo de impuesto en el formulario introduciendo algunos datos inválidos. - Se debe presionar el botón Aceptar.
		EP 1.4: Cancelar operación.	<ul style="list-style-type: none"> - Se debe presionar el botón Modificar. - Se debe introducir los datos del tipo de impuesto o no en el formulario. - Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Código.	Campo de texto.	NO	Combinación de letras y números.
2	Cuenta.	Lista desplegable.	NO	Selección.
3	Base salario	Campo de selección.	SI	Selección.
4	Empresarial.	Campo de selección.	SI	Selección.
5	Elemento de gasto.	Campo de selección.	SI	Selección.

6	Otra cuenta.	Campo de selección.	SI	Selección.
7	Denominación.	Campo de texto.	NO	Combinación de números.
8	Porciento.	Campo de texto.	NO	Combinación de letras y números.

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Código	Cuenta	Base salario devengado	Empresarial	Elemento de gasto	Otra cuenta	Denominación	Porciento	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar tipo de impuesto introduciendo los datos correctamente.	V(1a)	V(sueldo)	V(SI)	V(Vacío)	V(12)	V(Vacío)	V(dada 2)	V(23)	Muestra un mensaje: "El tipo de impuesto se modificó satisfactoriamente."	
EP 1.2	Modificar tipo de impuesto dejando campos requeridos en blanco.	I(Vacío)	V(sueldo)	V(SI)	V(Vacío)	V(12)	V(Vacío)	V(dada 2)	V(23)	Se muestra un mensaje de error: "Por favor verifique nuevamente que hay campo(s) con valor(es)	
		V(1a)	I(Vacío)	V(SI)	V(Vacío)	V(12)	V(Vacío)	V(dada 2)	V(23)		
		V(1a)	V(sueldo)	V(Vacío)	V(SI)	V(Vacío)	V(salario)	I(Vacío)	V(23)		

			o)							incorrecto(s).”.	
		V(1a)	V(sueldo)	V(Vacio)	V(SI)	V(Vacio)	V(salario)	V(dada 2)	I(Vacio)		
EP 1.3	Modificar tipo de impuesto introduciendo errores en los datos.	I(#\$%)	V(sueldo)	V(SI)	V(Vacio)	V(12)	V(Vacio)	V(dada 2)	V(23)	Se muestra un mensaje de error: “Por favor	
		V(1a)	V(sueldo)	V(SI)	V(Vacio)	V(12)	V(Vacio)	I(@#%)	V(23)	verifique nuevamente que hay campo(s)	
		V(1a)	V(sueldo)	V(SI)	V(Vacio)	V(12)	V(Vacio)	V(dada 2)	I(@aasas)	con valor(es) incorrecto(s).”.	
		V(1a)	I(#\$%)	V(SI)	V(Vacio)	V(12)	V(Vacio)	V(dada 2)	V(23)	El sistema no permite la entrada de ningún dato.	
		V(1a)	V(sueldo)	V(Vacio)	V(SI)	I(dsda@)	V(Vacio)	V(dada 2)	V(23)		

		V(1a)	V(sueldo)	V(Vacío)	V(SI)	V(Vacío)	I(dsda@)	V(dada2)	V(23)		
EP 1.4	Cancelar operación	NA	NA	NA	NA	NA	NA	NA	NA	Se cancela la operación y se cierra la ventana.	

Diseño de caso de prueba del requisito Eliminar impuesto y contribuciones

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Gestionar impuesto y contribuciones.
- Se debe seleccionar la opción Gestionar impuesto y contribuciones.
- Debe de existir por lo menos un tipo de impuesto en el sistema.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
----------------------	---------------------	-----------------------	---------------------

1. Eliminar impuesto y contribuciones.	Se elimina un tipo de impuesto.	EP 1.1: Eliminar tipo de impuesto correctamente.	<ul style="list-style-type: none"> – Se debe seleccionar el tipo de impuesto a eliminar. – Se debe presionar el botón Eliminar. – Se debe presionar el botón Aceptar.
		EP 1.2: Cancelar operación.	<ul style="list-style-type: none"> – Se debe seleccionar el tipo de impuesto a eliminar. – Se debe presionar el botón Eliminar. – Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar tipo de impuesto correctamente.	Muestra un mensaje: "El impuesto se ha eliminado satisfactoriamente."	
EP 1.2	Cancelar operación	Se cancela la operación y se cierra la ventana.	

Anexo 14

Diseño de caso de prueba del requisito Adicionar concepto de pago

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Gestionar conceptos de pago.
- Se debe seleccionar la opción Gestionar conceptos de pago.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Adicionar conceptos de pago.	Se adiciona un concepto de pago.	EP 1.1: Adicionar concepto de pago introduciendo los datos correctamente.	<ul style="list-style-type: none">– Se debe presionar el botón Adicionar.– Se debe introducir los datos del concepto de pago en el formulario.– Se debe presionar el botón Aceptar.

		<p>EP 1.2: Adicionar concepto de pago dejando campos requeridos en blanco.</p>	<ul style="list-style-type: none"> - Se debe presionar el botón Adicionar. - Se debe introducir los datos del concepto de pago en el formulario dejando al menos un campo requerido en blanco. - Se debe presionar el botón Aceptar.
		<p>EP 1.3: Adicionar concepto de pago introduciendo errores en los datos.</p>	<ul style="list-style-type: none"> - Se debe presionar el botón Adicionar. - Se debe introducir los datos del concepto de pago en el formulario introduciendo algunos datos inválidos. - Se debe presionar el botón Aceptar.
		<p>EP 1.4: Cancelar operación.</p>	<ul style="list-style-type: none"> - Se debe presionar el botón Adicionar. - Se debe introducir los datos del concepto de pago o no en el formulario. - Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Código.	Campo de texto.	NO	Combinación de letras y números.
2	Acumula importe.	Lista desplegable.	NO	Selección.
3	Acumula tiempo.	Campo de selección.	SI	Selección.
4	Elemento de gasto.	Campo de selección.	SI	Selección.
5	Otra cuenta.	Campo de selección.	SI	Selección.
6	Denominación.	Campo de texto.	NO	Combinación de letras y números.
7	Descripción.	Campo de texto.	NO	Combinación de letras y números.
8	Impuestos empresariales	Campo de selección.	SI	Selección.

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Código	Acumula importe	Acumula tiempo	Elemento de gasto	Otra cuenta	Denominación	Descripción	Impuestos empresariales	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar concepto de	V(1a)	V(dasdas)	V(SI)	V(elemento)	V(Vacío)	V(dada2)	V(aaa2)	V(Seleccionar impuesto)	Muestra un mensaje : “El	

	pago introduciendo	V(1a)	V(dasdas)	V(Vacío)	V(elemento)	V(Vacío)	V(dada2)	V(aaa2)	V(Seleccionar impuesto)	concepto de pago se	
EP 1.2	Adicionar concepto de pago dejando o campos requeridos en blanco.	I(Vacío)	V(dasdas)	V(SI)	V(elemento)	V(Vacío)	V(dada2)	V(aaa2)	V(Seleccionar impuesto)	Se muestra un mensaje de error: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)".	
		V(1a)	I(Vacío)	V(SI)	V(elemento)	V(Vacío)	V(dada2)	V(aaa2)	V(Seleccionar impuesto)		
		V(1a)	V(dasdas)	V(Vacío)	V(Vacío)	V(salario)	I(Vacío)	V(aaa2)	V(Seleccionar impuesto)		
		V(1a)	V(dasdas)	V(Vacío)	V(Vacío)	V(salario)	V(dada2)	I(Vacío)	V(Seleccionar impuesto)		
EP 1.3	Adicionar concepto de pago introduciendo errores	I(#\$%)	V(dasdas)	V(SI)	V(elemento)	V(Vacío)	V(dada2)	V(aaa2)	V(Seleccionar impuesto)	Se muestra un mensaje de error: "Por favor verifique	

	en los datos.	V(1a)	V(dasdas)	V(SI)	V(elemento)	V(Vacío)	I(@#%)	V(aaa2)	V(Seleccionar impuesto)	nuevamente que hay campo(s) con valor(es) incorrecto(s).".	
		V(1a)	V(dasdas)	V(SI)	V(elemento)	V(Vacío)	V(dada2)	I(@#\$)	V(Seleccionar impuesto)		
		V(1a)	I(#\$%)	V(SI)	V(elemento)	V(Vacío)	V(dada2)	V(aaa2)	V(Seleccionar impuesto)	El sistema no permite la entrada de ningún dato.	
		V(1a)	V(dasdas)	V(Vacío)	V(elemento)	V(Vacío)	V(dada2)	V(aaa2)	I(\$#)		
		V(1a)	V(dasdas)	V(Vacío)	I(dsda@)	V(Vacío)	V(dada2)	V(aaa2)	V(Seleccionar impuesto)		
		V(1a)	V(dasdas)	V(Vacío)	V(Vacío)	I(dsd a@)	V(dada2)	V(aaa2)	V(Seleccionar impuesto)		
EP 1.4	Cancelar operación	NA	NA	NA	NA	NA	NA	NA	NA	Se cancela la operación y se cierra la ventana.	

Diseño de caso de prueba del requisito Modificar concepto de pago

1. Condiciones de ejecución.

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Gestionar conceptos de pago.
- Se debe seleccionar la opción Gestionar conceptos de pago.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Modificar conceptos de pago.	Se modifica un concepto de pago.	EP 1.1: Modificar concepto de pago introduciendo los datos correctamente.	<ul style="list-style-type: none"> – Se debe presionar el botón Modificar. – Se debe introducir los datos del concepto de pago en el formulario. – Se debe presionar el botón Aceptar.
		EP 1.2: Modificar concepto de pago dejando campos requeridos en blanco.	<ul style="list-style-type: none"> – Se debe presionar el botón Modificar. – Se debe introducir los datos del concepto de pago en el formulario dejando al menos un campo requerido en blanco. – Se debe presionar el botón Aceptar.

		EP 1.3: Modificar concepto de pago introduciendo errores en los datos.	<ul style="list-style-type: none"> – Se debe presionar el botón Modificar. – Se debe introducir los datos del concepto de pago en el formulario introduciendo algunos datos inválidos. – Se debe presionar el botón Aceptar.
		EP 1.4: Cancelar operación.	<ul style="list-style-type: none"> – Se debe presionar el botón Modificar. – Se debe introducir los datos del concepto de pago o no en el formulario. – Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Código.	Campo de texto.	NO	Combinación de letras y números.
2	Acumula importe.	Lista desplegable.	NO	Selección.
3	Acumula tiempo.	Campo de selección.	SI	Selección.
4	Elemento de gasto.	Campo de selección.	SI	Selección.
5	Otra cuenta.	Campo de selección.	SI	Selección.

6	Denominación.	Campo de texto.	NO	Combinación de letras y números.
7	Descripción.	Campo de texto.	NO	Combinación de letras y números.
8	Impuestos empresariales	Campo de selección.	SI	Selección.

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Código	Acumula importe	Acumula tiempo	Elemento de gasto	Otra cuenta	Denominación	Descripción	Impuestos empresariales	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar concepto de pago introduciendo los datos correctamente.	V(1a)	V(dasdas)	V(SI)	V(elemento)	V(Vacío)	V(dada2)	V(aaa2)	V(Seleccionar impuesto)	Muestra un mensaje: "El concepto de pago se modifico satisfactoriamente."	
		V(1a)	V(dasdas)	V(Vacío)	V(elemento)	V(Vacío)	V(dada2)	V(aaa2)	V(Seleccionar impuesto)		
EP 1.2	Modificar concepto de pago	V(Vacío)	V(dasdas)	V(SI)	V(elemento)	V(Vacío)	V(dada2)	V(aaa2)	V(Seleccionar impuesto)	Se muestra un mensaje de error:	
		V(1a)	V(Vacío)	V(SI)	V(elemento)	V(Vacío)	V(dada2)	V(aaa2)	V(Seleccionar impuesto)		

	dejando campo(s) requeridos en blanco.	V(1a)	V(dasdas)	V(Vacío)	V(Vacío)	V(salario)	I(Vacío)	V(aaa2)	V(Seleccionar impuesto)	“Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s).”.
		V(1a)	V(dasdas)	V(Vacío)	V(Vacío)	V(salario)	V(dada2)	I(Vacío)	V(Seleccionar impuesto)	
EP 1.3	Modificar concepto de pago introduciendo errores en los datos.	I(#\$%)	V(dasdas)	V(SI)	V(elemento)	V(Vacío)	V(dada2)	V(aaa2)	V(Seleccionar impuesto)	Se muestra un mensaje de error: “Por favor verifique nuevamente que
		V(1a)	V(dasdas)	V(SI)	V(elemento)	V(Vacío)	I(@#%)	V(aaa2)	V(Seleccionar impuesto)	El sistema no permite la entrada de
		V(1a)	V(dasdas)	V(SI)	V(elemento)	V(Vacío)	V(dada2)	I(@#\$)	V(Seleccionar impuesto)	
		V(1a)	I(#\$%)	V(SI)	V(elemento)	V(Vacío)	V(dada2)	V(aaa2)	V(Seleccionar impuesto)	
		V(1a)	V(dasdas)	V(Vacío)	V(elemento)	V(Vacío)	V(dada2)	V(aaa2)	I(\$#)	

		V(1a)	V(dasd as)	V(V acío)	I(dsda @)	V(Vací o)	V(dada2)	V(aaa2)	V(Selec cionar impuest o)	ningún dato.	
		V(1a)	V(dasd as)	V(V acío)	V(Vací o)	I(dsda @)	V(dada2)	V(aaa2)	V(Selec cionar impuest o)		
EP 1.4	Cancel ar operaci ón	NA	NA	NA	NA	NA	NA	NA	NA	Se cancela la operació n y se cierra la ventana.	

Diseño de caso de prueba del requisito Eliminar concepto de pago.

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Gestionar conceptos de pago.
- Se debe seleccionar la opción Gestionar conceptos de pago.
- Debe de existir por lo menos un concepto de pago en el sistema.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
----------------------	---------------------	-----------------------	---------------------

1. Eliminar conceptos de pago.	Se elimina un concepto de pago.	EP 1.1: Eliminar concepto de pago correctamente.	<ul style="list-style-type: none"> - Se debe seleccionar el concepto de pago a eliminar. - Se debe presionar el botón Eliminar. - Se debe presionar el botón Aceptar.
		EP 1.2: Cancelar operación.	<ul style="list-style-type: none"> - Se debe seleccionar el concepto de pago. - Se debe presionar el botón Eliminar. - Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar concepto de pago correctamente.	Muestra un mensaje: "El concepto de pago se ha eliminado satisfactoriamente."	
EP 1.2	Cancelar operación	Se cancela la operación y se cierra la ventana.	

Anexo 15

Diseño de caso de prueba del requisito Adicionar tipos de nómina

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Gestionar tipos de nómina.
- Se debe seleccionar la opción Gestionar tipos de nómina.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Adicionar tipos de nómina.	Se adiciona un tipo de nómina.	EP 1.1: Adicionar tipo de nómina introduciendo los datos correctamente.	<ul style="list-style-type: none">– Se debe presionar el botón Adicionar.– Se debe introducir los datos del tipo de nómina en el formulario.– Se debe presionar el botón Aceptar.
		EP 1.2: Adicionar tipo de nómina dejando campos requeridos en blanco.	<ul style="list-style-type: none">– Se debe presionar el botón Adicionar.– Se debe introducir los datos del tipo de nómina en el formulario dejando al menos un campo requerido en blanco.– Se debe presionar el botón Aceptar.

		EP 1.3: Adicionar tipo de nómina introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se debe presionar el botón Adicionar. - Se debe introducir los datos del tipo de nómina en el formulario introduciendo algunos datos inválidos. - Se debe presionar el botón Aceptar.
		EP 1.4: Cancelar operación.	<ul style="list-style-type: none"> - Se debe presionar el botón Adicionar. - Se debe introducir los datos del tipo de nómina o no en el formulario. - Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Código.	Campo de texto.	NO	Combinación de letras y números.
2	Procesar.	Campo de selección.	NO	Selección.
3	Denominación.	Campo de texto.	NO	Combinación de letras y números.
4	Descripción.	Campo de texto.	NO	Combinación de letras y números.

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Código	Procesar	Denominación	Descripción	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar tipo de nómina introduciendo los datos correctamente.	V(1a)	V(Seleccionar valores)	V(dada2)	V(pago de trabajadores)	Muestra un mensaje: "El tipo de nómina se creo satisfactoriamente."	
EP 1.2	Adicionar tipo de nómina dejando campos requeridos en blanco.	I(Vacío)	V(Seleccionar valores)	V(dada2)	V(pago de trabajadores)	Se muestra un mensaje de error: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)."	
		V(1a)	I(Vacío)	V(dada2)	V(pago de trabajadores)		
		V(1a)	V(Seleccionar valores)	I(Vacío)	V(pago de trabajadores)		
		V(1a)	V(Seleccionar valores)	V(dada2)	I(Vacío)		
EP 1.3	Adicionar tipo de nómina introduciendo errores en los datos.	I(##\$%)	V(Seleccionar valores)	V(dada2)	V(pago de trabajadores)	Se muestra un mensaje de error: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)."	
		V(1a)	V(Seleccionar valores)	I(@#%)	V(pago de trabajadores)		
		V(1a)	V(Seleccionar valores)	V(dada2)	I(@a%^&)		
		V(1a)	I(##\$%)	V(dada2)	V(pago de trabajadores)		

		V(1a)	V(Seleccionar valores)	V(dada2)	V(pago de trabajadores)	de ningún dato.	
		V(1a)	V(Seleccionar valores)	V(dada2)	V(pago de trabajadores)		
EP 1.4	Cancelar operación	NA	NA	NA	NA	Se cancela la operación y se cierra la ventana.	

Diseño de caso de prueba del requisito Modificar tipos de nómina

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Gestionar tipos de nómina.
- Se debe seleccionar la opción Gestionar tipos de nómina.
- Debe de existir al menos un tipo de nómina en el sistema.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Modificar tipos de nómina.	Se modifica un tipo de nómina.	EP 1.1: Modificar tipo de nómina introduciendo los datos correctamente.	<ul style="list-style-type: none"> – Se debe presionar el botón Modificar. – Se debe introducir los datos del tipo de nómina en el formulario. – Se debe presionar el botón Aceptar.

		<p>EP 1.2: Modificar tipo de nómina dejando campos requeridos en blanco.</p>	<ul style="list-style-type: none"> – Se debe presionar el botón Modificar. – Se debe introducir los datos del tipo de nómina en el formulario dejando al menos un campo requerido en blanco. – Se debe presionar el botón Aceptar.
		<p>EP 1.3: Modificar tipo de nómina introduciendo errores en los datos.</p>	<ul style="list-style-type: none"> – Se debe presionar el botón Modificar. – Se debe introducir los datos del tipo de nómina en el formulario introduciendo algunos datos inválidos. – Se debe presionar el botón Aceptar.
		<p>EP 1.4: Cancelar operación.</p>	<ul style="list-style-type: none"> – Se debe presionar el botón Modificar. – Se debe introducir los datos del tipo de nómina o no en el formulario. – Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Código.	Campo de texto.	NO	Combinación de letras y números.
2	Procesar.	Campo de selección.	NO	Selección.
3	Denominación.	Campo de texto.	NO	Combinación de letras y números.
4	Descripción.	Campo de texto.	NO	Combinación de letras y números.

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Código	Procesar	Denominación	Descripción	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar tipo de nómina introduciendo los datos correctamente.	V(1a)	V(Seleccionar valores)	V(dada2)	V(pago de trabajadores)	Muestra un mensaje: "El tipo de nómina se modifico satisfactoriamente."	
EP 1.2	Modificar tipo de nómina dejando campos requeridos	I(Vacío)	V(Seleccionar valores)	V(dada2)	V(pago de trabajadores)	Se muestra un mensaje de error: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)."	
		V(1a)	I(Vacío)	V(dada2)	V(pago de trabajadores)		
		V(1a)	V(Seleccionar valores)	I(Vacío)	V(pago de trabajadores)		

	en blanco.	V(1a)	V(Seleccionar valores)	V(dada2)	I(Vacío)		
EP 1.3	Modificar tipo de nómina introduciendo errores en los datos.	I(##\$%)	V(Seleccionar valores)	V(dada2)	V(pago de trabajadores)	Se muestra un mensaje de error: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s).".	
		V(1a)	V(Seleccionar valores)	I(@#%)	V(pago de trabajadores)		
		V(1a)	V(Seleccionar valores)	V(dada2)	I(@a%^&)		
		V(1a)	I(##\$%)	V(dada2)	V(pago de trabajadores)	El sistema no permite la entrada de ningún dato.	
		V(1a)	V(Seleccionar valores)	V(dada2)	V(pago de trabajadores)		
		V(1a)	V(Seleccionar valores)	V(dada2)	V(pago de trabajadores)		
EP 1.4	Cancelar operación	NA	NA	NA	NA	Se cancela la operación y se cierra la ventana.	

Diseño de caso de prueba del requisito Eliminar tipos de nómina

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.

- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Gestionar tipos de nómina.
- Se debe seleccionar la opción Gestionar tipos de nómina.
- Debe de existir por lo menos un tipo de nómina en el sistema.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Eliminar tipos de nómina.	Se elimina un tipo de nómina.	EP 1.1: Eliminar tipo de nómina correctamente.	<ul style="list-style-type: none"> – Se debe seleccionar el tipo de nómina a eliminar. – Se debe presionar el botón Eliminar. – Se debe presionar el botón Aceptar.
		EP 1.2: Cancelar operación.	<ul style="list-style-type: none"> – Se debe seleccionar el tipo de nómina a eliminar. – Se debe presionar el botón Eliminar. – Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar tipo de nómina correctamente.	Muestra un mensaje: “El tipo de nómina se ha eliminado satisfactoriamente.”.	
EP 1.2	Cancelar operación	Se cancela la operación y se cierra la ventana.	

Anexo 16

Diseño de caso de prueba del requisito Adicionar período de pago

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar el menú inicio.
- Se debe seleccionar el menú Sub-sistemas.
- Se debe seleccionar el menú Capital Humano.
- Se debe seleccionar el menú Remuneración y nómina.
- Se debe seleccionar el menú Gestionar periodo de pago.
- Se abre la pestaña de funcionalidades (>>).
- Se debe seleccionar la opción Gestionar período de pago.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
----------------------	---------------------	-----------------------	---------------------

1. Adicionar período de pago.	Se adiciona un periodo de pago.	EP 1.1: Adicionar un período de pago introduciendo los datos correctamente.	<ul style="list-style-type: none"> - Se debe seleccionar la opción Adicionar. - Se deben introducir los datos del período de pago correctamente en el formulario. - Se debe presionar el botón Aplicar o Aceptar.
		EP 1.2: Adicionar un período de pago dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se debe seleccionar la opción Adicionar. - Se deben introducir los datos del período de pago en el formulario dejando al menos un campo requerido en blanco. - Se debe presionar el botón Aplicar o Aceptar.
		EP 1.3: Adicionar un período de pago introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se debe seleccionar la opción Adicionar. - Se deben introducir los datos del período de pago introduciendo errores en los datos. - Se debe presionar el botón Aplicar o Aceptar.

		EP 1.4: Cancelar operación.	<ul style="list-style-type: none"> - Se debe seleccionar la opción Adicionar. - Se deben introducir o no los datos a adicionar. - Se debe presionar el botón Cancelar.
--	--	-----------------------------	---

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Fecha inicio.	Campo de fecha.	NO	Selección.
2	Fecha fin.	Campo de fecha.	NO	Selección.
3	Programa a aplicar.	Lista desplegable.	NO	Selección.
4	Denominación.	Campo de texto.	NO	Combinación de letras.
5	Frecuencia de pago.	Lista desplegable.	NO	Selección.
6	Período de pago anterior.	Lista desplegable.	NO	Selección.

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Fecha Inicio	Fecha fin	Programa a aplicar	Denominación	Frecuencia de pago	Período de pago anterior	Respuesta del sistema	Resultado de la prueba
------------------	-----------	--------------	-----------	--------------------	--------------	--------------------	--------------------------	-----------------------	------------------------

EP 1.1	Adicionar un período de pago introduciendo los datos correctamente.	V(12/12/2008)	V(04/1/2009)	V(erp)	V(Denominación)	V(15 días)	V(2da quincena)	Se muestra un mensaje: “El periodo de pago ha sido adicionado satisfactoriamente.”	
EP 1.2	Adicionar un período de pago dejando campos requeridos en blanco.	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	Se muestra un mensaje: “Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s).”.	
		V(12/12/2008)	I(Vacío)	V(erp)	V(Denominación)	V(15 días)	V(2da quincena)		
		V(12/12/2008)	V(04/1/2009)	I(Vacío)	V(Denominación)	V(15 días)	V(2da quincena)		
		V(12/12/2008)	V(04/1/2009)	V(erp)	I(Vacío)	V(15 días)	V(2da quincena)		
		V(12/12/2008)	V(04/1/2009)	V(erp)	V(Denominación)	I(Vacío)	V(2da quincena)		
		V(12/12/2008)	V(04/1/2009)	V(erp)	V(Denominación)	V(15 días)	I(Vacío)		
		V(12/12/2008)	V(04/1/2009)	V(erp)	V(Denominación)	V(15 días)	V(2da quincena)		

EP 1.3	Adicionar un período de pago introduciendo errores en los datos.	V(12/12/2008)	V(04/1/2009)	V(erp)	I(/////)&_)	I(ghg)	I(dddsf)	Se muestra un mensaje: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)".
		V(12/12/2008)	V(04/1/2009)	V(erp)	I(89898)	V(15 días)	V(12)	
		V(12/12/2008)	V(04/1/2009)	V(erp)	V(Denominación)	I(wew)	V(12)	
		V(12/12/2008)	V(04/1/2009)	V(erp)	V(Denominación)	I(/*-)	V(12)	
		V(12/12/2008)	V(04/1/2009)	V(erp)	V(Denominación)	V(15 días)	I(/*-)	
		V(12/12/2008)	V(04/1/2009)	V(erp)	V(Denominación)	V(15 días)	V(2da quincena)	
EP 1.4	Cancelar operación.	NA	NA	NA	NA	NA	NA	Se cancela la operación y se cierra la ventana.

Diseño de caso de prueba Modificar período de pago

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar el menú inicio.
- Se debe seleccionar el menú Sub-sistemas.
- Se debe seleccionar el menú Capital Humano.

- Se debe seleccionar el menú Remuneración y nómina.
- Se debe seleccionar el menú Gestionar periodo de pago.
- Se abre la pestaña de funcionalidades (>>).
- Se debe seleccionar la opción Gestionar período de pago.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Modificar período de pago.	Se modifica un periodo de pago.	EP 1.1: Modificar un período de pago introduciendo los datos correctamente.	<ul style="list-style-type: none"> – Se debe seleccionar la opción Modificar. – Se deben introducir los datos del período de pago correctamente en el formulario. – Se debe presionar el botón Aplicar o Aceptar.
		EP 1.2: Modificar un período de pago dejando campos requeridos en blanco.	<ul style="list-style-type: none"> – Se debe seleccionar la opción Modificar. – Se deben introducir los datos del período de pago en el formulario dejando al menos un campo requerido en blanco. – Se debe presionar el

		EP 1.3: Modificar un período de pago introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se debe seleccionar la opción Modificar. - Se deben introducir los datos del período de pago introduciendo errores en los datos. - Se debe presionar el botón Aplicar o Aceptar.
		EP 1.4: Cancelar operación.	<ul style="list-style-type: none"> - Se debe seleccionar la opción Modificar. - Se deben introducir o no los datos a adicionar. - Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Fecha inicio.	Campo de fecha.	NO	Selección.
2	Fecha fin.	Campo de fecha.	NO	Selección.
3	Programa a aplicar.	Lista desplegable.	NO	Selección.
4	Denominación.	Campo de texto.	NO	Combinación de letras.
5	Frecuencia de pago.	Lista desplegable.	NO	Selección.
6	Período de pago anterior.	Lista desplegable.	NO	Selección.

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Fecha Inicio	Fecha fin	Programa a aplicar	Denominación	Frecuencia de pago	Período de pago anterior	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar un período de pago introduciendo los datos correctamente.	V(12/12/2008)	V(04/1/2009)	V(erp)	V(Denominación)	V(15 días)	V(2da quincena)	Se muestra un mensaje: "El periodo de pago ha sido modificado satisfactoriamente."	
EP 1.2	Modificar un período de pago dejando campos requeridos en blanco.	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	Se muestra un mensaje: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s).".	
		V(12/12/2008)	I(Vacío)	V(erp)	V(Denominación)	V(15 días)	V(2da quincena)		
		V(12/12/2008)	V(04/1/2009)	I(Vacío)	V(Denominación)	V(15 días)	V(2da quincena)		
		V(12/12/2008)	V(04/1/2009)	V(erp)	I(Vacío)	V(15 días)	V(2da quincena)		
		V(12/12/2008)	V(04/1/2009)	V(erp)	V(Denominación)	I(Vacío)	V(2da quincena)		

		V(12/1 2/2008)	V(04/1/ 2009)	V(erp)	V(Denomin ación)	V(15 días)	I(Vacío)		
		V(12/1 2/2008)	V(04/1/ 2009)	V(erp)	V(Denomin ación)	V(15 días)	V(2da quincen a)		
EP 1.3	Modificar un período de pago introducien do errores en los datos.	V(12/1 2/2008)	V(04/1/ 2009)	V(erp)	I(/////)&_)	I(ghg)	I(dddsf)	Se muestra un mensaje: “Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s).”.	
		V(12/1 2/2008)	V(04/1/ 2009)	V(erp)	I(89898)	V(15 días)	V(2da quincen a)		
		V(12/1 2/2008)	V(04/1/ 2009)	V(erp)	V(Denomin ación)	I(wew)	V(2da quincen a)		
		V(12/1 2/2008)	V(04/1/ 2009)	V(erp)	V(Denomin ación)	I(/*-)	V(2da quincen a)		
		V(12/1 2/2008)	V(04/1/ 2009)	V(erp)	V(Denomin ación)	V(15 días)	I(/*-)		
		V(12/1 2/2008)	V(04/1/ 2009)	V(erp)	V(Denomin ación)	V(15 días)	V(2da quincen a)		
EP 1.4	Cancelar operación.	NA	NA	NA	NA	NA	NA	Se cancela la operación y se cierra la ventana.	

Diseño de caso de prueba del requisito Eliminar período de pago

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar el menú inicio.
- Se debe seleccionar el menú Sub-sistemas.
- Se debe seleccionar el menú Capital Humano.
- Se debe seleccionar el menú Remuneración y nómina.
- Se debe seleccionar el menú Gestionar período de pago.
- Se abre la pestaña de funcionalidades (>>).
- Se debe seleccionar la opción Gestionar período de pago.
- Debe de existir al menos un período de pago.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Eliminar período de pago.	Se elimina un período de pago.	EP 1.1: Eliminar un período de pago correctamente.	– Se debe seleccionar el período de pago a eliminar. – Se debe presionar el botón Eliminar . – Se debe presionar el botón Aceptar .
		EP 1.2: Cancelar operación.	– Se debe seleccionar el período de pago a eliminar. – Se debe presionar el botón Eliminar . – Se debe presionar el botón Cancelar .

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Se elimina un período de pago correctamente.	Se muestra un mensaje: "El período de pago ha sido eliminado satisfactoriamente."	
EP 1.2	Cancelar operación.	Se muestra un mensaje: "La operación ha sido cancelada."	

Diseño de caso de prueba del requisito Generar período de pago

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar el menú inicio.
- Se debe seleccionar el menú Sub-sistemas.
- Se debe seleccionar el menú Capital Humano.
- Se debe seleccionar el menú Remuneración y nómina.
- Se debe seleccionar el menú Gestionar periodo de pago.
- Se abre la pestaña de funcionalidades (>>).
- Se debe seleccionar la opción Gestionar período de pago.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
----------------------	---------------------	-----------------------	---------------------

1. Generar período de pago.	Se genera un periodo de pago.	EP 1.1: Generar un período de pago introduciendo los datos correctamente.	<ul style="list-style-type: none"> – Se debe seleccionar la opción Generar. – Se deben introducir los datos del período de pago correctamente en el formulario. – Se debe presionar el botón Aplicar o Aceptar.
		EP 1.2: Generar un período de pago dejando campos requeridos en blanco.	<ul style="list-style-type: none"> – Se debe seleccionar la opción Generar. – Se deben introducir los datos del período de pago en el formulario dejando al menos un campo requerido en blanco. – Se debe presionar el botón Aplicar o Aceptar.
		EP 1.3: Generar un período de pago introduciendo errores en los datos.	<ul style="list-style-type: none"> – Se debe seleccionar la opción Generar. – Se deben introducir los datos del período de pago introduciendo errores en los datos. – Se debe presionar el botón Aplicar o Aceptar.

		EP 1.4: Cancelar operación.	<ul style="list-style-type: none"> - Se debe seleccionar la opción Generar. - Se deben introducir o no los datos a adicionar. - Se debe presionar el botón Cancelar.
--	--	-----------------------------	---

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Mensual.	Campo de selección.	SI	Selección.
2	Quincenal.	Campo de selección.	SI	Selección.
3	Año.	Campo de texto.	NO	Combinación de números.
4	Programa primera quincena.	Lista desplegable.	NO	Selección.
5	Programa segunda quincena.	Lista desplegable.	NO	Selección.

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Mensual	Quincenal	Año	Programa primera quincena	Programa segunda quincena	Respuesta del sistema	Resultado de la prueba
EP 1.1	Generar un período de pago introduciendo	V(SI)	V(Vacío)	V(2008)	V(programa)	V(programa)	Se muestra un mensaje: "El periodo de pago ha sido	

	do los datos correctame	V(Vacío)	V(SI)	V(2008)	V(programa)	V(programa)	generado satisfactoriamente.	
EP 1.2	Generar un período de pago dejando campos requeridos en blanco.	V(Vacío)	V(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	Se muestra un mensaje: “Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s).”.	
		V(SI)	V(Vacío)	I(Vacío)	V(programa)	V(programa)		
		V(SI)	V(Vacío)	V(2008)	I(Vacío)	V(programa)		
		V(SI)	V(Vacío)	V(2008)	V(programa)	I(Vacío)		
EP 1.3	Generar un período de pago introduciendo errores en los datos.	V(SI)	V(Vacío)	I(%\$)	V(programa)	V(programa)	Se muestra un mensaje: “Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s).”.	
		V(SI)	V(Vacío)	V(2008)	I(\$%&)	V(programa)		
		V(SI)	V(Vacío)	V(2008)	V(programa)	I(\$%&)		
EP 1.4	Cancelar operación.	NA	NA	NA	NA	NA	Se cancela la operación y se cierra la ventana.	

Anexo 17

Diseño de caso de prueba del requisito Adicionar tipo de ajuste

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar el menú inicio.
- Se debe seleccionar el menú Sub-sistemas.

- Se debe seleccionar el menú Capital Humano.
- Se debe seleccionar el menú Remuneración-Nómina.
- Se debe seleccionar el menú Gestionar ajustes.
- Se abre la pestaña de funcionalidades (>>).
- Se debe seleccionar la opción Gestionar tipo de ajuste.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Adicionar tipo de ajuste.	Se adiciona un tipo de ajuste.	EP 1.1 Adicionar un tipo de ajuste introduciendo los datos correctamente.	<ul style="list-style-type: none"> – Se debe seleccionar la opción Adicionar. – Se debe introducir los datos del tipo de ajuste correctamente en el formulario. – Se debe presionar el botón Aceptar.
		EP 1.2: Adicionar un tipo de ajuste dejando campos requeridos en blanco.	<ul style="list-style-type: none"> – Se debe seleccionar la opción Adicionar. – Se debe introducir los datos del tipo de ajuste en el formulario dejando al menos un campo requerido en blanco. – Se debe presionar el botón Aceptar.

		EP 1.3: Adicionar un tipo de ajuste introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se debe seleccionar la opción Adicionar. - Se debe introducir los datos del tipo de ajuste introduciendo errores en los datos. - Se debe presionar el botón Aceptar.
		EP 1.4: Cancelar operación.	<ul style="list-style-type: none"> - Se debe seleccionar la opción Adicionar. - Se debe introducir o no los datos a adicionar. - Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Denominación.	Campo de texto.	NO	Combinación de letras.
2	Descripción.	Campo de texto.	NO	Combinación de letras y números.
3	Código.	Campo de texto.	NO	Combinación de números y letras.

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Denominación	Descripción	Código	Respuesta del sistema	Resultado de la prueba

EP 1.1	Adicionar un tipo de ajuste introduciendo los datos correctamente.	V(Denominación)	V(descripción)	V(32a)	Se muestra un mensaje: “El tipo de ajuste ha sido adicionado satisfactoriamente.”	
EP 1.2	Adicionar un tipo de ajuste dejando	I(Vacío)	V(descripción)	V(32a)	Se muestra un mensaje: “Complete los campos requeridos.”	
		V(Denominación)	I(Vacío)	I(Vacío)		
EP 1.3	Adicionar un tipo de ajuste introduciendo errores en los	I(*-+=&)	V(descripción)	V(32)	Se muestra un mensaje: “Existen errores en los datos.”	
		V(Denominación)	I(#\$@#)	V(32)		
		V(Denominación)	V(descripción)	I(\$%^)		
EP 1.4	Cancelar operación.	NA	NA	NA	Se muestra un mensaje: “La	

Diseño de caso de prueba del requisito Modificar tipo de ajuste

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar el menú inicio.
- Se debe seleccionar el menú Sub-sistemas.
- Se debe seleccionar el menú Capital Humano.
- Se debe seleccionar el menú Remuneración-Nómina.
- Se debe seleccionar el menú Gestionar ajustes.

- Se abre la pestaña de funcionalidades (>>).
- Se debe seleccionar la opción Gestionar tipo de ajuste.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Modificar tipo de ajuste.	Se modifica un tipo de ajuste.	EP 1.1 Modificar un tipo de ajuste introduciendo los datos correctamente.	<ul style="list-style-type: none"> – Se debe seleccionar la opción Modificar. – Se debe introducir los datos del tipo de ajuste correctamente en el formulario. – Se debe presionar el botón Aceptar.
		EP 1.2: Modificar un tipo de ajuste dejando campos requeridos en blanco.	<ul style="list-style-type: none"> – Se debe seleccionar la opción Modificar. – Se debe introducir los datos del tipo de ajuste en el formulario dejando al menos un campo requerido en blanco. – Se debe presionar el botón Aceptar.
		EP 1.3: Modificar un tipo de ajuste introduciendo errores en los datos.	<ul style="list-style-type: none"> – Se debe seleccionar la opción Modificar. – Se debe introducir los datos del tipo de ajuste introduciendo errores en los datos. – Se debe presionar el

			botón Aceptar .
		EP 1.4: Cancelar operación.	<ul style="list-style-type: none"> – Se debe seleccionar la opción Modificar. – Se debe introducir o no los datos a adicionar. – Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Denominación.	Campo de texto.	NO	Combinación de letras.
2	Descripción.	Campo de texto.	NO	Combinación de letras y números.
3	Código.	Campo de texto.	NO	Combinación de números y letras.

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Denominación	Descripción	Código	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar un tipo de ajuste introduciendo los datos correctamente	V(Denominación)	V(descripción)	V(32a)	Se muestra un mensaje: "El tipo de ajuste ha sido modificado satisfactoriamente."	
EP	Modificar un	I(Vacío)	V(descripción)	V(32a)	Se muestra un mensaje:	

1.2	tipo de ajuste dejando	V(Denominación)	I(Vacío)	I(Vacío)	“Complete los campos requeridos.”	
EP 1.3	Modificar un tipo de ajuste introduciendo errores en los datos.	I(*-+=&)	V(descripción)	V(32)	Se muestra un mensaje: “Existen errores en los datos.”	
		V(Denominación)	I(#\$@#)	V(32)		
		V(Denominación)	V(descripción)	I(\$%^)		
EP 1.4	Cancelar operación.	NA	NA	NA	Se muestra un mensaje: “La operación ha sido cancelada.”.	

Diseño de caso de prueba Eliminar tipo de ajuste

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar el menú inicio.
- Se debe seleccionar el menú Sub-sistemas.
- Se debe seleccionar el menú Capital Humano.
- Se debe seleccionar el menú Remuneración-Nómina.
- Se debe seleccionar el menú Gestionar ajustes.
- Se abre la pestaña de funcionalidades (>>).
- Se debe seleccionar la opción Gestionar tipo de ajuste.
- Debe de existir al menos un tipo de ajuste.
- Se debe seleccionar la opción **Eliminar**.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
----------------------	---------------------	-----------------------	---------------------

1. Eliminar un tipo de ajuste.	Se elimina un tipo de ajuste.	EP 1.1: Eliminar un tipo de ajuste correctamente.	<ul style="list-style-type: none"> – Se debe seleccionar el tipo de ajuste a eliminar. – Se debe presionar botón Eliminar. – Se debe presionar el botón Aceptar
		EP 1.2: Cancelar la operación.	<ul style="list-style-type: none"> – Se debe presionar el botón Eliminar. – Se debe presionar el botón Cancelar. – Se debe presionar el botón Aceptar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar un tipo de ajuste correctamente.	Se muestra un mensaje: "El tipo de ajuste ha sido eliminado satisfactoriamente."	
EP 1.2	Cancelar la operación.	Se muestra un mensaje: "La operación ha sido cancelada."	

Anexo 18

Diseño de caso de prueba del requisito Modificar nómina

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Administrar nómina.
- Se debe seleccionar la opción Administrar nómina.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Modificar nómina.	Se modifica una nómina.	EP 1.1: Modificar nómina introduciendo los datos correctamente.	<ul style="list-style-type: none">– Se debe presionar el botón Modificar.– Se debe introducir los datos de la nómina en el formulario.– Se debe presionar el botón Aceptar.
		EP 1.2: Modificar nómina dejando campos requeridos en blanco.	<ul style="list-style-type: none">– Se debe presionar el botón Modificar.– Se debe introducir los datos de la nómina en el formulario dejando al menos un campo requerido en blanco.– Se debe presionar el botón Aceptar.

		EP 1.3: Modificar nómina introduciendo errores en los datos.	<ul style="list-style-type: none"> – Se debe presionar el botón Modificar. – Se debe introducir los datos de la nómina en el formulario introduciendo algunos datos inválidos. – Se debe presionar el botón Aceptar.
		EP 1.4: Cancelar operación.	<ul style="list-style-type: none"> – Se debe presionar el botón Modificar. – Se debe introducir los datos de la nómina o no en el formulario. – Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Denominación.	Campo de texto.	NO	Combinación de letras y números.
2	Tipo nómina.	Lista desplegable.	NO	Selección.
3	Período pago.	Lista desplegable.	NO	Selección.
4	Fecha.	Campo de fecha.	NO	Selección.
5	Código.	Campo de texto.	NO	Combinación de letras y números.

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Denominación	Tipo nómina	Período pago	Fecha	Código	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar nómina introduciendo los datos correctamente.	V(Enero)	V(Subsidio)	V(Enero)	V(12/12/2008)	V(1a)	Muestra un mensaje: "La nómina se modifico satisfactoriamente."	
EP 1.2	Modificar nómina dejando campos requeridos en blanco.	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	Se muestra un mensaje de error: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)."	
		I(Vacío)	V(Subsidio)	V(Enero)	V(12/12/2008)	V(1a)		
		V(Enero)	I(Vacío)	V(Enero)	V(12/12/2008)	V(1a)		
		V(Enero)	V(Subsidio)	I(Vacío)	V(12/12/2008)	V(1a)		
		V(Enero)	V(Subsidio)	V(Enero)	I(Vacío)	V(1a)		
		V(Enero)	V(Subsidio)	V(Enero)	V(12/12/2008)	I(Vacío)		

EP 1.3	Modificar nómina introduciendo errores en los datos.	I(E#\$\$)	V(Vacaciones)	V(Febrero)	V(12/05/2008)	V(1a)	Se muestra un mensaje de error: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s).".	
		V(Enero)	I(jshd)	V(Enero)	V(12/05/2008)	V(1a)	El sistema no permite la entrada de ningún dato.	
		V(Enero)	V(Subsidio)	I(sadas)	V(12/05/2008)	V(1a)		
		V(Enero)	V(Subsidio)	V(Enero)	I(werwetr)	V(1a)		
		V(Enero)	V(Subsidio)	V(Enero)	V(12/05/2008)	I(##\$%)		
EP 1.4	Cancelar operación	NA	NA	NA	NA	NA	Se cancela la operación y se cierra la ventana.	

Diseño de caso de prueba Eliminar nómina

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.

- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Administrar nómina.
- Se debe seleccionar la opción Administrar nómina.
- Debe existir al menos una nómina.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Eliminar nómina.	Se elimina una nómina.	EP 1.1: Eliminar nómina correctamente.	<ul style="list-style-type: none"> – Se debe seleccionar la nómina a eliminar. – Se debe presionar el botón Eliminar. – Se debe presionar el botón Aceptar.
		EP 1.2: Cancelar operación.	<ul style="list-style-type: none"> – Se debe seleccionar la nómina a eliminar. – Se debe presionar el botón Eliminar. – Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

1.1.2 Juegos de datos a probar

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

EP 1.1	Eliminar nómina correctamente.	Muestra un mensaje: “La nómina se ha eliminado satisfactoriamente.”.	
EP 1.2	Cancelar operación	Se cancela la operación y se cierra la ventana.	

Diseño de caso de prueba del requisito Eliminar trabajadores a la nómina

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Administrar nómina.
- Se debe seleccionar la opción Administrar nómina.
- Debe existir al menos un trabajador adicionado a una nómina.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Eliminar trabajadores de la nómina.	Se elimina un trabajador a la nómina.	EP 1.1: Eliminar trabajadores a la nómina correctamente.	<ul style="list-style-type: none"> – Se debe seleccionar el trabajador a eliminar de la nómina. – Se debe presionar el botón Eliminar. – Se debe presionar el botón Aceptar.

		EP 1.2: Cancelar operación.	<ul style="list-style-type: none"> - Se debe seleccionar el trabajador a eliminar de la nómina. - Se debe presionar el botón Eliminar. - Se debe presionar el botón Cancelar.
--	--	-----------------------------	--

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar trabajadores de la nómina correctamente.	Muestra un mensaje: "El trabajador de la ha sido eliminado satisfactoriamente."	
EP 1.2	Cancelar operación	Se cancela la operación y se cierra la ventana.	

Diseño de caso de prueba Filtrar por período

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.

- Se debe seleccionar Administrar nómina.
- Se debe seleccionar la opción Administrar nómina.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Filtrar por período.	Se filtran las nóminas por el período de pago.	EP 1.1: Filtrar las nóminas por período.	– Se debe seleccionar el período de pago.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Período de pago.	Lista desplegable.	NO	Selección.

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Filtrar las nóminas por período.	Muestra un mensaje: “Las nóminas están filtradas por el período de pago seleccionado.”.	

Diseño de caso de prueba del requisito Mostrar detalles del trabajador de una nómina

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Revisar nómina.
- Se debe seleccionar la opción Revisar nómina.

- Debe existir al menos una nómina.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Mostrar detalles del trabajador de una nómina.	Se muestran los detalles del trabajador de una nómina.	EP 1.1: Mostrar detalles del trabajador de una nómina.	<ul style="list-style-type: none"> – Se debe seleccionar la nómina. – Se debe seleccionar el trabajador. – Se debe presionar el botón Detalles.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Mostrar detalles del trabajador de una nómina.	Se muestra en un grid los datos del trabajador en la nómina.	

Diseño de caso de prueba del requisito Mostrar comprobante de operaciones de la nómina

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Revisar nómina.

- Se debe seleccionar la opción Revisar nómina.
- Debe existir al menos una nómina.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Mostrar comprobante de operaciones de la nómina.	Se muestra el comprobante de operaciones de la nómina.	EP 1.1: Mostrar comprobante de operaciones de la nómina.	<ul style="list-style-type: none"> – Se debe seleccionar la nómina. – Se debe presionar el botón Mostrar comprobante.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Mostrar comprobante de operaciones de la nómina.	Se muestra en un grid los datos del comprobante de operaciones de la nómina.	

Diseño de caso de prueba del requisito Ajustar nómina

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.

- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Revisar nómina.
- Se debe seleccionar la opción Revisar nómina.
- Debe existir al menos una nómina.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Ajustar nómina.	Se ajusta la nómina de un trabajador.	EP 1.1: Ajustar nómina a un trabajador correctamente.	<ul style="list-style-type: none"> – Se debe seleccionar el la nómina a ajustar. – Se debe seleccionar el trabajador de la nómina a ajustar. – Se debe presionar el botón Detalles. – Se selecciona el identificador a ajustar. – Se debe presionar el botón Ajuste. – Se deben introducir los datos correctamente. – Se debe presionar el botón Aceptar.

		<p>EP 1.2: Ajustar nómina a un trabajador dejando campos obligatorios en blanco.</p>	<ul style="list-style-type: none"> – Se debe seleccionar el la nómina a ajustar. – Se debe seleccionar el trabajador de la nómina a ajustar. – Se debe presionar el botón Detalles. – Se selecciona el identificador a ajustar. – Se debe presionar el botón Ajuste. – Se deben introducir los datos dejando algún campo requerido en blanco. – Se debe presionar el botón Aceptar.
--	--	--	---

		<p>EP 1.3: Ajustar nómina a un trabajador introduciendo errores en los datos.</p>	<ul style="list-style-type: none"> – Se debe seleccionar el la nómina a ajustar. – Se debe seleccionar el trabajador de la nómina a ajustar. – Se debe presionar el botón Detalles. – Se selecciona el identificador a ajustar. – Se debe presionar el botón Ajuste. – Se deben introducir los datos introduciendo errores en algún campo. – Se debe presionar el botón Aceptar.
--	--	---	--

		EP 1.4: Cancelar.	<ul style="list-style-type: none"> - Se debe seleccionar el la nómina a ajustar. - Se debe seleccionar el trabajador de la nómina a ajustar. - Se debe presionar el botón Detalles. - Se selecciona el identificador a ajustar. - Se debe presionar el botón Ajuste. - Se deben introducir o no los datos. - Se debe presionar el botón Aceptar.
--	--	-------------------	--

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Tipo de ajuste.	Lista desplegable.	NO	Selección.
2	Valor anterior.	Campo de texto.	NO	Combinación de números.
3	Nuevo valor.	Campo de texto.	NO	Combinación de números.

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Tipo de ajuste	Valor anterior	Nuevo valor	Respuesta del sistema	Resultado de la prueba
------------------	-----------	----------------	----------------	-------------	-----------------------	------------------------

EP 1.1	Adicionar un documento ajuste a un trabajador en la nómina correctamente.	V(Ajuste de salario)	V(12)	V(120)	Muestra un mensaje: "La nómina se ha ajustado satisfactoriamente."	
		V(Ajuste de salario)	V(12)	V(120)		
EP 1.2	Ajustar nómina a un trabajador dejando campos obligatorios en blanco.	I(Vacío)	V(12)	V(120)	Se muestra un mensaje de error.	
		V(Ajuste de salario)	I(Vacío)	V(120)		
		V(Ajuste de salario)	V(12)	I(Vacío)		
EP 1.3	Ajustar nómina a un trabajador introduciendo errores en los datos.	V(Ajuste de salario)	I(wewe)	V(120)	Se muestra un mensaje de error.	
		V(Ajuste de salario)	I(-/*-)	V(120)		
		V(Ajuste de salario)	V(12)	I(e3)		
		V(Ajuste de salario)	V(12)	I(/*-+)		
		I(%\$^)	V(12)	V(120)	El sistema no permite la entrada de ningún dato.	
EP 1.4	Cancelar.	NA	NA	NA	Se cancela la operación y se cierra la ventana.	

Diseño de caso de prueba del requisitos

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Revisar nómina.
- Se debe seleccionar la opción Revisar nómina.
- Debe existir al menos una nómina.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Revertir nómina.	Se revierte la nómina.	EP 1.1: Revertir la nómina.	<ul style="list-style-type: none"> – Se debe seleccionar la nómina a revertir. – Se debe presionar el botón Revertir.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Revertir la nómina.	El sistema muestra un mensaje: "La nómina ha sido revertida."	

Diseño de caso de prueba del requisito Confirmar nómina

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Revisar nómina.
- Se debe seleccionar la opción Revisar nómina.
- Debe existir al menos una nómina.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Confirmar nómina.	Se confirma la nómina.	EP 1.1: Confirmar una nómina.	<ul style="list-style-type: none"> – Se debe seleccionar la nómina a confirmar. – Se debe presionar el botón Confirmar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

1.1.2 Juegos de datos a probar

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

EP 1.1	Confirmar una nómina.	El sistema muestra un mensaje: "La nómina ha sido confirmada."	
--------	-----------------------	--	--

Diseño de caso de prueba del requisito Revisar comprobante de operaciones

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Comprobante de operaciones.
- Se debe seleccionar la opción Comprobante de operaciones.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Revisar de comprobante de operaciones.	Se muestran los detalles de un comprobante de operación.	EP 1.1: Mostrar detalles de un comprobante de operaciones	<ul style="list-style-type: none"> – Se debe seleccionar el comprobante. – Se debe presionar el botón Detalles.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

1.1.2 Juegos de datos a probar

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

EP 1.1	Mostrar detalles de un comprobante de operaciones.	Se muestra en un grid los datos del comprobante de operaciones.	
--------	--	---	--

Diseño de caso de prueba del requisito Emitir comprobante de operaciones

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Comprobante de operaciones.
- Se debe seleccionar la opción Comprobante de operaciones.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Emitir comprobante de operaciones.	Se emite un comprobante de operación.	EP 1.1: Emitir un comprobante de operaciones	<ul style="list-style-type: none"> – Se debe seleccionar el comprobante. – Se debe presionar el botón Emitir.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

1.1.2 Juegos de datos a probar

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

EP 1.1	Emitir un comprobante de operaciones.	El sistema muestra un mensaje: "El comprobante de operaciones ha sido emitido correctamente."	
--------	---------------------------------------	---	--

Anexo 19

Diseño de caso de prueba del requisito Efectuar cierre contable

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Cierre.
- Se debe seleccionar la opción Cierre.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Efectuar cierre contable.	Se realiza el cierre contable del período o del ejercicio.	EP 1.1: Efectuar cierre del período contable.	<ul style="list-style-type: none"> – Se debe presionar el botón Cerrar período contable. – Se debe presionar el botón Aceptar.
		EP 1.2: Efectuar cierre del ejercicio contable.	<ul style="list-style-type: none"> – Se debe presionar el botón Cerrar ejercicio contable. – Se debe presionar el botón Aceptar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Efectuar cierre del período contable.	El sistema muestra un mensaje: “Se efectuó el cierre del período contable.”.	
EP 1.1	Efectuar cierre del ejercicio contable.	El sistema muestra un mensaje: “Se efectuó el cierre del ejercicio contable.”.	

Anexo 20

Diseño de caso de prueba del requisito Establecer configuración de cuentas

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar el menú inicio.
- Se debe seleccionar el menú Sub-sistemas.
- Se debe seleccionar el menú Capital Humano.
- Se debe seleccionar el menú Remuneración-Nómina.
- Se debe seleccionar el menú Configuración.
- Se abre la pestaña de funcionalidades (>>).
- Se debe seleccionar la opción Configuración.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
----------------------	---------------------	-----------------------	---------------------

1. Establecer configuración de cuentas.	Se establece la configuración de cuentas.	EP 1.1 Establecer la configuración de cuentas correctamente.	<ul style="list-style-type: none"> – Se debe introducir los datos de la configuración en el formulario. – Se debe presionar el botón Aceptar.
		EP 1.2: Establecer la configuración de cuentas dejando campos requeridos en blanco.	<ul style="list-style-type: none"> – Se debe introducir los datos de la configuración en el formulario dejando al menos un campo requerido en blanco. – Se debe presionar el botón Aceptar.
		EP 1.3: Establecer la configuración de cuentas introduciendo errores en los datos.	<ul style="list-style-type: none"> – Se debe introducir los datos de la configuración introduciendo errores en los datos. – Se debe presionar el botón Aceptar.
		EP 1.4: Cancelar operación.	<ul style="list-style-type: none"> – Se debe introducir o no los datos de la configuración. – Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Cuenta de nóminas por pagar.	Lista desplegable.	NO	Selección.

2	Cuenta de salario básico.	Lista desplegable.	NO	Selección.
3	Cuenta de contabilización adicional.	Lista desplegable.	NO	Selección.
4	Cuenta de contabilización adicional.	Lista desplegable.	NO	Selección.
5	Cuenta de contabilización adicional.	Lista desplegable.	NO	Selección.
6	Enviar comprobante al confirmar la nómina.	Lista desplegable.	SI	Selección.

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Cuenta de nóminas por pagar	Cuenta de salario básico	Cuenta de contabilización adicional	Cuenta de contabilización adicional	Cuenta de contabilización adicional	Enviar comprobante al confirmar la nómina	Respuesta del sistema	Resultado de la prueba
EP 1.1	Establecer la configuración de cuentas introduciendo los datos correctamente.	V(cuenta1)	V(cuenta2)	V(cuenta3)	V(cuenta4)	V(cuenta5)	V(SI)	Se muestra un mensaje: "Las cuentas han sido configuradas satisfactoriamente."	
		V(cuenta1)	V(cuenta2)	V(cuenta3)	V(cuenta4)	V(cuenta5)	V(Vacío)		
EP	Establecer	I(Vacío)	V(cuenta2)	V(cuenta3)	V(cuenta4)	V(cuenta5)	V(SI)	Se muestra	

1.2	la configuración de cuentas dejando campos requeridos en blanco.	V(cuenta1)	I(Vacío)	V(cuenta3)	V(cuenta4)	V(cuenta5)	V(SI)	un mensaje: "Complete los campos requeridos."	
		V(cuenta1)	V(cuenta2)	I(Vacío)	V(cuenta4)	V(cuenta5)	V(SI)		
		V(cuenta1)	V(cuenta2)	V(cuenta3)	I(Vacío)	V(cuenta5)	V(SI)		
		V(cuenta1)	V(cuenta2)	V(cuenta3)	V(cuenta4)	I(Vacío)	V(SI)		
EP 1.3	Establecer la configuración de cuentas introduciendo errores en los datos.	I(*-+=&)	V(cuenta2)	V(cuenta3)	V(cuenta4)	V(cuenta5)	V(SI)	El sistema no permite la entrada de datos.	
		V(cuenta1)	I(#\$@#)	V(cuenta3)	V(cuenta4)	V(cuenta5)	V(SI)		
		V(cuenta1)	V(cuenta2)	I(\$%^)	V(cuenta4)	V(cuenta5)	V(SI)		
		V(cuenta1)	V(cuenta2)	V(cuenta3)	I(\$%^)	V(cuenta5)	V(SI)		
		V(cuenta1)	V(cuenta2)	V(cuenta3)	V(cuenta4)	I(\$%^)	V(SI)		
EP 1.4	Cancelar operación.	NA	NA	NA	NA	NA	NA	Se muestra un mensaje: "La operación ha sido cancelada."	

Anexo 21

Diseño de caso de prueba del requisito Adicionar frecuencia de pago

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Gestionar frecuencia de pagos.
- Se debe seleccionar la opción Gestionar frecuencia de pagos.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Adicionar frecuencia de pago.	Se adiciona una frecuencia de pago.	EP 1.1: Adicionar frecuencia de pago introduciendo los datos correctamente.	<ul style="list-style-type: none"> – Se debe presionar el botón Adicionar. – Se debe introducir los datos de la frecuencia de pago en el formulario. – Se debe presionar el botón Aceptar.
		EP 1.2: Adicionar frecuencia de pago dejando campos requeridos en blanco.	<ul style="list-style-type: none"> – Se debe presionar el botón Adicionar. – Se debe introducir los datos de la frecuencia de pago en el formulario dejando al menos un campo requerido en blanco. – Se debe presionar el botón Aceptar.

		EP 1.3: Adicionar frecuencia de pago introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se debe presionar el botón Adicionar. - Se debe introducir los datos de la frecuencia de pago en el formulario introduciendo algunos datos inválidos. - Se debe presionar el botón Aceptar.
		EP 1.4: Cancelar operación.	<ul style="list-style-type: none"> - Se debe presionar el botón Adicionar. - Se debe introducir los datos de la frecuencia de pago o no en el formulario. - Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Denominación.	Campo de texto.	NO	Combinación de letras y números.
2	Cantidad por mes.	Campo de texto.	NO	Combinación de letras y números.

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Denominación	Cantidad por mes	Respuesta del sistema	Resultado de la prueba
------------------	-----------	--------------	------------------	-----------------------	------------------------

EP 1.1	Adicionar frecuencia de pago introduciendo los datos correctamente.	V(denominación 1)	V(cantidad 1)	Muestra un mensaje: “La frecuencia de pago se creo satisfactoriamente.”.	
EP 1.2	Adicionar frecuencia de pago dejando campos requeridos en blanco.	I(Vacío)	V(cantidad 1)	Se muestra un mensaje de error: “Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s).”.	
		V(1a)	I(Vacío)		
EP 1.3	Adicionar frecuencia de pago introduciendo errores en los datos.	I (#\$%)	V(cantidad 1)	Se muestra un mensaje de error: “Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s).”.	
		V(denominación 1)	(#\$%)		
EP 1.4	Cancelar operación	NA	NA	Se cancela la operación y se cierra la ventana.	

Diseño del caso de prueba Modificar frecuencia de pago

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.
- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.

- Se debe seleccionar Gestionar frecuencia de pagos.
- Se debe seleccionar la opción Gestionar frecuencia de pagos.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Modificar frecuencia de pago.	Se modifica una frecuencia de pago.	EP 1.1: Modificar frecuencia de pago introduciendo los datos correctamente.	<ul style="list-style-type: none"> – Se debe presionar el botón Modificar. – Se debe introducir los datos de la frecuencia de pago en el formulario. – Se debe presionar el botón Aceptar.
		EP 1.2: Modificar frecuencia de pago dejando campos requeridos en blanco.	<ul style="list-style-type: none"> – Se debe presionar el botón Modificar. – Se debe introducir los datos de la frecuencia de pago en el formulario dejando al menos un campo requerido en blanco. – Se debe presionar el botón Aceptar.

		EP 1.3: Modificar frecuencia de pago introduciendo errores en los datos.	<ul style="list-style-type: none"> – Se debe presionar el botón Modificar. – Se debe introducir los datos de la frecuencia de pago en el formulario introduciendo algunos datos inválidos. – Se debe presionar el botón Aceptar.
		EP 1.4: Cancelar operación.	<ul style="list-style-type: none"> – Se debe presionar el botón Modificar. – Se debe introducir los datos de la frecuencia de pago o no en el formulario. – Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Denominación.	Campo de texto.	NO	Combinación de letras y números.
2	Cantidad por mes.	Campo de texto.	NO	Combinación de letras y números.

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Denominación	Cantidad por mes	Respuesta del sistema	Resultado de la prueba
------------------	-----------	--------------	------------------	-----------------------	------------------------

EP 1.1	Modificar frecuencia de pago introduciendo los datos correctamente.	V(denominación 1)	V(cantidad1)	Muestra un mensaje: “La frecuencia de pago se modifico satisfactoriamente.”.	
EP 1.2	Modificar frecuencia de pago dejando campos requeridos en blanco.	I(Vacío)	V(cantidad1)	Se muestra un mensaje de error: “Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s).”.	
		V(1a)	I(Vacío)		
EP 1.3	Modificar frecuencia de pago introduciendo errores en los datos.	I(##\$%)	V(cantidad1)	Se muestra un mensaje de error: “Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s).”.	
		V(denominación 1)	(##\$%)		
EP 1.4	Cancelar operación	NA	NA	Se cancela la operación y se cierra la ventana.	

Diseño de caso de prueba del requisito Eliminar frecuencia de pago

1. Condiciones de ejecución

- El usuario debe estar autenticado en el sistema.
- Se debe seleccionar menú Inicio.
- Se debe seleccionar Sub-sistemas.

- Se debe seleccionar Capital Humano.
- Se debe seleccionar Remuneración - Nómina.
- Se debe seleccionar Gestionar frecuencia de pago.
- Se debe seleccionar la opción Gestionar frecuencia de pago.
- Debe de existir por lo menos una frecuencia de pago en el sistema.

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Eliminar frecuencia de pago.	Se elimina una frecuencia de pago.	EP 1.1: Eliminar frecuencia de pago correctamente.	<ul style="list-style-type: none"> – Se debe seleccionar el frecuencia de pago a eliminar. – Se debe presionar el botón Eliminar.
		EP 1.2: Cancelar operación.	<ul style="list-style-type: none"> – Se debe seleccionar el frecuencia de pago a eliminar. – Se debe presionar el botón Eliminar. – Se debe presionar el botón Cancelar.

1.1.1 Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

1.1.2 Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar frecuencia de pago correctamente.	Muestra un mensaje: “La frecuencia de pago se ha eliminado satisfactoriamente.”.	
EP 1.2	Cancelar operación	Se cancela la operación y se cierra la ventana.	

Anexo 22

Descripciones de los diagramas de clases del diseño.

Período de pago

Nombre: Gestionar Período de pago Controller	
Tipo de clase: Controller	
Para cada responsabilidad	
Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	gestionarperiodopagoAction()
Descripción:	Constructor de la clase.
Nombre:	insertarperiodopagoAction()
Descripción:	Este método carga el insertar período de pago.
Nombre:	modificarperiodopagoAction()
Descripción:	Este método carga el modificar período de pago.
Nombre:	eliminarperíodopagoAction()
Descripción:	Este método carga el eliminar período de pago.

Nombre:	generarperiodopagoAction()
Descripción:	Este método carga el generar los periodos de pago.

Nombre: Dat Período de pago Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	Insertar(DatPeriodopago : DatPeriodopago)
Descripción:	Se inserta el período de pago pasado por el parámetro.
Nombre:	Actualizar(DatPeriodopago : DatPeriodopago)
Descripción:	Se actualizan los datos del período de pago pasado por el parámetro.
Nombre:	Eliminar(DatPeriodopago)
Descripción:	Se elimina el período de pago pasado por el parámetro.
Nombre:	Buscar(idperiodopago)
Descripción:	Busca el período de pago dado el id pasado por el parámetro.
Nombre:	GenerarPeriodopago()
Descripción:	Se genera un período de pago.

Nombre: Dat Período de pago Model	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de dato.
Nombre:	GetLLave()
Descripción:	Devuelve la llave primaria de la tabla.
Nombre:	Buscar(idperiodo)

Descripción:	Busca un período de pago determinado.
Nombre:	Validar(idperiodo)
Descripción:	Valida que el periodo se encuentre y devuelve todos sus datos en un arreglo.
Nombre:	ValidarInc(idperiodo)
Descripción:	Valida que el periodo tenga incidencias y devuelve verdadero o falso.
Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	GetPorLimite(limite = 10, inicio = 0, idestructuracomun)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.
Nombre:	CountPeriodos(idestructuracomun)
Descripción:	Cuenta la cantidad de periodo de pago que hay en la tabla.
Nombre:	getPeriodopagoById(identificador)
Descripción:	Devuelve un periodo de pago dado un identificador.
Nombre:	GetDenom(idperiodopago)
Descripción:	Devuelve la denominación de un periodo de pago dado el id de este.
Nombre:	ExistePrograma(idprograma)
Descripción:	Dado el id del programa verifica que este exista.

Frecuencia de pago

Nombre: Dat Frecuencia de pago Model	
Tipo de clase: Model (Bussines).	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	Insertar(DatFrecuenciapago : DatFrecuenciapago)

Descripción:	Se inserta la frecuencia de pago pasada por parámetro.
Nombre:	Actualizar(DatFrecuenciapago : DatFrecuenciapago)
Descripción:	Se actualizan los datos de la frecuencia de pago pasada por parámetro.
Nombre:	Eliminar(DatFrecuenciapago)
Descripción:	Se elimina la frecuencia de pago pasada por parámetro.

Nombre: Dat Frecuencia de pago Model.	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de datos.
Nombre:	GetLLave()
Descripción:	Devuelve la llave primaria de la tabla.
Nombre:	Buscar(idfrecuenciapago)
Descripción:	Busca una frecuencia de pago dado el id.
Nombre:	Validar(idfrecuenciapago)
Descripción:	Verifica que exista la frecuencia de pago dado el id y devuelve todos sus datos en un arreglo.
Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	GetPorLimite(limite = 10, inicio = 0, idestructuracomun)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.
Nombre:	getFrecuenciapagoById(identificador)
Descripción:	Devuelve una frecuencia de pago dado el identificador.
Nombre:	GetDenom(idfrecuenciapago)
Descripción:	Devuelve la denominación de una frecuencia de pago dado su id.

Nombre: Gestionar frecuencia de pago Controller	
Tipo de clase: Controller	
Para cada responsabilidad	
Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	gestionarfrecuenciapagoAction()
Descripción:	Constructor de la clase.
Nombre:	insertarfrecuenciapagoAction()
Descripción:	Este método carga el insertar frecuencia de pago.
Nombre:	modificarfrecuenciapagoAction()
Descripción:	Este método carga el modificar frecuencia de pago.
Nombre:	eliminarfrecuenciapagoAction()
Descripción:	Este método carga el eliminar frecuencia de pago.

Nomenclador tipo de ajuste

Nombre: Gestionar tipo de ajuste Controller	
Tipo de clase: Controller	
Para cada responsabilidad	
Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	gestionartipoajusteAction()
Descripción:	Constructor de la clase.
Nombre:	insertartipoajusteAction()
Descripción:	Este método carga el insertar tipo de ajuste.

Nombre:	modificartipoajusteAction()
Descripción:	Este método carga el modificar tipo de ajuste.
Nombre:	eliminartipoajusteAction()
Descripción:	Este método carga el eliminar tipo de ajuste.

Nombre: Nom Tipo de ajuste nomina Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	Insertar(NomTipoajuste : NomTipoajustenomina)
Descripción:	Se inserta el tipo de ajuste pasado por parámetro.
Nombre:	Actualizar(NomTipoajuste : NomTipoajustenomina)
Descripción:	Se modifican el tipo de ajuste pasado por parámetro.
Nombre:	Eliminar(NomTipoajuste)
Descripción:	Se elimina el tipo de ajuste pasado por parámetro.

Nombre: Nom Tipo de ajuste nomina	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de dato.
Nombre:	GetLLave()
Descripción:	Devuelve la llave primaria de la tabla.

Nombre:	Buscar(idtipoajuste)
Descripción:	Busca un tipo de ajuste dado el id.
Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.
Nombre:	CountTAjuste()
Descripción:	Cuenta la cantidad de ajustes que hay en la tabla.
Nombre:	getTipoajusteById(identificador)
Descripción:	Devuelve una frecuencia de pago dado el identificador.
Nombre:	Validar(idajuste)
Descripción:	Valida que exista el tipo de ajuste dado el id y lo devuelve en un arreglo.
Nombre:	ValidarCodigo(codigo)
Descripción:	Verifica que el código sea un valor valido.

Cierre contable

Nombre: Cierre contable Controller	
Tipo de clase: Controller	
Para cada responsabilidad	
Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	cierrecontableAction()
Descripción:	Constructor de la clase.
Nombre:	cerrarperiodoAction()
Descripción:	Este método carga el cierre del período.

Nombre:	cerrarejercicioAction()
Descripción:	Este método carga el cierre del ejercicio.

Nombre: Cierre Contable Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	CerrarPeriodo()
Descripción:	Se cierra el período contable.
Nombre:	CerrarEjercicio()
Descripción:	Se cierra el ejercicio contable.
Nombre:	Verificarestadonomina()
	Este método verifica que el estado de las nóminas de ese período sea confirmado.
Nombre:	Verificarestadocomprobante()
Descripción:	Este método verifica que los comprobantes hayan sido enviados a contabilidad.
Nombre:	Conciliarsubmayores()
Descripción:	Este método verifica que el saldo de los submayores coincida con las cuentas de contabilidad.

Comprobante de operaciones

Nombre: Comprobante de operaciones Controller	
Tipo de clase: Controller	
Para cada responsabilidad	

Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	comprobantedeoperacionesAction()
Descripción:	Constructor de la clase.
Nombre:	revisarcomprobanteoperacionesAction()
Descripción:	Este método permite ver los detalles del comprobante de operaciones.
Nombre:	emitircomprobanteoperacionesAction()
Descripción:	Este método carga el emitir comprobante de operaciones.

Nombre: Dat Comprobante operaciones Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	ActualizarComprobante(DatComprobanteoperaciones)
Descripción:	Se actualizan los datos del comprobante de operación pasado por parámetro.
Nombre:	CrearComporbante(DatComprobanteoperaciones: DatComprobanteoperaciones)
Descripción:	Se crea el comprobante de operación pasado por parámetro.
Nombre:	Emitir comprobante(DatComprobanteoperaciones)
Descripción:	Este método envía el comprobante de operaciones de la nómina a Contabilidad.

Nombre: Dat Comprobante operaciones	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	

Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de dato.
Nombre:	GetLLave()
Descripción:	Devuelve la llave primaria de la tabla.
Nombre:	Buscar(idcomproboperaciones)
Descripción:	Busca un comprobante de operaciones dado el id.
Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.
Nombre:	getdatoscomprobanteByld(idcomproboperaciones)
Descripción:	Devuelve un comprobante de operaciones dado el identificador.
Nombre:	Validar(idcomproboperaciones)
Descripción:	Valida que exista el comprobante de operaciones dado el id y lo devuelve en un arreglo.

Asientos

Nombre: Dat Asientos Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	Insertar(DatAsientos : DatAsientos)
Descripción:	Se insertan el asiento pasado por parámetro.
Nombre:	Actualizar(DatAsientos : DatAsientos)
Descripción:	Se actualiza el asiento pasado por parámetro.

Nombre:	Eliminar(DatAsientos)
Descripción:	Se elimina el asiento pasado por parámetro.

Nombre: Dat Asientos	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de dato.
Nombre:	GetLLave()
Descripción:	Devuelve la llave primaria de la tabla.
Nombre:	Buscar(idasiento)
Descripción:	Busca un asiento dado el id.
Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.
Nombre:	getdatoscomprobanteByld(idasiento)
Descripción:	Devuelve un asiento dado el identificador.

Pases

Nombre: Dat Pases Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	Insertar(DatPases : DatPases)

Descripción:	Se insertan el pase pasado por parámetro.
Nombre:	Actualizar(DatPases : DatPases)
Descripción:	Se actualiza el pase pasado por parámetro.
Nombre:	Eliminar(DatPases)
Descripción:	Se elimina el pase pasado por parámetro.

Nombre: Dat Pases	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de dato.
Nombre:	GetLLave()
Descripción:	Devuelve la llave primaria de la tabla.
Nombre:	Buscar(idpase)
Descripción:	Busca un pase dado el id.
Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.
Nombre:	getdatoscomprobanteByld(idpase)
Descripción:	Devuelve un pase dado el identificador.

Configuración

Nombre: Configuración Controller
Tipo de clase: Controller
Para cada responsabilidad

Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	configuracionAction()
Descripción:	Constructor de la clase.
Nombre:	establecerconfiguraciondecuentasAction()
Descripción:	Este método carga el establecer configuración de cuentas.

Nombre: Dat Configuración Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	Actualizar(DatConfiguracion : DatConfiguracion)
Descripción:	Se actualizan los datos de la configuración pasada por parámetro.
Nombre:	Establecerconfiguracion()
Descripción:	Se establecen la configuración de las cuentas.

Nombre: Dat Configuración	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de dato.
Nombre:	GetLLave()
Descripción:	Devuelve la llave primaria de la tabla.
Nombre:	Buscar(idconfiguracion)
Descripción:	Busca una configuración dado el id.

Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.
Nombre:	getdatoscomprobanteByld(idconfiguracion)
Descripción:	Devuelve una configuración dado el identificador.

Nomenclador impuesto y contribuciones

Nombre: Gest impuesto Controller	
Tipo de clase: Controller	
Para cada responsabilidad	
Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	gestimpuestoAction()
Descripción:	Constructor de la clase.
Nombre:	insertarimpuestoAction()
Descripción:	Este método carga el insertar impuesto.
Nombre:	modificarimpuestoAction()
Descripción:	Este método carga el modificar impuesto
Nombre:	eliminarimpuestoAction()
Descripción:	Este método carga el eliminar impuesto.

Nombre: Nom Impuesto Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	NomImpuestoModel()

Descripción:	Constructor de la clase.
Nombre:	Insertar(NomImpuesto : NomImpuesto)
Descripción:	Se inserta el impuesto pasado por parámetro.
Nombre:	Actualizar(NomImpuesto : NomImpuesto)
Descripción:	Se actualizan los datos del impuesto pasado por parámetro.
Nombre:	Eliminar(NomImpuesto)
Descripción:	Se elimina el impuesto pasado por parámetro.

Nombre: Nom Impuesto	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de dato.
Nombre:	GetLLave()
Descripción:	Devuelve la llave primaria de la tabla.
Nombre:	Buscar(idimpuesto)
Descripción:	Busca un impuesto dado el id.
Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.
Nombre:	GetDenom(idimpuesto)
Descripción:	Devuelve la denominación dado el identificador de un impuesto.
Nombre:	Cantimpuestos()
Descripción:	Cuenta la cantidad de impuestos.
Nombre:	ValidarCodigo(codigo)

Descripción:	Verifica que el valor del código sea válido.
--------------	--

Concepto de pago

Nombre: Gestionar concepto pago Controller	
Tipo de clase: Controller	
Para cada responsabilidad	
Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	gestionarconceptopagoAction()
Descripción:	Constructor de la clase.
Nombre:	insertarconceptopagoAction()
Descripción:	Este método carga el insertar concepto de pago.
Nombre:	modificarconceptopagoAction()
Descripción:	Este método carga el modificar concepto de pago
Nombre:	eliminarconceptopagoAction()
Descripción:	Este método carga el eliminar concepto de pago.

Nombre: Dat Concepto pago Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	Insertar(DatConceptopago : DatConceptopago)
Descripción:	Se inserta el concepto de pago pasado por parámetro.
Nombre:	Actualizar(DatConceptopago : DatConceptopago)
Descripción:	Se actualizan los datos del concepto de pago pasado por parámetro.

Nombre:	Eliminar(DatConceptopago)
Descripción:	Se elimina el concepto de pago pasado por parámetro.

Nombre: Dat Concepto pago	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de dato.
Nombre:	GetLLave()
Descripción:	Devuelve la llave primaria de la tabla.
Nombre:	Buscar(idconceptopago)
Descripción:	Busca un concepto de pago dado el id.
Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.
Nombre:	GetDenom(idconceptopago)
Descripción:	Devuelve la denominación dado el identificador.
Nombre:	ValidarCodigo(codigo)
Descripción:	Verifica que el valor del codigo sea valido.
Nombre:	Validar(idconceptopago)
Descripción:	Verifica que el concepto de pago exista y lo devuelve en un arreglo.
Nombre:	getConceptopagoById(identificador)
Descripción:	Devuelve el concepto de pago dado el identificador.

Nomenclador tipo de nómina

Nombre: Gestionar concepto pago Controller

Tipo de clase: Controller	
Para cada responsabilidad	
Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	gestionartiponominaAction()
Descripción:	Constructor de la clase.
Nombre:	insertartiponominaAction()
Descripción:	Este método carga el insertar tipo de nómina.
Nombre:	modificartiponominaAction()
Descripción:	Este método carga el modificar tipo de nómina.
Nombre:	eliminartiponominaAction()
Descripción:	Este método carga el eliminar tipo de nómina.

Nombre: Nom Tipo nómina Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	Insertar(NomTiponomina : NomTiponomina)
Descripción:	Se inserta el tipo de nómina pasado por parámetro.
Nombre:	Actualizar(NomTiponomina : NomTiponomina)
Descripción:	Se actualizan los datos del tipo de nómina pasado por parámetro.
Nombre:	Eliminar(NomTiponomina)
Descripción:	Se elimina el tipo de nómina pasado por parámetro.

Nombre: Nom Tipo nómina	
Tipo de clase: Model (Domain)	

Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de dato.
Nombre:	GetLLave()
Descripción:	Devuelve la llave primaria de la tabla.
Nombre:	Buscar(id_tiponominas)
Descripción:	Busca un tipo de nómina dado el id.
Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.
Nombre:	GetDenom(id_tiponominas)
Descripción:	Devuelve la denominación dado el identificador.
Nombre:	ValidarCodigo(codigo)
Descripción:	Verifica que el valor del código sea válido.
Nombre:	Validar(id_tiponominas)
Descripción:	Verifica que el tipo de nómina exista y lo devuelve en un arreglo.

Nomenclador funciones

Nombre: Nom funciones Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	Verfunciones()
Descripción:	Este método permite ver todas las funciones existentes.

Nombre: Nom funciones	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de dato.
Nombre:	GetLLave()
Descripción:	Devuelve la llave primaria de la tabla.
Nombre:	Buscar(id_funciones)
Descripción:	Busca una función dado el id.
Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.
Nombre:	GetDenom(id_funciones)
Descripción:	Devuelve la denominación dado el identificador.

Administrar nómina

Nombre: Administrar nómina Controller	
Tipo de clase: Controller	
Para cada responsabilidad	
Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	administrarnominaAction()

Descripción:	Constructor de la clase.
Nombre:	insertarnominaAction()
Descripción:	Este método carga el insertar nómina.
Nombre:	agregartrabajadorAction()
Descripción:	Este método carga el agregar trabajador.
Nombre:	eliminartrabajadorAction()
Descripción:	Este método carga el eliminar trabajador.
Nombre:	modificarnominaAction()
Descripción:	Este método carga el modificar nómina.
Nombre:	eliminarnominaAction()
Descripción:	Este método carga el eliminar nómina.
Nombre:	procesarnominaAction()
Descripción:	Este método carga el procesar nómina.
Nombre:	filtrarporperiodoAction()
Descripción:	Este método filtra por período las nóminas.

Procesamiento de nómina

Nombre: Dat Procesamiento nómina Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	Insertar(DatProcesamientonomina : DatProcesamientonomina)
Descripción:	Se inserta el procesamiento pasado por parámetro.
Nombre:	Actualizar(DatProcesamientonomina : DatProcesamientonomina)
Descripción:	Se actualizan los datos del procesamiento pasado por parámetros.
Nombre:	Eliminar(DatProcesamientonomina)
Descripción:	Se elimina el procesamiento pasado por parámetro.

Nombre:	Procesarnomina(DatNomina)
Descripción:	Se procesa la nómina pasada por parámetro.
Nombre:	Eliminartrabajador(trabajador)
Descripción:	Se elimina el trabajador de una nómina.
Nombre:	Agregartrabajador()
Descripción:	Se adiciona el trabajador a una nómina.

Nombre: Dat Procesamiento nómina	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de dato.
Nombre:	GetLLave()
Descripción:	Devuelve la llave primaria de la tabla.
Nombre:	Buscar(idprocesamientonomina)
Descripción:	Busca un procesamiento dado el id.
Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	GetTodosbyidnomina(idnomina)
Descripción:	Devuelve una nómina dado el id.
Nombre:	ordenarpagosadd(idprocesamientonomina)
Descripción:	Ordena los pagos adicionales dado el id del procesamiento de una nómina.
Nombre:	ordenarincidencias(idprocesamientonomina)
Descripción:	Ordena las incidencias dado el id del procesamiento de una nómina.
Nombre:	ordenarretenciones(idprocesamientonomina)
Descripción:	Ordena las retenciones dado el id del procesamiento de una nómina.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.

Nombre:	GetSalarioEscala(idprocesamientonomina)
Descripción:	Se obtiene el salario escala del trabajador dado el procesamiento de la nómina.
Nombre:	GetIdNomina(idprocesamientonomina)
Descripción:	Se obtiene el id de la nómina dado el id del procesamiento.
Nombre:	CantNProcesadas()
Descripción:	Cuenta la cantidad de nóminas procesadas.
Nombre:	Trabajadores(idnomina)
Descripción:	Devuelve dado el id de una nómina los trabajadores de esta.
Nombre:	canttrabsbynomina(idnomina)
Descripción:	Cuenta la cantidad de trabajadores por nómina.
Nombre:	IDproceso(idnomina)
Descripción:	Devuelve todos los procesamientos que tiene una nómina dado el id de esta.
Nombre:	acumulados(idnomina)
Descripción:	Devuelve los días acumulados, impuestos acumulados y el id del trabajador.

Revisar nómina

Nombre: Revisar nómina Controller	
Tipo de clase: Controller	
Para cada responsabilidad	
Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	revisarnominaAction()
Descripción:	Constructor de la clase.
Nombre:	confirmarAction()

Descripción:	Este método carga el confirmar nómina.
Nombre:	revertirnominaAction()
Descripción:	Este método carga el revertir nómina.
Nombre:	ajustarAction()
Descripción:	Este método carga el ajustar nómina.

Nómina

Nombre: Dat Nomina Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	Insertar(DatNomina : DatNomina)
Descripción:	Se inserta una nómina pasada por parámetro.
Nombre:	Actualizar(DatNomina : DatNomina)
Descripción:	Se actualizan los datos de la nómina pasada por parámetro.
Nombre:	Eliminar(DatNomina)
Descripción:	Se elimina la nómina pasada por parámetro.
Nombre:	InsertarTrabajador()
Descripción:	Se insertan trabajadores a una nómina.
Nombre:	Filtrarporperiodo()
Descripción:	Este método filtra por período las nóminas.
Nombre:	Confirmar(DatNomina)
Descripción:	Este método confirma la nómina.
Nombre:	Revertir(DatNomina)
Descripción:	Este método revierte la nómina.
Nombre:	Ajustar(DatNomina)

Descripción:	Este método ajusta la nómina.
--------------	-------------------------------

Nombre: Dat Nomina	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de dato.
Nombre:	GetLLave()
Descripción:	Devuelve la llave primaria de la tabla.
Nombre:	Buscar(idnomina)
Descripción:	Busca una nómina dado el id.
Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	GetTodosporestado()
Descripción:	Devuelve el estado de todas las nominas de la tabla.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.
Nombre:	denominacion(idnomina)
Descripción:	Devuelve la denominación dado el identificador.
Nombre:	datos(idnomina)
Descripción:	Devuelve los datos de una nómina dado el id.
Nombre:	getIldtipoNomina(idnomina)
Descripción:	Devuelve el tipo de la nómina dado el id.
Nombre:	nominascontabilizadas(idestructura)
Descripción:	Devuelve si están o no contabilizadas las nóminas dado el id de la estructura.
Nombre:	CountNominas(idestructura)
Descripción:	Cuentas las nóminas que existen.
Nombre:	GetLastLLave()

Descripción:	Devuelve el id mayor de la tabla.
Nombre:	horas(idnomina)
Descripción:	Devuelve las horas trabajadas.
Nombre:	existenomina(idnomina)
Descripción:	Verifica que exista la nómina dado el id.
Nombre:	procesada(idnomina)
Descripción:	Devuelve la nómina si esta procesada dado el id.
Nombre:	confirmada(idnomina)
Descripción:	Devuelve la nómina si esta confirmada dado el id.
Nombre:	estados()
Descripción:	Devuelve todas las nóminas que no están procesadas.
Nombre:	verificareliminacion(idnomina)
Descripción:	Valida que la nómina haya sido eliminada dado el id.
Nombre:	elaborada(idnomina)
Descripción:	Devuelve las nóminas que no han sido confirmas.

Nomenclador estado nómina

Nombre: Nom Estado nomina Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	Insertar(NomEstado : NomEstadonomina)
Descripción:	Se inserta el estado pasado por parámetro.
Nombre:	Actualizar(NomEstado : NomEstadonomina)
Descripción:	Se actualizan los datos del estado pasado por parámetros.
Nombre:	Eliminar(NomEstado)
Descripción:	Se elimina el estado pasado por parámetro.

Nombre: Nom Estado nomina	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de dato.
Nombre:	GetLLave()
Descripción:	Devuelve la llave primaria de la tabla.
Nombre:	Buscar(idestado)
Descripción:	Busca una función dado el id.
Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.

Ajuste de concepto de pago por procesamiento de nómina

Nombre: Ajuste concepto_procnomina Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	Insertar(Ajusteconcepto_procnomina : Ajusteconcepto_procnomina)
Descripción:	Se inserta el ajuste pasado por parámetro.
Nombre:	Actualizar(Ajusteconcepto_procnomina : Ajusteconcepto_procnomina)
Descripción:	Se actualizan los datos del ajuste pasado por parámetros.
Nombre:	Eliminar(Ajusteconcepto_procnomina)
Descripción:	Se elimina el ajuste pasado por parámetro.

Nombre: Ajuste concepto_procnomina	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de dato.
Nombre:	GetLLave()
Descripción:	Devuelve la llave primaria de la tabla.
Nombre:	Buscar(idajuste)
Descripción:	Busca una función dado el id.
Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.

Concepto de pago por procesamiento de nómina

Nombre: Concepto_procnomina Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	Insertar(Concepto_procnomina : Concepto_procnomina)
Descripción:	Se inserta el concepto_procnomina pasado por parámetro.
Nombre:	Actualizar(Concepto_procnomina : Concepto_procnomina)
Descripción:	Se actualizan los datos del concepto_procnomina pasado por parámetros.
Nombre:	Eliminar(Concepto_procnomina)
Descripción:	Se elimina el concepto_procnomina pasado por parámetro.

Nombre: Concepto_procnomina	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de dato.
Nombre:	GetLLave()
Descripción:	Devuelve la llave primaria de la tabla.
Nombre:	Buscar(idconcepto_procnomina)
Descripción:	Busca un concepto_procnomina dado el id.
Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.
Nombre:	GetDenom(idconcepto_procnomina)
Descripción:	Devuelve el concepto_procnomina dado el id.

Procesamiento de pagos adicionales

Nombre: Dat Procesamiento pagos adicionales Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	Insertar(DatProcesamientopagosadicionales : DatProcesamientopagosadicionales)
Descripción:	Se inserta el procesamiento de pagos adicionales pasado por parámetro.

Nombre:	Actualizar(DatProcesamientopagosadicionales DatProcesamientopagosadicionales) :
Descripción:	Se actualizan los datos del procesamiento de pagos adicionales pasado por parámetros.
Nombre:	Eliminar(DatProcesamientopagosadicionales)
Descripción:	Se elimina el procesamiento de pagos adicionales pasado por parámetro.

Nombre: Dat Procesamiento pagos adicionales	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de dato.
Nombre:	GetLLave()
Descripción:	Devuelve la llave primaria de la tabla.
Nombre:	Buscar(idprocesamientopagosadicionales)
Descripción:	Busca un procesamiento de pagos adicionales dado el id.
Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	Pagostrabajador(idtrabajador)
Descripción:	Devuelve los pagos de un trabajador dado el id.
Nombre:	GetIdPagoAdd(idprocesamientonomina)
Descripción:	Devuelve el id del pago adicional dado el id del procesamiento de la nómina.
Nombre:	Pagos(idtrabajador)
Descripción:	Devuelve el total a pagar al trabajador por los pagos adicionales.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.

Procesamiento de retenciones

Nombre: Dat Procesamiento retenciones Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	Insertar(DatProcesamientoretenciones: DatProcesamientoretenciones)
Descripción:	Se inserta el procesamiento de retenciones pasado por parámetro.
Nombre:	Actualizar(DatProcesamientoretenciones: DatProcesamientoretenciones)
Descripción:	Se actualizan los datos del procesamiento de retenciones pasado por parámetros.
Nombre:	Eliminar(DatProcesamientoretenciones)
Descripción:	Se elimina el procesamiento de retenciones pasado por parámetro.

Nombre: Dat Procesamiento retenciones	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de dato.
Nombre:	GetLLave()
Descripción:	Devuelve la llave primaria de la tabla.
Nombre:	Buscar(idprocesamientoretenciones)
Descripción:	Busca un procesamiento de retenciones dado el id.
Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	Existe(idprocesamientonomina, idtrabajador, idregistroretenciones)
Descripción:	Devuelve si el trabajador tiene o no retenciones.
Nombre:	TotalRetenciones(idtrabajador)
Descripción:	Devuelve el total de retenciones del trabajador dado el id.

Nombre:	GetIdProcRet(idprocesamientonomina)
Descripción:	Devuelve el id del procesamiento de retenciones.
Nombre:	retencionesbynomina(idprocesamiento)
Descripción:	Devuelve las retenciones de una nómina.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.

Procesamiento de incidencias

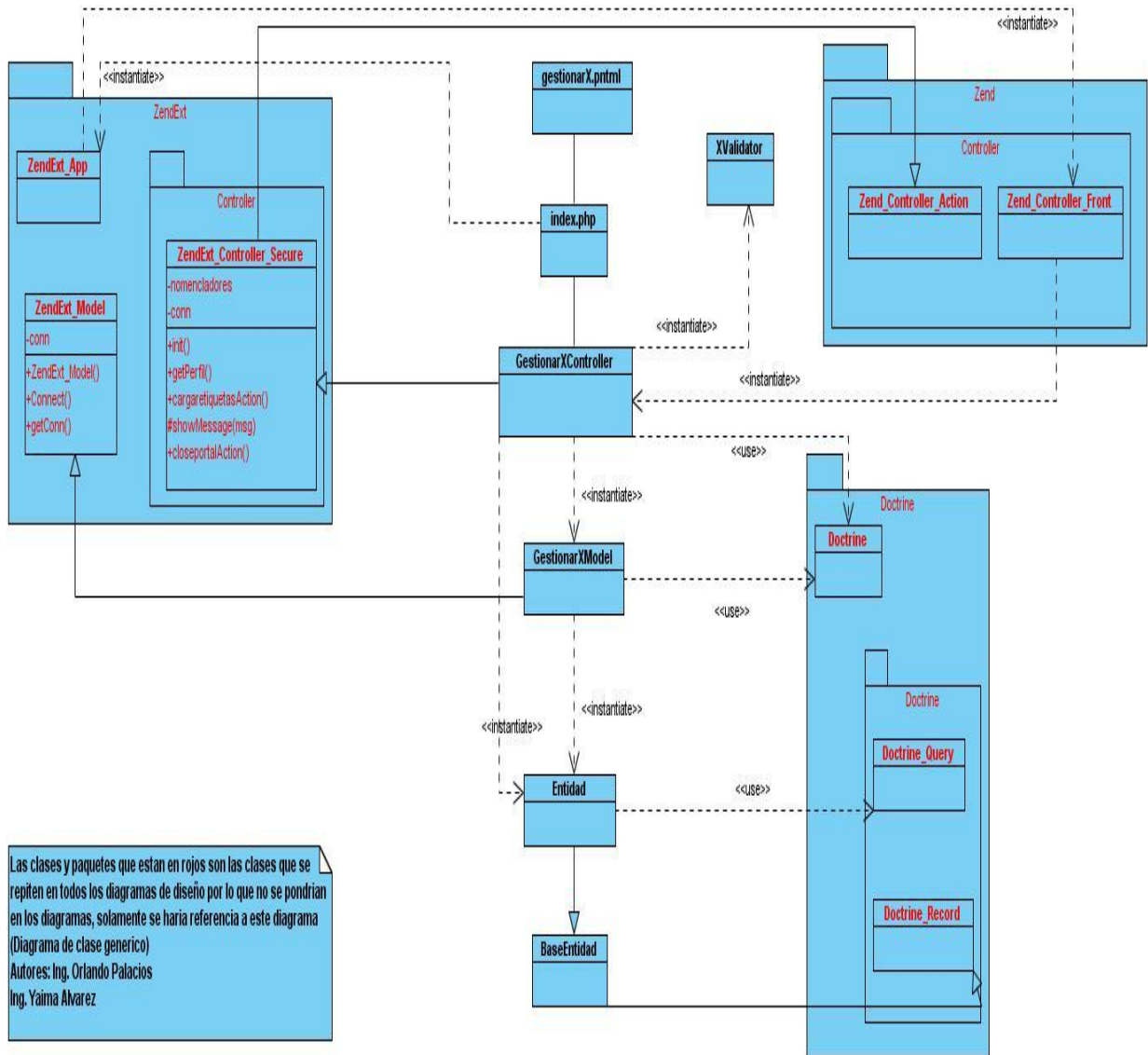
Nombre: Dat Procesamiento incidencium Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Constructor de la clase.
Nombre:	Insertar(DatProcesamientoincidencium: DatProcesamientoincidencium)
Descripción:	Se inserta el procesamiento de incidencias pasado por parámetro.
Nombre:	Actualizar(DatProcesamientoincidencium: DatProcesamientoincidencium)
Descripción:	Se actualizan los datos del procesamiento de incidencias pasado por parámetros.
Nombre:	Eliminar(DatProcesamientoincidencium)
Descripción:	Se elimina el procesamiento de incidencias pasado por parámetro.

Nombre: Dat Procesamiento incidencium	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	Setup()
Descripción:	Este método relaciona la tabla con otra de la base de dato.
Nombre:	GetLLave()

Descripción:	Devuelve la llave primaria de la tabla.
Nombre:	Incidenciasrestan(idtrabajador)
Descripción:	Devuelve las incidencias que restan dado el id del trabajador.
Nombre:	GetTodos()
Descripción:	Devuelve todas las tuplas de la tabla.
Nombre:	Incidenciassuman(idtrabajador)
Descripción:	Devuelve las incidencias que suman dado el id del trabajador.
Nombre:	Buscar(idprocesamientoincidencia)
Descripción:	Busca el procesamiento de incidencia dado el id.
Nombre:	GetIdIncidenciaSaldo(idprocesamientonomina)
Descripción:	Devuelve el saldo de la incidencia.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve la cantidad de tuplas que se le define en límite que se encuentran a partir del valor que se le define en el inicio.

Anexo 23

Diagrama de clases genérico del diseño



GLOSARIO

Proceso: Conjunto de actividades enlazadas entre si que, partiendo de una o más entradas las transforman en salidas.

Subsistema: Cada uno de los componentes principales de un sistema que este dividido en componentes. Cada subsistema abarca aspectos del sistema que comparten alguna propiedad común.

Devengado: Derecho ganado que todavía no ha sido cobrado.

Subsidio: Se refiere a los ingresos que recibe el trabajador en sustitución del salario, cuando se enferma o accidenta.

Pagos Adicionales: Son formas de pagos adicionales que se expresan en términos de por cientos, de importes fijos o de importes proporcionales.

Módulo: Es un componente autocontrolado de un sistema, el cual posee una interfaz bien definida hacia otros componentes. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará una de dichas tareas (o quizás varias en algún caso).

Nomenclador: Es una especie de catálogo que tiene la nomenclatura, o vocabulario de una rama.

Opción: Alternativas posibles a seleccionar.

Sistema: Es un conjunto de procesos o elementos interrelacionados con un medio para formar una totalidad encauzada hacia un objetivo común.

Período de Pago: Un período puede ser un mes calendario, o una quincena, o una decena, o en fin, cualquier rango de fecha. Todo período de pago debe estar enteramente comprendido dentro de un período contable, aunque diferentes períodos contables pueden estructurarse de diferente forma en períodos de pago.

Actualización: Puesta al día de algo que se ha quedado atrasado. Última versión.

Componente: Es una unidad de composición de aplicaciones software, que posee un conjunto de interfaces y un conjunto de requisitos, y que ha de poder ser desarrollado, adquirido, incorporado al sistema y compuesto con otros componentes de forma independiente, en tiempo y espacio.

Elemento: Parte integrante de una cosa.

Configuración: Es una adaptación a condiciones definidas por el entorno, que le es hecha a una aplicación.

Nomenclador: Es una especie de catalogo que tiene la nomenclatura, o vocabulario de una rama.

Funcionalidad: Representa la forma en que un dispositivo funciona; es decir, los mecanismo o secuencia de eventos que hacen que el objeto realice cierta función.