

Universidad de las Ciencias Informáticas

Facultad 4



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

Título: Análisis y Diseño del componente Submayores del Subsistema Capital Humano del sistema Cedrux

Autoras: Mисleydis Esther Caballero Batista
Beatriz Escobar Pupo

Tutora: Ing. Lisett Pérez Quintero
Co-tutor: Ing. Roberto Oscar Labrada Sedeño

Ciudad de la Habana, Junio del 2008
“Año del 50 Aniversario del Triunfo de la Revolución”

Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.
Albert Einstein

DATOS DE CONTACTO

Datos del Tutor:

Ing. en Ciencias Informáticas Lisett Pérez Quintero. Profesor graduado de la Universidad de las Ciencias Informáticas en el año 2008, actualmente trabaja en la Dirección de Producción número 4 de la Universidad de las Ciencias Informáticas.

Correo: lquintero@uci.cu

Datos del Co-Tutor:

Ing. en Ciencias Informáticas Roberto Oscar Labrada Sedeño. Profesor graduado en la Universidad de las Ciencias Informáticas en el año 2008, actualmente profesor de Física en la Facultad Regional de Granma.

Correo: rolabrada@grm.uci.cu

AGRADECIMIENTOS

Ante todo les agradezco a mis padres por darme siempre lo mejor y por consertirme en todos mis caprichos, a ellos que siempre han confiado en mí y que me han enseñado a ser sacrificada y persistente. Por quererme tanto, por sus consejos, apoyo y sacrificios.

A mi hermano por tenerme siempre presente, por quererme y ayudarme en todo lo que esta a su alcance y por tener infinita confianza en mi.

A mi novio por su comprensión y amor durante todos estos años, por entenderme y ayudarme a alcanzar el sueño de ser ingeniera.

A mi compañera de tesis por comprenderme, apoyarme y por haber sido paciente en todo momento.

A mis amigos por estar siempre presente en todo momento y brindarme su apoyo y compañía.

A todas aquellas personas que de una forma u otra me han apoyado y han contribuido con este trabajo de diploma.

Misly

A mis padres por haberme dado todo su amor y apoyo.

A mi hermana por siempre estar ahí.

A mis abuelos por quererme tanto.

A mis tios y tias, en especial a mi tío Landy.

A mis primos y primas por su cariño.

A mis amigos, tanto las viejas amistades como las nuevas.

A mi compañera de tesis por su apoyo, comprensión y guía.

A todas aquellas personas que de una manera u otra me ayudaron en la realización de este trabajo.

Betica

DEDICATORIA

A mis padres por su infinito amor, cariño y dedicación.

A mi hermano por quererme, apoyarme y confiar siempre en mí.

A mi novio por brindarme su apoyo, comprensión y amor.

A mi tía y abuela por tenerme siempre presente y consentirme en todo lo que han podido.

Misly

A mi mamá por ser la mejor madre del mundo.

A mi papá por ser el padre más loco y bueno que se pueda imaginar.

A mi Nana por ser la mejor hermana del mundo.

A mis abuelos por su infinito amor.

A mi tío Landy por su apoyo y cariño.

Al resto de mi familia por siempre estar ahí para mí.

Betica

RESUMEN

En el departamento de Capital Humano de todas las entidades presupuestadas y empresariales, específicamente en Remuneración y nómina se llevan a cabo la gestión de la información referente a los Submayores de vacaciones y de retenciones, estos procesos arrojan como resultado los distintos registros de los trabajadores donde se analizan las cuentas y las subcuentas que lo requieran.

En la actualidad a quedado probado que el uso de las tecnologías en los procesos de negocios contables aumenta la eficiencia y la productividad en las entidades, provocando que la información y los cálculos sean más manejables. Con el propósito de contribuir a la gestión de los Submayores en las entidades existentes en el país, el presente trabajo realiza el análisis de los procesos que se llevan a cabo en el negocio y los distintos requisitos que lo conforman, tributando a alcanzar un adecuado grado de satisfacción por parte de las entidades. Se construye el diseño para satisfacer los requerimientos y otros artefactos que ayudarán a que la solución dada sea lo más configurable y estable posible. Lo que permitirá que con el desarrollo de esta propuesta, el trabajo que se realiza en las entidades presupuestadas y empresariales tenga a su alcance una aplicación capaz de agilizar los procesos que diariamente se gestionan. Contribuyendo a la confiabilidad, seguridad y rapidez en la entrada de los datos.

PALABRAS CLAVE

Capital Humano, Remuneración y nómina, Submayor de vacaciones, Submayor de retenciones.

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	7
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	11
1.1. Sistemas de gestión	11
1.1.1. Sistemas de gestión internacionales	11
1.1.2. Sistemas de gestión nacionales	15
1.2. Marco Teórico.....	18
1.2.1. Capital Humano	18
1.2.2. Remuneración y nómina.....	18
1.2.2.a Submayores	19
1.3. Modelo de desarrollo orientado a componente	19
1.4. Lenguajes de modelado.....	20
1.4.1. Notación para el modelado de los procesos del negocio (BPMN)	21
1.4.2. Lenguaje unificado de modelado (UML)	22
1.5. Herramienta de modelado	23
1.5.1. Visual Paradigm para UML 6.1	24
1.6. Lenguaje de programación web.....	24
1.6.1. PHP 5.2 o Superior	25
1.7. Zend Studio	25
1.8. Gestor de base de datos.....	26
1.8.1. PostgreSQL 8.3 o Superior	26
1.9. Servidor web.....	27
1.9.1. Apache.....	27
CAPÍTULO 2: MODELAMIENTO DEL NEGOCIO.....	30
2.1. Técnicas para el levantamiento de requisitos	30
2.2. Patrones.....	31
2.2.1. Patrones CRUD.....	31
2.3. Descripción del negocio	32
2.4. Reglas del negocio.....	33
2.5. Procesos del negocio	34
2.6. Descripción de los procesos del negocio.....	35

2.6.1. Apertura del Submayor de vacaciones	35
2.6.2. Apertura del Submayor de retenciones	35
2.6.3. Actualizar Submayor de vacaciones.....	35
2.6.4. Actualizar Submayor de retenciones	35
2.7. Modelo conceptual	35
Fig.1. Modelo conceptual	36
2.8. Requisitos funcionales del sistema.....	38
Tabla 1. Agrupación de los requisitos funcionales (ARF)	39
2.9. Especificación de los requisitos.....	39
2.9.1. Tipo de retención.....	40
2.9.2. Tipo de programa	43
2.9.3. Gestionar tipo de recargo	45
2.9.4. Apertura y cierre del submayor de vacaciones	47
2.9.5. Actualización del submayor de vacaciones	50
2.9.6. Detalles del submayor de vacaciones	52
2.9.7. Conciliación con contabilidad del submayor de vacaciones.....	53
2.9.8. Configuración del submayor de vacaciones	54
2.9.9. Ajuste del submayor de vacaciones	55
2.9.10. Dar alta automática a un trabajador.....	56
2.9.11. Generar comprobante de operaciones dado una modificación	58
2.9.12. Gestionar submayor de retenciones	59
2.9.13. Detalles del submayor de retenciones.....	64
2.9.14. Actualizar Submayor de Retención.....	66
2.9.15. Conciliación con contabilidad del submayor de retenciones	68
2.9.16. Filtrar del submayor de retenciones.....	69
2.9.17. Pago directo al banco del submayor de retenciones	70
2.9.18. Alerta de cumplido con el pago de las retenciones de los trabajadores.....	71
2.9.19. Gestionar recargo a la retención	72
2.9.20. Descuento adelantado de las retenciones	76
2.9.21. Mostrar fichero	77
2.9.22. Modificar fichero	78

2.9.23. Alta automática al fichero	79
2.9.24. Resumen devengado del trabajador en meses anteriores.....	80
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	83
3.1. Modelo de Diseño	83
3.2. Patrones arquitectónicos.....	83
3.2.1. Patrón Modelo-Vista-Controlador (MVC)	83
3.3. Patrones de asignación de responsabilidades (Patrones GRASP).....	84
3.4. Patrones de diseño	85
3.4.1. Patrones GOF	85
3.5. Diagrama de clases.....	86
3.5.1. Diagramas de clases del diseño.....	87
Fig.1. Diagrama de clases del Gestionar submayor de vacaciones.....	87
Fig.2. Diagrama de clases del Gestionar submayor de retenciones.....	88
Fig.3. Diagrama de clases del Gestionar concepto de retención.....	89
Fig.4. Diagrama de clases del Gestionar operaciones.....	90
Fig.5. Diagrama de clases del Gestionar tipo de retenciones.....	91
Fig.6. Diagrama de clases del Gestionar nomenclador de programas.....	92
Fig.7. Diagrama de clases de la Configuración del submayor de vacaciones.....	93
Fig.8. Diagrama de clases del Fichero histórico devengado.....	94
Fig.9. Diagrama de clases del Gestionar tipo de recargo.....	95
3.6. Diagrama de componente	96
CAPÍTULO 4: VALIDACIÓN DE REQUISITOS.....	98
4.1. Técnicas para la validación de los requisitos.....	98
4.1.1. Construcción de prototipos de interfaz usuario.....	99
4.1.2. Diseño de casos de prueba.....	99
4.1.2.a. Caso de prueba del Gestionar Submayor de vacaciones	100
4.1.2.b. Casos de prueba del Gestionar submayor de retenciones.....	111
4.2. Métricas para la validación de los requisitos	125
4.2.1. Criterios para la evaluación y aceptación de los requisitos.....	126
4.3. Métricas orientadas a clases	128
4.3.1. Tamaño de clase (TC).....	128

4.3.2. Árbol de profundidad de herencia (APH)	131
4.3.3. Relaciones entre clases (RC)	131
CONCLUSIONES	136
RECOMENDACIONES	137
REFERENCIA BIBLIOGRÁFICA.....	138
BILIOGRAFÍA	140
ANEXOS.....	143
Prototipos de interfaz de usuario.....	143
Anexo1: Adicionar tipo de retención.....	143
Anexo 2: Modificar tipo de retención	143
Anexo 3: Eliminar tipo de retención.....	144
Anexo 4: Adicionar tipo de programa	144
Anexo 5: Modificar tipo de programa.....	144
Anexo 6: Eliminar tipo de programa	145
Anexo 7: Adicionar tipo de recargo	145
Anexo 8: Modificar tipo de recargo.....	145
Anexo 9: Eliminar tipo de recargo	146
Anexo 10: Apertura del submayor de vacaciones	146
Anexo 11: Cierre del submayor de vacaciones	146
Anexo 12: Detalles del submayor de vacaciones	147
Anexo 13: Configuración del submayor de vacaciones	147
Anexo 14: Ajuste del submayor de vacaciones	148
Anexo 15: Apertura del submayor de retenciones.....	148
Anexo 16: Modificar submayor de retenciones.....	149
Anexo 17: Cierre del submayor de retenciones.....	149
Anexo 18: Detalles del submayor de retenciones.....	150
Anexo 19: Filtrar del submayor de retenciones	150
Anexo 20: Pago directo al banco del submayor de retenciones	151
Anexo 21: Alerta de cumplido con el pago de la retención de los trabajadores	151
Anexo 22: Adicionar recargo a la retención.....	152
Anexo 23: Modificar recargo a la retención	152

Anexo 24: Eliminar recargo a la retención.....	153
Anexo 25: Mostrar fichero	153
Anexo 26: Modificar fichero	154
Anexo 27: Resumen devengado del trabajador en meses anteriores.....	155
Casos de Prueba	155
Anexo 28: Adicionar tipo de retención	155
Anexo 29: Modificar tipo de retención	160
Anexo 30: Eliminar tipo de retención	165
Anexo 31: Adicionar tipo de programa	167
Anexo 32: Modificar tipo de programa.....	170
Anexo 33: Eliminar tipo de programa	174
Anexo 34: Adicionar tipo recargo	175
Anexo 35: Modificar tipo recargo.....	178
Anexo 36: Eliminar tipo recargo	182
Anexo 37: Detalles del submayor de vacaciones	184
Anexo 38: Configuración del submayor de vacaciones	185
Anexo 39: Detalles del submayor de retenciones.....	187
Anexo 40: Filtrar del submayor de retenciones	189
Anexo 41: Pago directo al banco del submayor de retenciones	191
Anexo 42: Alerta de cumplido con el pago de la retención de los trabajadores	194
Anexo 43: Adicionar recargo a la retención.....	195
Anexo 44: Modificar recargo a la retención	198
Anexo 45: Eliminar recargo a la retención.....	202
Anexo 46: Mostrar fichero	203
Anexo 47: Modificar fichero.....	205
Anexo 48: Tabla de seguimiento de los requisitos.....	209
Decripción de las clases del diseño	211
Anexo 49: Gestionar submayor de vacaciones	211
Anexo 50: Gestionar submayor de retenciones.....	212
Anexo 51: Gestionar concepto de retención.....	215
Anexo 52: Gestionar operaciones	217

Anexo 53: Gestionar tipo de retenciones.....	218
Anexo 54: Gestionar nomenclador de programas	219
Anexo 55: Configuración del submayor de vacaciones	221
Anexo 56: Fichero histórico devengado	222
Anexo 57: Gestionar tipo de recargo.....	223
GLOSARIO	226

INTRODUCCIÓN

La revolución tecnológica que vive la humanidad actualmente se corresponde en gran medida a los avances significativos en las tecnologías de la información. Debido a que permiten tener un acceso fácil a una gran fuente de información, realizan procesos rápidos y fiables de todo tipo de datos, tienen una gran capacidad de almacenamiento, son interactivas y proporcionan la digitalización de la información.

Las nuevas tecnologías, en este caso, las tecnologías de la información y las comunicaciones deben usarse para distribuir información alternativa que sea novedosa de forma que llegue a millones de personas de manera casi inmediata y sin las tradicionales limitaciones impuestas por la distancia, para crear nuevas vías de organizarse y coordinar, nuevas estructuras y modalidades de operación, para potenciar nuevas formas de solidaridad; y nuevas formas de compartir experiencias, tributando a la gestión del conocimiento.

Hoy en día las organizaciones influenciadas por los cambios que ocurren en su entorno turbulento, tanto en la esfera económica, como en la política y social, y fundamentalmente los cambios que son provocados por las tecnologías de la información y las comunicaciones, han incrementado significativamente el nivel de incertidumbre en las organizaciones, teniendo que rediseñar su paradigma de actuación, por lo que se impone un cambio en sus estilos de gestión, que les permita alcanzar vías más efectivas para obtener una mejora continua en la realización de sus actividades a partir de la cual se obtiene, despliega y utiliza una variedad de recursos básicos para apoyar los objetivos de las organizaciones.

A esto se integra el sector económico, ya que el desarrollo vertiginoso de la tecnología ha provocado que las clásicas gestiones empresariales estén involucradas en procesos de constante evolución. Las competencias entre las grandes corporaciones han conllevado a que surjan nuevas exigencias en las formas de llevar a cabo las técnicas contables que se han ido automatizando. Por esta razón a los departamentos de contabilidad y economía se les ha hecho más sencillo el trabajo, optimizando tiempo y esfuerzo en los procesos que se realizaban de forma manual.

A nivel mundial se han realizado varios sistemas informáticos que posibilitan el desarrollo de múltiples procesos en las compañías y empresas, un caso particular lo constituyen los que gestionan los

procesos de nóminas que se llevan a cabo en estas instituciones. Por disímiles temas existe una gran desigualdad entre las naciones, el acceso a estos tipos de software está restringido a aquellos que no posean una vasta economía que implica capital para comprar y mantener los privilegios que proporciona este tipo de tecnología.

Se ha reconocido que el siglo XXI es el siglo de la automatización y la aplicación de esta técnica en el desarrollo de las distintas actividades; la economía cubana ha recorrido un intenso camino de transformaciones en estos años, lo cual ha permitido que la automatización haya adquirido un gran desarrollo e importancia en el trabajo contable de las entidades presupuestadas y empresariales. Los Recursos Humanos es uno de los departamentos en el que sin lugar a dudas es indispensable la puesta en práctica de esta técnica, pues en él se realizan diariamente procesos que suelen ser muy complejos cuando no se cuenta con un sistema de software. Entre los procesos más importantes que se realizan en este departamento se encuentra la gestión de los Submayores, tanto de vacaciones como de retenciones, en ellos se registrarán una gran cantidad de información correspondiente a cada uno de los trabajadores existentes en la entidad, que de no contar con un sistema que facilite la entrada y el cálculo de los datos, se tornaría muy difícil por las complejas cuentas que en él se realizan.

Cuba cuenta con algunos software que gestionan los procesos de los submayores en las empresas, su inconveniente radica en que tratan los procesos que ocurren dentro del área de Capital Humano desligados con respecto a los que se llevan a cabo en Contabilidad. Esto provoca que la información se encuentre duplicada en los departamentos, facilitando la contaminación de la misma y creando condiciones favorables para que se cometan fraudes económicos. Además trae consigo que se atrasen las tareas que se desarrollan en las entidades, provocando el descontento por parte de los trabajadores.

Por lo que se puede decir que la correcta automatización de esta actividad facilita el desarrollo de las fases de procesos contables con una gran productividad, posibilitando la realización de todos los cálculos con una mayor rapidez, exactitud y eficiencia, brindando un incremento en la confiabilidad de las operaciones.

Teniendo en cuenta lo expuesto anteriormente se define el siguiente problema:

¿Cómo contribuir a los procesos de gestión de los Submayores de vacaciones y de retenciones en las entidades presupuestadas y empresariales?

Se define como objeto de estudio: Los procesos de gestión del Capital Humano para las entidades presupuestadas y empresariales y como campo de acción: La gestión de los Submayores de vacaciones y de retenciones existentes en el proceso de Remuneración y nómina.

Planteando como objetivo general de la investigación: Analizar y diseñar los Submayores de vacaciones y de retenciones para facilitar la gestión de los Recursos Humanos en las entidades presupuestadas y empresariales.

Considerando como idea a defender: La elaboración del análisis y el diseño de los Submayores de vacaciones y de retenciones, contribuirá a la gestión de estos procesos en las entidades presupuestadas y empresariales.

Para dar solución al objetivo propuesto se definieron las siguientes tareas de la investigación:

1. Estudiar el estado del arte de los procesos de gestión de los Submayores de vacaciones y de retenciones en las entidades presupuestadas y empresariales para diagnosticar su funcionamiento.
2. Determinar las herramientas que respondan a la automatización del proceso de gestión de los Submayores de vacaciones y de retenciones.
3. Modelar el negocio referente a los procesos de gestión de los Submayores y llevar a cabo el levantamiento de requisitos necesarios para ello.
4. Analizar y diseñar los procesos de gestión de los Submayores de vacaciones y de retenciones para dar solución al problema planteado.
5. Validar los requisitos para conocer si su especificación es satisfactoria.

Estructura del trabajo de diploma:

El presente trabajo de diploma está conformado por tres capítulos, en el Capítulo 1 se hace un estudio de temas relacionados con la investigación, se abordan metodologías y patrones que sirven para la realización del análisis y el diseño, y se caracterizan las posibles herramientas que se usarán para dar

solución al problema planteado. En el Capítulo 2 se realiza la modelación del negocio, lo cual brinda una base para conocer cuáles serán las actividades fundamentales que se automatizarán y a partir de las funcionalidades identificadas se realizará la propuesta del sistema. En el Capítulo 3 se realiza el análisis y el diseño del sistema para confeccionar los diagramas correspondientes al diseño, se realiza una breve descripción de los diagramas y se muestra el diagrama de componentes, lo que le permitirá a los futuros desarrolladores un trabajo más fácil y claro, así como la obtención de los resultados esperados. En el capítulo 4 se realiza la validación de los requisitos a través de dos técnicas y se llevan a cabo tres métricas que permitirán conocer si los requisitos del sistema pueden ser aceptados o rechazados, además de las métricas orientadas a clases para validar el diseño. Lo que contribuye a validar los resultados finales de esta investigación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se abarcan un grupo de conceptos y teorías muy importantes relacionados con el negocio a modelar, entre los que se pueden encontrar Capital Humano, Remuneración y nómina, Submayores, Submayor de vacaciones y Submayor de retenciones. Se mencionan las características de algunos de los sistemas de gestión existentes en el país y a nivel mundial, además se muestra el modelo de desarrollo, las herramientas y lenguajes propuestos para el desarrollo del sistema, lo que contribuirá al enfoque teórico hacia el cumplimiento de los objetivos propuestos.

1.1. Sistemas de gestión

“Un sistema de gestión es una estructura probada para la gestión y mejora continua de las políticas, los procedimientos y procesos de la organización” (1). Ayudan a lograr los objetivos de la organización mediante una serie de estrategias, que incluyen la optimización de procesos, el enfoque centrado en la gestión y el pensamiento disciplinado. Su implementación puede ayudar a gestionar los riesgos sociales, medioambientales y financieros, mejorar la efectividad operativa, reducir costos, aumentar la satisfacción de clientes y partes interesadas, proteger la marca y la reputación, lograr mejoras continuas, potenciar la innovación, eliminar las barreras al comercio y aportar claridad al mercado. Después de haber realizado un estudio de algunos de los sistemas de gestión existentes a nivel internacional y nacional, se muestran a continuación aquellos que fueron analizados con mayor detalle.

1.1.1. Sistemas de gestión internacionales

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, 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.

Capítulo 1. Fundamentación Teórica

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 los Submayores son las siguientes:

- Pagos de vacaciones
- Reintegros de salario, subsidio y vacaciones
- Submayor de vacaciones
- Submayor de retenciones
- Registro de salario y tiempo de servicio
- Pagos de subsidios (2)

Debido a que el ASSETS es un software extranjero, conlleva a comprar todos los derechos para poder utilizarlo, además de estar desarrollado sobre herramientas propietarias y teniendo en cuenta las necesidades del país, este sistema no puede ser utilizado como solución del problema expuesto.

SAGE MAS 500

Sage Software fue quien desarrolló el sistema Sage MAS 500, es una empresa que radica en California EEUU. Este sistema ofrece servicios de automatización de los procesos de Recursos Humanos, c, además de permitir realizar un seguimiento de la asistencia, los empleados y de la nómina.

Capítulo 1. Fundamentación Teórica

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, 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.

Este módulo 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.

Este software está desarrollado sobre herramientas privadas lo que implica que de ser utilizado debe comprarse la licencia de todos los productos que necesita para su correcto funcionamiento, además de que para cumplir con todos sus funciones deben ser comprados todos sus módulos. Las características negativas del sistema van en contra de las políticas de independencia tecnológica del país, lo que implica que no sea una solución óptima para el problema en cuestión. (3)

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

Capítulo 1. Fundamentación Teórica

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.

Entre las características que presenta este sistema es que 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 está vigente. Permite procesar un número ilimitado de conceptos y la codificación de los mismos 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). Posibilita los conceptos fijos por trabajador, el centro de costo, la distribución de moneda y las cartas de depósitos bancarios. Procesar múltiples contratos de nómina siendo la definición de estas determinada por el usuario (semanal, quincenal, mensual, confidencial, contratado, etc.). Maneja diferentes turnos de trabajo los cuales serán igualmente determinados por el usuario (fijos, rotativos, eventuales, planificada por mes/año). Admite 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 permita incluir variaciones de nómina a futuro (no se debe esperar a último momento) o en el momento que ocurren y se procesarán dependiendo de la fecha. (4)

Este sistema no cumple con las exigencias requeridas para darle solución al problema planteado, por ser un producto extranjero no certificado. Además no integra los Recursos de Humanos con la nómina, siendo este uno de los principales objetivos del presente trabajo.

1.1.2. Sistemas de gestión nacionales

VERSAT

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, está 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.

“El VERSAT-Sarasola es un sistema integrado de gestión económica diseñado para permitirle a los directivos, analizar, controlar y evaluar los resultados de su negocio en tiempo real, al contar con un instrumento seguro, rápido, eficaz y de fácil manejo para la planificación, control y el análisis de la gestión económica y financiera. Permite llevar el control y el registro contable individual de todos los hechos económicos que se originan en las estructuras internas de las entidades y permite obtener los estados financieros y los análisis económicos y financieros en estos niveles. Se estructura en un grupo de subsistemas, donde se procesan y contabilizan documentos primarios y se anotan los movimientos de los recursos materiales, financieros y laborales que se utilizan en una entidad.

El subsistema de nóminas es lo suficientemente adaptable 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. Da tratamiento a cuatro modalidades de pagos a trabajadores, por tarifa, horaria, diaria, como sueldista, pago libre, vacaciones. Permite la configuración de impuestos, bonificaciones, condiciones anormales, retenciones, descuentos y penalizaciones.

Requerimientos técnicos:

Servidor

Pentium IV 128 MB (megabyte) RAM (Memoria de acceso aleatorio)

40Gigabyte de HDD (Disco duro)

Windows 2000 ó superior

Estaciones de trabajo

Pentium III, ó superior 64 MB RAM

Capítulo 1. Fundamentación Teórica

4 Gigabyte de HDD

Windows 2000 ó superior (5)

VERSAT-Sarasola realiza operaciones útiles y que resuelven las necesidades que se quieren satisfacer con el presente trabajo, pero no deja de ser un software desarrollado sobre herramientas propietarias y al igual que los software analizados anteriormente su licencia debe ser comprada, además de tener que comprar este sistema completo, pues comprar solo algunos módulos no es de mucha ayuda ya que necesita de la integración de todos para su correcto funcionamiento.

Rodas XXI

“Rodas XXI es un sistema multiempresa 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 de forma independiente. Cada uno de los módulos del sistema está diseñado para trabajar en un entorno de red e intercambiar información mediante correo electrónico o disquetes.

Las posibilidades que ofrece Rodas es que permite manejar un número ilimitado de empresas, puede instalarse tanto en estaciones de trabajo como en redes locales, sus módulos pueden ajustarse fácilmente a las características de cada usuario, permite el intercambio automático de los comprobantes generados por cada módulo con el de contabilidad, trabaja con doble moneda, crea reportes fácilmente, permite la visualización y la impresión de los reportes de forma opcional, la información está protegida por claves, lleva un registro de las operaciones relacionadas con el sistema permitiendo realizarles auditorías, todos los módulos cuentan con una ayuda en línea, un manual de usuarios detallado, así como con un CD tutorial el cual le ofrecerá la oportunidad de aprender a trabajar con un potente paquete económico. Apoyándose en las bondades que ofrece la multimedia, estos tutoriales le permitirán adentrarse dentro del sistema Rodas XXI guiado por textos e ilustraciones explicativas.” (6)

El módulo de nóminas está diseñado para satisfacer las necesidades de las entidades cubanas ya que cumple con las regulaciones establecidas en los sistemas de pagos en el país. Además es compatible con regulaciones específicas que establecen los distintos organismos de la administración central del estado para sus entidades debido a su gran flexibilidad y adaptabilidad a distintos sistemas de pago siempre dentro del marco de lo establecido por el Ministerio del Trabajo y Seguridad Social y brinda

Capítulo 1. Fundamentación Teórica

grandes facilidades a las entidades para el cálculo y emisión de sus nóminas. Se pueden calcular y emitir todos los tipos de nóminas que se utilizan en Cuba.

Para la obtención de las nóminas en cada uno de los períodos del año el sistema divide las operaciones en dos pasos fundamentales, el cálculo de las 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 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 le permitirá además visualizar información correspondiente a períodos anteriores, tan sólo con cambiar de período contable a períodos anteriores ya cerrados, aunque en dichos períodos no podrá realizar ninguna operación. Esta característica es compartida por todos los módulos de Rodas XXI (6)

“Requerimientos técnicos:

Computadora IBM PC o compatible.

Pentium III o superior.

256 MB de memoria RAM o más.

Unidad de disco flexible, un disco duro, lector de CD (Disco compacto) y sistema operativo Windows 2000/XP.

20 GB (Gigabytes) de espacio en disco duro.” (6)

El Rodas XXI tiene funcionalidades útiles, pero sigue siendo un sistema desarrollado sobre herramientas privativas, lo cuál afecta la independencia tecnológica que el país quiere lograr; además no integra los Recursos Humanos con la nómina. Esto conlleva a que no sea una solución óptima para dar respuesta al problema planteado.

1.2. Marco Teórico

1.2.1. Capital Humano

“Capital Humano es el conjunto de conocimientos, experiencias, habilidades, sentimientos, actitudes, motivaciones, valores y capacidad para hacer, portados por los trabajadores para crear más riquezas con eficiencia. Es, además, conciencia, ética, solidaridad, espíritu de sacrificio y heroísmo.” (7)

1.2.2. Remuneración y nómina

“Remuneración es la 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. “ (7)

“La nómina es el documento donde se va a 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.” (7)

Por lo antes planteado se puede decir entonces que:

Capítulo 1. Fundamentación Teórica

Remuneración y nómina es la estructura de cargos, políticas y retribución en función de prioridades y objetivos vinculados a la situación financiera y presupuestaria del Instituto, así como la distribución justa y equitativa de los salarios con respecto a los cargos.

1.2.2.a 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. Existen múltiples tipos de Submayores, pero solo se hará referencia a los estudiados en el presente trabajo, ellos se clasifican en:

Submayores de vacaciones

Es la forma de registrar y controlar por trabajador, el tiempo e importe acumulado para las vacaciones. Es un documento que será de uso obligatorio para todas las entidades, además de que cada entidad definirá las áreas productoras del documento y el personal facultado para emitir el mismo, así como la distribución de copias, si procede, acorde a sus características organizativas.

Submayores de retenciones

Es el proceso que se efectúa afectando periódicamente, según se determine mensual, quincenal o semanal, el salario devengado por los trabajadores, por adeudos contraídos por los mismos cuando se reciban los documentos primarios autorizados por sentencia judicial o notificación de retenciones.

1.3. Modelo de desarrollo orientado a componente

El Modelo de desarrollo orientado a componentes es un modelo que busca dentro de otros objetivos, reducir el tiempo de trabajo, el esfuerzo que requiere implementar una aplicación y los costos del proyecto. Pertenece al paradigma de programación de sistemas abiertos, los cuales son extensibles y tienen una interacción con componentes heterogéneos que ingresan o abandonan el sistema de forma dinámica, es decir que los componentes pueden ser remplazados, por otros independientemente de su arquitectura y desarrollo.

Un componente se puede decir que es una parte no trivial, casi independiente, y reemplazable de un sistema que llena claramente una funcionalidad dentro de un contexto en una arquitectura bien definida. Este provee la realización física por medio de un conjunto de interfaces. Para que un elemento pueda ser catalogado como componente es necesario que sea identificable para tener una

Capítulo 1. Fundamentación Teórica

identificación que permita acceder fácilmente a sus servicios y que permita su clasificación. Auto contenido, es decir un componente no debe requerir de la utilización de otros para finalizar la función para la cual fue diseñado. Puede ser remplazado por otro componente, nuevas versiones u otro componente que lo reemplace y mejore. Con acceso solamente a través de su interfaz, debe asegurar que estas no cambiaran a lo largo de su implementación. Sus servicios no varían, las funcionalidades ofrecidas en su interfaz no deben variar, pero su implementación sí. Bien documentado, un componente debe estar correctamente documentado para facilitar su búsqueda si se quiere actualizar, integrar con otros o adaptarlo. Es genérico, sus servicios debe servir para varias aplicaciones. Reutilizado dinámicamente, puede ser cargado en tiempo de ejecución en una aplicación. Independiente de la plataforma hardware o software.

Los componentes se pueden clasificar por el tamaño, la complejidad, las métricas de complejidad, el mantenimiento, la reusabilidad, la frecuencia de re-uso y la confiabilidad. En este modelo de desarrollo en el análisis y el diseño se generan los artefactos de plan de trabajo individual, modelo de procesos de negocio, descripción de procesos de negocio, modelo conceptual, prototipo de IU (Interfaz usuario), especificación de requisitos, casos de prueba, diagrama de clases, descripción del diseño de clases. (8)

Por ser este un modelo de desarrollo que está orientado a componentes y por realizar el modelado del negocio a través de procesos. Se define este modelo para guiar la construcción de los distintos artefactos en cada uno de los flujos de trabajo.

1.4. Lenguajes de modelado

La complejidad de los sistemas informáticos representa un reto importante para los ingenieros y arquitectos del software. Se ha pasado de la definición de la estructura y calidad del código final, 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 es decir, con una sintaxis y semántica precisa, y que puede ser interpretado y manipulado por un ordenador.”

(9)

1.4.1. Notación para el modelado de los procesos del negocio (BPMN)

BPMN (Notación para el modelado de los procesos del negocio) es un lenguaje que permite proporcionar una notación que sea fácilmente comprensible por todos los usuarios del negocio, desde los analistas del negocio, los desarrolladores responsables de la implementación de la aplicación, hasta las personas que interactúan con el negocio administrando y supervisando estos procesos. Además crea un puente estandarizado entre el diseño de procesos de negocio y su implementación; y asegura que los lenguajes para la ejecución de procesos de negocio puedan ser visualizados con una notación común.

BPMN especifica un diagrama del proceso de negocio único, llamado el Diagrama de Procesos de Negocio (BPD, por sus siglas en inglés). BPD se compone de un conjunto de elementos gráficos. Estos elementos permiten el fácil desarrollo de sencillos diagramas que será familiar a la mayoría de los analistas del negocio. Cabe destacar que uno de los motores para el desarrollo de BPMN es crear un mecanismo sencillo para la creación de modelos de procesos de negocio, mientras que al mismo tiempo debe ser capaz de manejar la complejidad inherente a los procesos de negocio. El enfoque adoptado para tratar estos dos requisitos fue organizar la gráfica de los aspectos de la notación en categorías específicas. Esto proporciona un conjunto de categorías de notación para que el que utilice un BPD pueda reconocer fácilmente los tipos básicos de elementos y comprender los diagramas. Las cuatro categorías básicas de elementos son el flujo de objetos, los conectores, las swimlanes (carriles de piscina) y los artefactos.

El proceso de modelado del negocio es usado para comunicar una amplia variedad de información a diferentes audiencias. BPMN está diseñado para cubrir muchos tipos de modelación y permite la creación de segmentos de procesos, así como el desarrollo de procesos del negocio de inicio a fin a diferentes niveles de confiabilidad. Dentro de la variedad objetivos del proceso de modelado, hay dos tipos básicos de modelos que se pueden crear con un BPD: los procesos de colaboración B2B (Público) y los procesos de negocio interno (Privado).

Procesos de colaboración B2B

Un proceso de colaboración B2B describe las interacciones entre dos o más entidades empresariales. Los diagramas para este tipo de procesos son generalmente de un punto de vista global. Es decir, no toman en cuenta la opinión de ningún participante en particular, pero muestran la interacción entre los participantes. Las interacciones se representan como una secuencia de actividades y el intercambio de mensajes patrones entre los participantes. Las actividades para los participantes en la colaboración puede ser considerado como el "punto de contacto" entre ellos, por lo que el proceso define las interacciones que son visibles al público para cada uno de los participantes. Al mirar el proceso que se muestra en una sola Piscina (es decir, para uno de los participantes), el proceso público también se le llama proceso abstracto. El proceso de negocio interno es propenso a tener más actividades y detalles de los que se muestran en los procesos de colaboración B2B.

Procesos de negocio interno

El proceso de negocio interno se centra generalmente en el punto de vista de una sola organización empresarial. Aunque los procesos internos a menudo muestran las interacciones con los participantes externos, que definen la actividades que no son visibles al público y, por lo tanto, las actividades privadas. Si se utilizan calles entonces un proceso de negocio interno será dentro de una sola piscina. La secuencia del flujo del proceso es, por tanto, dentro de la piscina y no pueden cruzar las fronteras de la misma. El flujo de mensajes puede cruzar la frontera para mostrar las interacciones que existen entre procesos del negocio separados. Así, un simple diagrama de procesos de negocio puede mostrar múltiples procesos de negocio privado. (10)

Teniendo en cuenta que BPMN permite el modelado por procesos, donde se representan gráficamente todas sus etapas, con una notación fácil de leer y de entender por todos los involucrados en el negocio, se decide utilizar este modelo para realizar el modelamiento del negocio.

1.4.2. Lenguaje unificado de modelado (UML)

“UML es un lenguaje estándar que sirve para escribir los planos del software, puede utilizarse para visualizar, especificar, construir y documentar todos los artefactos que componen un sistema con gran cantidad de software. Puede usarse para modelar desde sistemas de información hasta aplicaciones distribuidas basadas en Web, pasando por sistemas de tiempo real. Es solamente un lenguaje por lo que, es sólo una parte de un método de desarrollo software, es independiente del proceso aunque para

Capítulo 1. Fundamentación Teórica

que sea óptimo debe usarse en un proceso dirigido por Casos de Uso, centrado en la arquitectura, iterativo e incremental.

Este lenguaje de modelado ayuda a interpretar grandes sistemas mediante gráficos o mediante texto obteniendo modelos explícitos que ayudan a la comunicación durante el desarrollo ya que al ser estándar, los modelos podrán ser interpretados por personas que no participaron en su diseño (e incluso por herramientas) sin ninguna ambigüedad. En este contexto, sirve para especificar modelos concretos no ambiguos y completos.

Debido a su estandarización y su definición completa no ambigua, se puede conectar de manera directa a lenguajes de programación como Java, C++ o Visual Basic, esta correspondencia permite lo que se denomina como ingeniería directa (obtener el código fuente partiendo de los modelos). Además de que tiene la capacidad de modelar actividades de planificación de proyectos y de sus versiones, expresar requisitos y las pruebas sobre el sistema, representar todos sus detalles así como la propia arquitectura. Mediante estas capacidades se obtiene una documentación que es válida durante todo el ciclo de vida de un proyecto. “ (11)

UML proporciona una gran flexibilidad y expresividad a la hora de modelar sistemas, característica que entre otras, hacen que este lenguaje se utilice para llevar cabo el flujo de análisis y diseño del sistema que se quiere modelar.

1.5. Herramienta de modelado

En los últimos años el desarrollo del software ha generado que surjan herramientas que permitan a los analistas y diseñadores de software realizar estos procesos de forma eficiente, con mayor calidad y fiabilidad. Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son un ejemplo de ello. Estas son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero; pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

1.5.1. Visual Paradigm para UML 6.1

“Visual Paradigm es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Las características del Visual Paradigm es que tiene un soporte de UML versión 2.1, modelado colaborativo con sistemas de control de versiones (CVS) y subversión, diagramas de flujo de datos, generación de bases de datos, transformación de diagramas de entidad-relación en tablas de base de datos, generador de informes para generación de documentación, distribución automática de diagramas, reorganización de las figuras y conectores de los diagramas UML y editor de figuras.

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.” (12)

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 las notaciones definidas por BPMN. Estas características permiten que esta herramienta sea óptima para realizar el modelado del software.

1.6. Lenguaje de programación web

Un lenguaje de programación web está construido por hiertextos y es un entramado de páginas conectadas con enlaces. Estas páginas no solo se limitan a presentar textos y enlaces, sino que

también puede ofrecernos imágenes, videos, sonido y todo tipo de presentaciones, llegando a ser el servicio más rico en medios que tiene Internet. (13)

1.6.1. PHP 5.2 o Superior

“PHP (Procesador Interpretado) es un lenguaje de código abierto interpretado, de alto nivel, embebido en páginas HTML (Lenguaje de Marcado Interpretado) y ejecutado en el servidor. Se trata de un lenguaje de script (o de guiones), diseñado para, entre otras cosas, aumentar el dinamismo de las páginas web. La mayor parte de su sintaxis ha sido tomada de los lenguajes de programación C, Java y Perl con algunas características específicas de sí mismo.

La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. No es un lenguaje de marcas como podría ser HTML, XML (Lenguaje Etiquetado Extensible) o WML (Lenguaje de Marcado Inalámbrico). Está más cercano a los lenguajes de programación JavaScript o C.

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas. Algunas de estas características permiten: el soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL y Informix. Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) y analizar código XML. Ofrece una solución simple y universal para las paginaciones dinámicas de la Web. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.” (14)

Por ser PHP un lenguaje libre, lo que representa una alternativa de fácil acceso para todos, multiplataforma, además de todas las características expuestas con antelación, se escoge como lenguaje de programación para la implementación futura del sistema.

1.7. Zend Studio

“Zend Studio es un editor web orientado a la programación de páginas PHP, con ayudas en la gestión de proyectos y depuración de código. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene la interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache¹ y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración.” (15)

Las características de Zend Studio permiten la evaluación de comentarios de documentos PHP, permiten el auto completado de código, tienen un soporte integrado para Subversión y un visor de la estructura de base de datos y de contenidos, cliente SQL para diferentes tipos de bases de datos. Por estas y otras se decidió utilizar Zend Studio como entorno de desarrollo.

1.8. Gestor de base de datos

Los sistemas de gestión de base de datos (SGBD); (en inglés: DataBase Management System, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. En general un sistema de gestión de bases de datos es una aplicación que permite trabajar con bases de datos, definir, insertar, eliminar, consultar, ordenar y filtrar información. (16)

1.8.1. PostgreSQL 8.3 o Superior

“PostgreSQL es un poderoso sistema manejador de base de datos, es decir, un sistema diseñado para administrar grandes cantidades de datos, que tiene la fama de ser la base de datos de código abierto más avanzada del mundo.

PostgreSQL se ha preocupado por ser una solución real a los complejos problemas del mundo empresarial y a la vez mantener la eficiencia al consultar los datos. Con este fin, se han desarrollado y añadido al postgre las más interesantes y útiles características que antes solo podían hallarse en sistemas manejadores de base de datos comerciales.” (17)

¹ Apache: Servidor Web.

Capítulo 1. Fundamentación Teórica

Es altamente escalable, tanto en la enorme cantidad de datos que puede administrar como en el número de usuarios concurrentes que puede acomodar. Por estas características se define como gestor de base de datos para el desarrollo del sistema al PostgreSQL.

1.9. Servidor web

Un servidor web es un programa que implementa el protocolo HTTP (HyperText Transfer Protocol). Este protocolo pertenece a la capa de aplicación del modelo OSI y está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

Es un programa que se ejecuta continuamente en un ordenador (también se emplea el término para referirse al ordenador que lo ejecuta), manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error. (18)

1.9.1. Apache

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

Capítulo 1. Fundamentación Teórica

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. (19)

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

Conclusiones parciales del capítulo

En este capítulo se trataron conceptos muy importantes con el objetivo de tener un mejor entendimiento del negocio. Se realizó un estudio de algunos de los sistemas de gestión que presentan características similares al que se quiere construir, teniendo en cuenta sus distintas funcionalidades, así como los inconvenientes que presentan y el porque no pueden ser considerados como una propuesta de solución. Y por último se presentan las características del modelo de desarrollo, las herramientas y lenguajes a utilizar, los cuales contribuirán al desarrollo del sistema de software que se quiere construir.

CAPÍTULO 2: MODELAMIENTO DEL NEGOCIO

En el presente capítulo se hace una descripción del negocio, se identifican y describen los procesos, dando paso a la construcción del Modelo conceptual y a la descripción de las distintas entidades que se relacionan, se capturan y especifican los requisitos funcionales, los cuales deberán presentar la solución del sistema que se quiere construir. Se muestran las técnicas empleadas para la captura de requisitos y los patrones utilizados para la agrupación de los mismos. También se hace referencia a los prototipos de interfaz de usuario que muestran una primera versión del producto que se quiere construir.

2.1. Técnicas para el levantamiento de requisitos

Existen técnicas y herramientas para el levantamiento de requisitos que permiten obtener, documentar y desarrollar los requisitos de una forma más sencilla. Muchas de estas técnicas son fundamentales para capturar los requisitos de software, permitiendo obtener la mayor cantidad de información del sistema que se quiere automatizar. A continuación se muestran y explican de forma breve las empleadas para la captura de los requisitos.

Entrevistas: Las entrevistas se emplean para reunir información proveniente de personas o de grupos. Durante la entrevista, el analista conversa con el encuestado para obtener información deseada. Las entrevistas se emplean para recolectar información en forma verbal, a través de preguntas elaboradas por el analista, estas preguntas se les realiza a quienes se encuentren afectados 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 la realización de las entrevistas se debe coordinar previamente la fecha y hora, y realizar un plan de agenda, en el cual se realiza un punteo del objetivo de la entrevista y se prepara un cuestionario.

Arqueología de documentos: Con la aplicación de esta herramienta se tratan de determinar posibles requerimientos sobre la base de inspeccionar la documentación utilizada por la empresa. Esta herramienta sirve más que nada como complemento de las demás técnicas, y ayuda a obtener información que de otra manera sería sumamente difícil conseguir. Se debe recolectar cualquier formulario o documento que sea utilizado para registrar o enviar información.

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.

Glosario: El glosario es una simple lista de términos en donde se explica su significado. En esta lista se incluyen y definen todos los términos que requieren explicación, mejorando así la comunicación intergrupala y la comunicación con el cliente, y mitigando el riesgo de malos entendidos. Los términos que se incluyen provienen de todas las áreas del proyecto.

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. 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. (20)

2.2. Patrones

“Un patrón es una descripción de un problema y la solución, a la que se da un nombre, y que se puede aplicar a nuevos contextos; idealmente, proporciona consejos sobre el modo de aplicarlo en varias circunstancias, y considera los puntos fuertes y compromisos. (21). A continuación se muestra el patrón utilizado para la agrupación de los requisitos.

2.2.1. Patrones CRUD

“El patrón CRUD (Create, Read, Update, Delete [Crear, Leer, Actualizar, Eliminar]) se utiliza en los casos donde se quiere realizar altas, bajas, cambios y consultas a alguna entidad del sistema. Su nombre es un acrónimo de las palabras crear, leer, actualizar y eliminar.

Este patrón se clasifica en:

CRUD Completo

CRUD Completo consiste en un Caso de Uso para administrar la información, permitiendo modelar las diferentes operaciones para administrar una entidad de información, tales como crear, leer, cambiar y dar de baja.

CRUD Parcial

CRUD Parcial permite que algunas de las alternativas del Caso de Uso puedan ser modeladas como Caso de Uso independiente. Este patrón es preferible cuando uno de los flujos alternativos del Caso de Uso es más significativo, muy largo, o mucho más complejo que el patrón.” (22)

2.3. Descripción del negocio

Dentro del proceso de nóminas de todas las entidades laborales de la economía cubana, ya sean pertenecientes al sector empresarial como al presupuestado, el proceso de apertura, actualización y cierre de los Submayores de vacaciones y retenciones desempeñan un papel primordial por la incidencia directa que tienen en los importes por concepto de salario que devengan los trabajadores. Actualmente se encuentran en explotación un sinnúmero de soluciones informáticas desarrolladas para un entorno específico que en cierta medida logran satisfacer las necesidades para una determinada actividad, pero que no se adaptan a todas las peculiaridades y características de la amplia gama de centros laborales del país.

A raíz de las investigaciones y estudios realizados se concluyó en la necesidad de desarrollar una solución informática que permita de forma flexible, segura y con la calidad requerida la automatización de los procesos de apertura, actualización y cierre de los Submayores de vacaciones y retenciones y que cumpla con todas las exigencias y requerimientos establecidos en la legislación laboral vigente.

La Gestión del Submayor de vacaciones se inicia con el proceso de su apertura para cada trabajador de nuevo ingreso a la entidad laboral, en la gran mayoría de los casos el tiempo y los importes acumulados de las vacaciones son nulos, aunque en situaciones excepcionales se autoriza por la legislación vigente la transferencia de los días e importes acumulados de los trabajadores y que no fueron liquidados en sus entidades de procedencia. A medida que los trabajadores van desarrollando su vida laboral se va creando una provisión de vacaciones consistente en la acumulación del 9,09 % del tiempo y los importes devengados por concepto de salario, estos valores se van actualizando en el Submayor de vacaciones para cada período de pago hasta que el trabajador se acoge a su derecho de disfrutar de las vacaciones anuales pagadas en que los días a descansar con sus respectivos importes son descontados del Submayor de vacaciones. Una vez que el trabajador causa baja de la entidad laboral independientemente de la causa que la genere tiene derecho a la liquidación de todos los importes acumulados hasta esa fecha y debe procederse al cierre del Submayor de vacaciones.

Capítulo 2. Modelamiento del Negocio

La apertura del Submayor de retenciones tiene lugar cuando los trabajadores vinculados laboralmente a un centro de trabajo contraen obligaciones de adeudos y son presentadas las notificaciones y documentos primarios que autorizan los mismos. La apertura de este Submayor consiste en el registro para cada trabajador y cada tipo de retención de los saldos totales de la deuda, los importes a descontar en cada plazo y la cantidad de plazos en que se le realizarán los descuentos, existe un grupo de retenciones como pensión alimenticia, cuentas de ahorro y seguros para los cuales no es necesario registrar la cantidad de plazos ni el importe total del adeudo. A cada tipo de retención se le asigna un orden de prioridad en el que se le irán descontando los importes del salario.

En correspondencia con los períodos de pagos establecidos en las entidades laborales se efectúan los descuentos de las retenciones de las nóminas de salario, vacaciones, pagas por resultados y subsidios. Los importes retenidos a los trabajadores se contabilizan en cada Submayor hasta su entrega a los beneficiarios o la entidad bancaria correspondiente. El cierre del Submayor de retenciones procede cuando el trabajador concluya el pago de todo el adeudo contraído o cause baja de la entidad laboral. Los Submayores tanto el de vacaciones como el de retenciones tienen que estar actualizados y sus saldos deben corresponderse con las diferentes nóminas procesadas.

2.4. Reglas del negocio

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. El proceso de especificación implica que hay que “identificarlas” dentro del negocio, “evaluar” si son relevantes dentro del campo de acción que se está modelando e “implementarlas” en la propuesta de solución. A continuación se darán a conocer las reglas fundamentales que se siguen en el negocio.

Crear Submayores de retenciones: Para crear el Submayor de retenciones hay que tener en cuenta los tipos de adeudos mostrados anteriormente en la descripción del negocio. Por otro lado las notificaciones de retenciones pueden ser: notificación de descuentos a efectuar - agencia bancaria, oficio de embargo para pensiones alimenticias, resolución de la universidad.

Crear Submayores de vacaciones: Para crear el Submayor de vacaciones hay que tener en cuenta las regulaciones relativas a las vacaciones anuales pagadas. Existen causas que no acumulan tiempo

ni importe para el disfrute de vacaciones, estas causas pueden ser: enfermedad común y profesional, accidente común y profesional, ausencias injustificadas, ausencias por asuntos propios, personal en cursos de capacitación que cobran estipendios por este concepto, licencias sin sueldo, interrupciones laborales.

Los trabajadores acogidos al Decreto Ley 91 que devengan subvención económica, ésta no acumula descanso retribuido.

Las ausencias que no suspenden la acumulación para el disfrute del descanso son: licencias deportivas retribuidas, licencia por maternidad (se calcula por el importe del subsidio pagado), movilizaciones agro o micro, movilizaciones de las MTT (Milicias de Tropas Territoriales), licencia por actividades políticas y militares, licencia a trabajadores acogidos a cursos dirigidos, licencia de 2 o 3 días por fallecimientos de familiares, trabajadores internacionalistas civiles, trabajadores interrumpidos (Hasta 30 días acumulados por interrupción).

Casos especiales

En los casos de pago de horas extra y doble turno se acumula efectivo pero no tiempo.

El salario devengado por prima o vinculación acumula efectivo para el acumulado del descanso.

2.5. Procesos del negocio

“El proceso es la unión que mantiene juntas las capas de las tecnologías y permite un desarrollo racional y oportuno de la ingeniería del software. El proceso define el marco de trabajo para un conjunto de áreas claves que se deben establecer para la entrega efectiva de la tecnología de la ingeniería del software.” (23) A continuación se nombran los procesos que fueron identificados en el negocio.

Apertura de los Submayores

- Apertura del Submayor de vacaciones.
- Apertura del Submayor de retenciones.

Actualizar Submayores

- Actualizar Submayor de vacaciones.
- Actualizar Submayor de retenciones.

2.6. Descripción de los procesos del negocio

2.6.1. Apertura del Submayor de vacaciones

El proceso de **Apertura** se efectúa por el especialista de nómina, el cual es el encargado de adicionar y eliminar información al Submayor, así como asociar con una cuenta contable. Este proceso se realiza antes de iniciar el procesamiento de la nómina. Para garantizar su cuadro con los saldos de las cuentas de provisión de vacaciones.

2.6.2. Apertura del Submayor de retenciones

El proceso de **Apertura** se realiza por el especialista de nómina, esto ocurre cuando el trabajador contrae retenciones. Este Submayor permite llevar un control sobre la cantidad que se le va a ir descontando al trabajador de su salario todos los meses según los períodos de pagos que estén definidos en la empresa. Este proceso se va a repetir a un mismo trabajador cada vez que este contraiga una nueva retención. Es decir un trabajador va a tener tantos Submayores de retenciones como retenciones contraídas.

2.6.3. Actualizar Submayor de vacaciones

El proceso de **Actualizar** se efectúa cuando se confirma la nómina. Este proceso consiste en hacer los cálculos correspondientes para que los datos de los días e importe acumulado durante un período coincidan con la nómina emitida y poder de esta forma saber el acumulado actual en días e importe de cada trabajador. El encargado en contabilidad realiza los cálculos y actualiza el Submayor.

2.6.4. Actualizar Submayor de retenciones

El proceso de **Actualizar** se realiza cuando se confirma la nómina. Es decir, el responsable en contabilidad realiza los cálculos correspondientes para que los datos que están en el Submayor coincidan con los de la nómina y así evitar errores. Permitiendo de esta forma actualizar el Submayor.

2.7. Modelo conceptual

El modelo conceptual es una explicación de los conceptos significativos en un dominio del problema. A continuación se muestra el Modelo conceptual que representa las diferentes entidades existentes en el proceso de los Submayores del subsistema Capital Humano del sistema CedruX.

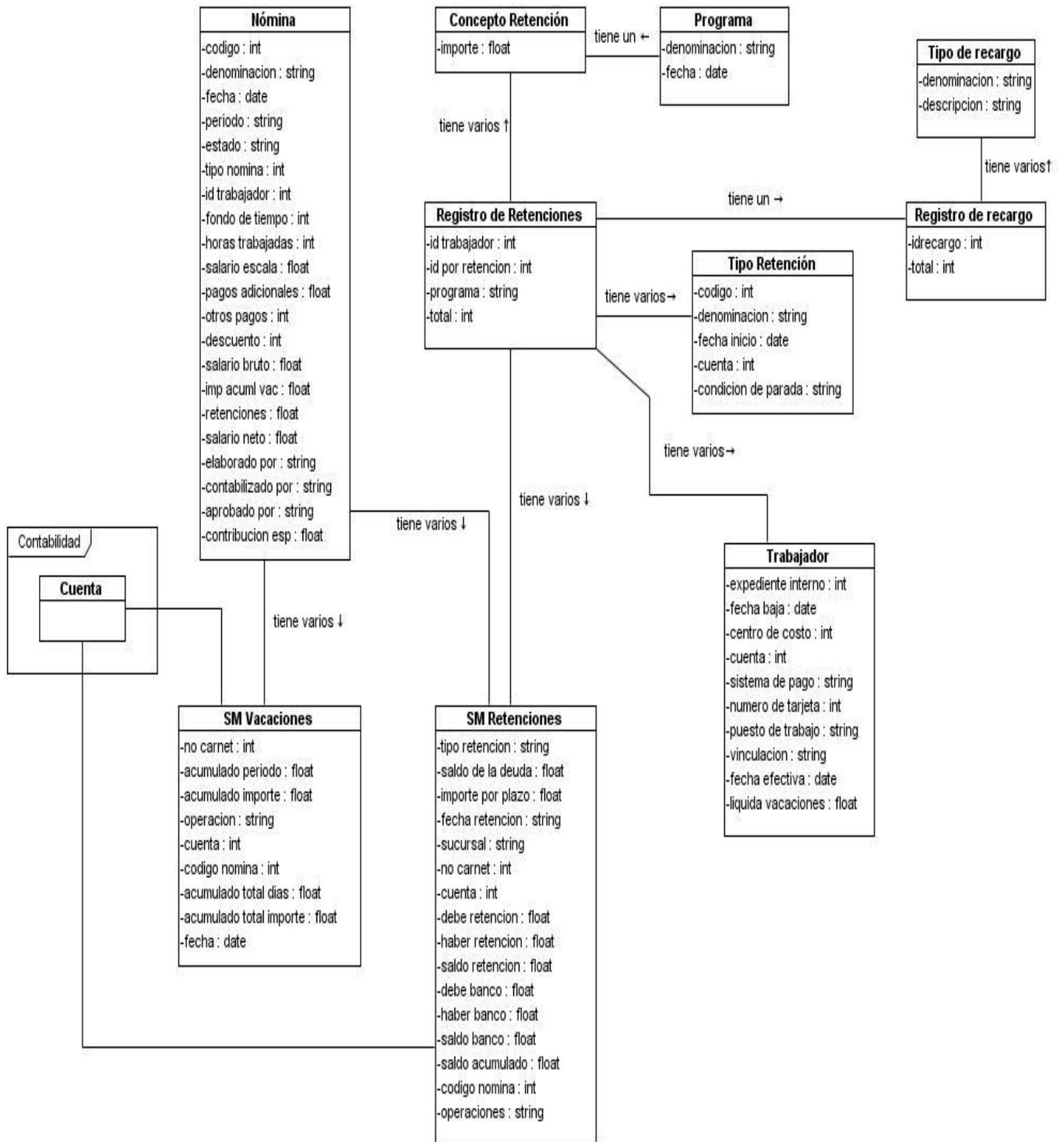


Fig.1. Modelo conceptual

Capítulo 2. Modelamiento del Negocio

La Figura 1 muestra el Modelo conceptual del componente Submayores del subsistema Capital humano del sistema Cedrux, donde se representan todas las entidades que lo forman. A continuación se describen las funcionalidades de cada una de estas entidades.

Concepto de retención: es donde se van a guardar todos los importes de las retenciones existentes en la entidad.

Registro de retenciones: es donde se van a mostrar todos los trabajadores de la entidad que tienen adeudos contraídos.

Programa: es el lugar donde se van a registrar los distintos tipos de programas de pagos definidos en la entidad, estos pueden ser, mensual, quincenal o semanal.

Tipo de retención: es donde se van a definir los distintos tipos de retenciones existentes en la entidad.

Tipo de recargo: es donde se van a definir todos los tipos de recargo que se van a aplicar en la entidad en un momento determinado.

Registro de recargo: es el lugar donde se mostrarán los trabajadores con los recargos que se le han aplicado a cada uno.

Trabajador: es donde se registrarán todos los trabajadores existentes en la entidad y todos aquellos de nuevo ingreso.

Submayor de vacaciones: es el lugar donde se registrarán el tiempo y el importe acumulado para las vacaciones en un período determinado.

Submayor de retenciones: es el lugar donde se registrarán los procesos que se llevan a cabo para afectar periódicamente, según se determine, mensual, quincenal o semanal el salario devengado por los trabajadores por adeudos contraídos por los mismos.

Nómina: es donde se van a relacionar todos los trabajadores de las entidades que persivan salarios y haberes.

Cuenta: es la unión de todos los valores referentes a una misma persona. En la cuenta se registrarán los aumentos y disminuciones de cada trabajador.

2.8. Requisitos funcionales del sistema

La IEEE Standard Glossary of Software Engineering Terminology define un requerimiento como condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. Por lo que se puede decir que los requerimientos deben ser especificados por escrito, claros y precisos. Estos se clasifican en funcionales y no funcionales. Los requisitos funcionales son capacidades o condiciones que el sistema debe tener, mientras que los no funcionales son propiedades o cualidades que el producto debe tener. A continuación se muestran los 35 requisitos identificados en el negocio:

ARF 1	Tipo de retención
	1.1. Adicionar tipo de retención 1.2. Modificar tipo de retención 1.3. Eliminar tipo de retención
ARF 2	Tipo de programa
	2.1. Adicionar tipo de programa 2.2. Modificar tipo de programa 2.3. Eliminar tipo de programa
ARF 3	Gestionar tipo recargo
	3.1. Adicionar tipo recargo 3.2. Modificar tipo recargo 3.3. Eliminar tipo recargo
ARF 4	Submayor de vacaciones
	4.1. Apertura del submayor de vacaciones 4.2. Cierre del submayor de vacaciones 4.3. Actualización del submayor de vacaciones 4.4. Detalles del submayor de vacaciones 4.5. Conciliación con contabilidad del submayor de vacaciones

	<ul style="list-style-type: none"> 4.6. Ajuste del submayor de vacaciones 4.7. Dar alta automática a un trabajador 4.8. Generar comprobante de operaciones dado una modificación
ARF 5	Configurar submayor de vacaciones
	5.1. Configuración del submayor de vacaciones
ARF 6	Submayor de retenciones
	6.1. Apertura del submayor de retenciones
	6.2. Modificar submayor de retenciones
	6.3. Cierre del submayor de retenciones
	6.4. Detalles del submayor de retenciones
	6.5. Actualizar submayor de retención
	6.6. Conciliación con contabilidad del submayor de retenciones
	6.7. Filtrar del submayor de retenciones
	6.8. Pago directo al banco del submayor de retenciones
	6.9. Alerta de cumplido con el pago de la retención de los trabajadores
	6.10. Adicionar recargo a la retención
	6.11. Modificar recargo a la retención
	6.12. Eliminar recargo a la retención
	6.13. Descuento adelantado de las retenciones
ARF 7	Fichero histórico devengado
	7.1. Mostrar fichero
	7.2. Modificar fichero
	7.3. Alta automática al fichero
	7.4. Resumen devengado del trabajador en meses anteriores Modificar fichero

Tabla 1. Agrupación de los requisitos funcionales (ARF)

2.9. Especificación de los requisitos

En el presente epígrafe se darán a conocer conceptos importantes y se especificarán los requisitos expuestos en el epígrafe anterior con el objetivo de guiar de forma correcta el desarrollo del sistema que se quiere construir.

2.9.1. Tipo de retención

En el requisito se gestionan los tipos de retenciones que se tendrán en una entidad, además de la relación que se establecerá entre una retención y un trabajador.

Especificación del requisito Adicionar tipo de retención.

Conceptos tratados	Conceptos	Atributos
	Tipo de retención.	Código, Denominación, Prioridad, Condición de parada, Cuenta, Fecha inicio.
Precondiciones	Precondiciones	Pre-requisito
	No procede.	No procede.
Descripción	<p>Se selecciona la opción tipo de retención.</p> <p>El sistema permite adicionar los siguientes datos de la retención:</p> <p>Código: número que identifica al tipo de retención.</p> <p>Denominación: nombre que va a recibir la retención.</p> <p>Prioridad: número que se le va a asignar a la retención para saber el orden que se va a seguir para atender cada una de las retenciones.</p> <p>Condición de parada: tiempo que va a tener la retención para ser pagada, esta condición puede ser por tiempo límite o por tiempo indefinido.</p> <p>Fecha inicio: es la fecha en que se va a comenzar a descontar esa retención al trabajador.</p> <p>Cuenta: se selecciona del nomenclador de cuenta de contabilidad.</p> <p>Se indica adicionar el tipo de retención.</p> <p>El sistema valida los datos.</p> <p>En caso de error en la entrada de datos informar al usuario y permitir corregirlos.</p> <p>Si los datos son correctos informar al usuario que se ha adicionado</p>	

Capítulo 2. Modelamiento del Negocio

	satisfactoriamente el tipo de retención.
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP-CH-Modelo Conceptual v1.7.doc.
Post-condiciones	Se ha adicionado un nuevo tipo de retención.
Post-requisito	No procede.

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 1.](#)

Especificación del requisito Modificar tipo de retención.

Conceptos tratados	Conceptos	Atributos
	Tipo de retención.	Código, Denominación, Prioridad, Condición de parada, Cuenta, Fecha inicio.
Precondiciones	Precondiciones	Pre-requisito
	Que se haya adicionado un tipo de retención.	Adicionar un tipo de retención.
Descripción	<p>Seleccionar la opción tipo de retención.</p> <p>Se escoge el tipo de retención a modificar, se deben deshabilitar los datos que no se pueden modificar como es el código.</p> <p>Se modifica él o los datos del tipo de retención.</p> <p>Se indica modificar el tipo de retención.</p> <p>El sistema valida los datos.</p> <p>En caso de error en la entrada de datos informar al usuario y permitir corregirlos.</p> <p>Si los datos son correctos informar al usuario que se ha modificado satisfactoriamente el tipo de retención.</p>	
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP-CH-Modelo Conceptual v1.7.doc.	

Capítulo 2. Modelamiento del Negocio

	Se debe garantizar que el código no se pueda modificar.
Post-condiciones	Se ha modificado el tipo de retención.
Post-requisito	No procede.

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 2.](#)

Especificación del requisito Eliminar tipo de retención.

Conceptos tratados	Conceptos	Atributos
	Tipo de retención.	Código, Denominación, Prioridad, Condición de parada, Cuenta, Fecha inicio.
Precondiciones	Precondiciones	Pre-requisito
	Que se haya adicionado un tipo de retención.	Adicionar tipo de retención.
Descripción	<p>Se selecciona la opción Tipo de retención.</p> <p>Se escoge el tipo de retención a eliminar.</p> <p>Se indica eliminar el tipo de retención.</p> <p>Se muestra un mensaje al usuario de confirmación. Ejemplo: ¿Desea eliminar el tipo de retención?</p> <p>Si se confirma se elimina el tipo de retención seleccionado.</p>	
Validaciones	No procede.	
Post-condiciones	Se ha eliminado el tipo de retención.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 3.](#)

2.9.2. Tipo de programa

En el requisito se gestionan los tipos de programas de retenciones que se tendrán en una entidad, que pueden ser mensual, quincenal o semanal. Posteriormente al registrar la retención se utiliza el programa para saber qué cantidad descontar en cada período de pago.

Especificación del requisito Adicionar tipo de programa

Conceptos tratados	Conceptos	Atributos
	Tipo de programa.	Denominación, Fecha de inicio.
Precondiciones	Precondiciones	Pre-requisito
	No procede.	No procede.
Descripción	<p>Se indica adicionar tipo de programa.</p> <p>El sistema permite adicionar los datos del programa: denominación y fecha inicio.</p> <p>Se presiona el botón aceptar.</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 programa.</p>	
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP-CH-Modelo Conceptual v1.7.doc.	
Post-condiciones	Se ha adicionado un nuevo tipo de programa.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 4.](#)

Especificación del requisito Modificar tipo de programa.

Conceptos tratados	Conceptos	Atributos
	Tipo de programa.	Denominación,

Capítulo 2. Modelamiento del Negocio

		Fecha de inicio.
Precondiciones	Precondiciones	Pre-requisito
	Que se haya adicionado un tipo de programa.	Adicionar un tipo de programa.
Descripción	<p>Se escoge el tipo de programa a modificar. El sistema permite modificar los datos.</p> <p>Se modifican los datos del tipo de programa.</p> <p>Se indica modificar el tipo de programa.</p> <p>El sistema valida los datos.</p> <p>En caso de error en la entrada de datos informar al usuario y permitir corregirlos.</p> <p>Si los datos son correctos informar al usuario que se ha modificado satisfactoriamente el tipo de programa.</p>	
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP-CH-Modelo Conceptual v1.7.doc.	
Post-condiciones	Se ha modificado el tipo de programa.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 5.](#)

Especificación del requisito Eliminar tipo de programa.

Conceptos tratados	Conceptos	Atributos
	Tipo de programa.	Denominación, Fecha de inicio.
Precondiciones	Precondiciones	Pre-requisito
	Que se haya adicionado un tipo de programa.	Adicionar tipo de programa.
Descripción	<p>Se escoge el tipo de programa a eliminar.</p> <p>Se indica eliminar el tipo de programa.</p> <p>Se muestra un mensaje al usuario de confirmación. Ejemplo: ¿Desea eliminar el tipo de programa?</p>	

	Si se confirma se elimina el tipo de programa seleccionado.
Validaciones	No procede.
Post-condiciones	Se ha eliminado el tipo de programa.
Post-requisito	No procede.

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 6.](#)

2.9.3. Gestionar tipo de recargo

Este requisito permite definir los distintos tipos de recargo que van a existir en la empresa.

Especificación del requisito Adicionar tipo de recargo.

Conceptos tratados	Conceptos	Atributos
	Tipo de recargo.	Denominación, Código, Descripción.
Precondiciones	Precondiciones	Pre-requisito
	No procede.	No procede.
Descripción	<p>Se selecciona la opción Gestionar tipo de recargo.</p> <p>El sistema permite adicionar los siguientes datos del recargo:</p> <p>Código: número que identifica al tipo de recargo.</p> <p>Denominación: nombre que va a recibir el recargo.</p> <p>Descripción: breve descripción del recargo.</p> <p>Se indica adicionar el tipo de recargo.</p> <p>El sistema valida los datos.</p> <p>En caso de error en la entrada de los 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 recargo.</p>	
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP-CH-Modelo Conceptual v1.7.doc.	
Post-condiciones	Se ha adicionado un nuevo tipo de recargo.	

Capítulo 2. Modelamiento del Negocio

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

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 7.](#)

Especificación del requisito **Modificar tipo de recargo.**

Conceptos tratados	Conceptos	Atributos
	Tipo de recargo.	Denominación, Código, Descripción.
Precondiciones	Precondiciones	Pre-requisito
	Que se haya adicionado un tipo de recargo.	Adicionar un tipo de recargo.
Descripción	<p>Seleccionar la opción Gestionar tipo de recargo.</p> <p>Se escoge el tipo de recargo a modificar, se debe deshabilitar los datos que no se pueden modificar como es el código.</p> <p>Se modifica el o los datos del tipo de recargo.</p> <p>Se indica modificar el tipo de recargo.</p> <p>El sistema valida los datos.</p> <p>En caso de error en la entrada de datos informar al usuario y permitir corregirlos.</p> <p>Si los datos son correctos informar al usuario que se ha modificado satisfactoriamente el tipo de recargo.</p>	
Validaciones	<p>El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP-CH-Modelo Conceptual v1.7.doc.</p> <p>Se debe garantizar que el código no se pueda modificar.</p>	
Post-condiciones	Se ha modificado el tipo de recargo.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 8.](#)

Especificación del requisito Eliminar tipo de recargo.

Conceptos tratados	Conceptos	Atributos
	Tipo de recargo.	Denominación, Código, Descripción.
Precondiciones	Precondiciones	Pre-requisito
	Que se haya adicionado un tipo de recargo.	Adicionar un tipo de recargo.
Descripción	Se selecciona la opción Gestionar tipo de recargo. Se escoge el tipo de recargo a eliminar. Se indica eliminar el tipo de recargo. Se muestra un mensaje al usuario de confirmación. Ejemplo: ¿Desea eliminar el tipo de recargo? Si se confirma se elimina el tipo de recargo seleccionado.	
Validaciones	No procede.	
Post-condiciones	Se ha eliminado el tipo de recargo.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 9.](#)

2.9.4. Apertura y cierre del submayor de vacaciones

Las cuentas de provisión de vacaciones son controladas por el módulo de nómina. Esto significa que después que se inicie el procesamiento no es posible hacer contabilizaciones a estas cuentas desde el módulo de contabilidad. Por esta razón antes de iniciar el procesamiento de la nómina es necesario hacer una carga inicial del submayor de vacaciones y garantizar su cuadro con los saldos de estas cuentas.

Especificación del requisito Apertura del submayor de vacaciones

Conceptos tratados	Conceptos	Atributos
	SM Vacaciones	No. carnet,

Capítulo 2. Modelamiento del Negocio

		<p>Nombre, Primer apellido, Segundo apellido, Operación, Cuenta, Los campos acumulado día período y acumulado importe período en la carga inicial tienen valor cero .Luego del procesamiento de las nóminas se actualizan los mismos.</p>
Precondiciones	Precondiciones	Pre-requisito
	<p>Tiene que existir en el maestro de trabajadores, al menos un trabajador.</p> <p>Tienen que estar parametrizada la cuenta provisión para vacaciones que el subsistema de nómina es propietario en la configuración general de este subsistema y su saldo en el acumulado del mayor de contabilidad general sea menor que cero.</p>	<p>Emitir Movimiento (Gestionar trabajador).</p>
Descripción	<p>Se selecciona la opción “Carga inicial” en la interfaz (Submayor de vacaciones). Se muestra el formulario de búsqueda del trabajador (Interfaz Buscar trabajador) donde se filtrará por los campos que desee, ya sea por: No. carnet, Nombre, Primer apellido, Segundo apellido. Se selecciona el trabajador deseado que se muestra como resultado de la búsqueda. Luego se acepta la búsqueda.</p>	

Capítulo 2. Modelamiento del Negocio

	<p>Se muestra el formulario Adicionar vacaciones con los siguientes campos cargados en la parte superior:</p> <p>No. carnet, Nombre, Primer apellido, Segundo apellido del trabajador que se obtuvo como resultado de la búsqueda anterior.</p> <p>En la parte inferior del formulario se mostrarán los siguientes campos para ser llenados:</p> <p>Acumulado en días, Acumulado en importe, Operación, Cuenta.</p> <p>Se acepta el formulario Adicionar vacaciones.</p> <p>Se muestra el trabajador en la interfaz (Submayor de vacaciones) con los datos que le fueron adicionados.</p>
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP- CH- Modelo Conceptual v1.7.doc.
Post-condiciones	Se ha realizado la apertura del submayor de vacaciones.
Post-requisito	No procede.

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 10.](#)

Especificación del requisito Cierre del submayor vacaciones

Conceptos tratados	Conceptos	Atributos
	SM Vacaciones.	No. carnet, Acumulado total en días, Acumulado total importe, Operación, Cuenta, Código nómina, Fecha, Acumulado día período,

Capítulo 2. Modelamiento del Negocio

		Acumulado importe período.
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir información adicionada al submayor de vacaciones.	Apertura del submayor de vacaciones.
Descripción	El especialista de nómina selecciona la opción Cierre del submayor de vacaciones. El especialista de nómina selecciona la opción "Cerrar".	
Validaciones	Se valida que el saldo de la cuenta de provisión para vacaciones sea igual al saldo de la cuenta en el mayor de Contabilidad. En caso de que exista descuadre mostrar el mensaje "No se puede realizar el Cierre, existe descuadre".	
Post-condiciones	Se ha realizado el cierre del submayor de vacaciones.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 11.](#)

2.9.5. Actualización del submayor de vacaciones

Cuando se confirma la nómina el sistema procede a actualizar el submayor de vacaciones, que no es más que hacer los cálculos correspondientes para que los datos de los (días e importe) acumulado durante un período coincidan con la nómina emitida y poder de esta forma saber el acumulado actual en (días e importe) de cada trabajador.

Especificación del Requisito Actualizar submayor de vacaciones

Conceptos tratados	Conceptos	Atributos
	SM Vacaciones.	No. carnet, Acumulado total en días, Acumulado total importe, Operación, Cuenta, Código nómina, Fecha,

Capítulo 2. Modelamiento del Negocio

		Acumulado día período, Acumulado importe período.
	Nómina	Código, Denominación, Fecha, Período, Estado, Tipo nómina, Id trabajador, Fondo tiempo, Horas trabajadas, Salario escala, Pagos adicionales, Otros pagos, Descuento, Salario bruto, Importe acumulado vacaciones, Retenciones, Salario neto, Elaborado por, Contabilizado por, Aprobado por, Contribución escala.
Precondiciones	Precondiciones	Pre-requisito
	Debe estar Confirmada la nómina.	Revisar nómina.
Descripción	Cuando es confirmada la nómina el sistema comienza a actualizar los Submayores de vacaciones, el proceso consiste en tomar de la nómina la cantidad acumulada por cada concepto (días e importe) y hacer la actualización en el Submayor de vacaciones de cada trabajador.	

Capítulo 2. Modelamiento del Negocio

	<p>El sistema valida los datos.</p> <p>Si los datos son correctos se realiza la actualización y se notifica al usuario.</p> <p>Si los datos no son correctos se notifica al usuario y se permite corregirlos.</p>
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP-CH-Modelo Conceptual v1.7.doc.
Post-condiciones	Se ha actualizado el submayor de vacaciones.
Post-requisito	No procede.

2.9.6. Detalles del submayor de vacaciones

En esta opción se mostrarán los detalles del submayor de vacaciones de un trabajador en específico, es decir se detallará cada una de las operaciones que han ido afectando el acumulado en días e importe del trabajador seleccionado.

Especificación del requisito Detalles del submayor de vacaciones

Conceptos	Conceptos	Atributos
tratados	SM vacaciones.	Fecha, Nómina, Acumulado período, Acumulado importe, Acumulado total en días, Acumulado total importe, Operación.
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir el submayor de vacaciones.	Apertura de submayor de vacaciones.
Descripción	<p>Se selecciona la opción Carga inicial y se busca por los campos seleccionados el o los trabajadores deseados.</p> <p>Se selecciona la opción detalles del trabajador seleccionado y se muestran desglosadas todas las operaciones que han afectado ese Submayor de vacaciones</p>	

	<p>y que han ido modificando la misma. Los campos que mostrará la interfaz son:</p> <p>Fecha: fecha de la operación que modificó el monto.</p> <p>Nómina: código de la nómina donde se origina la operación realizada en el submayor.</p> <p>Acumulado período: es el total de días que ha acumulado el trabajador durante los días de trabajo.</p> <p>Acumulado importe: es la cantidad de días que ha ido acumulando el trabajador durante los días de trabajo.</p> <p>Acumulado total en días: es el total de días que ha acumulado el trabajador durante el período.</p> <p>Acumulado total importe: es el total de dinero que ha acumulado el trabajador durante el período.</p> <p>Operación: es lo que se va a estar realizando en ese momento en el Submayor. Ejemplo de esto es: una actualización, una carga inicial, entre otras.</p>
Validaciones	Tiene que existir un acumulado en días e importe de al menos un trabajador.
Post-condiciones	No procede.
Post-requisito	No procede.

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 12.](#)

2.9.7. Conciliación con contabilidad del submayor de vacaciones

El subsistema de nómina controla la cuenta de provisión para vacaciones. Es necesario establecer un mecanismo de comparación del saldo de esta cuenta contable con el acumulado de la misma en el mayor de contabilidad general que se nombra conciliación de saldos.

Especificación del Requisito Conciliación con contabilidad del submayor de vacaciones

Conceptos tratados	Conceptos	Atributos
		Cuenta.
Precondiciones	Precondiciones	Pre-requisito

Capítulo 2. Modelamiento del Negocio

	Que se haya realizado la carga inicial del Submayor de vacaciones.	Apertura del submayor de vacaciones.
Descripción	<p>El sistema permite realizar la conciliación, se elige la cuenta deseada e inmediatamente se mostrará el saldo de esa cuenta de provisión para vacaciones y se mostrará el saldo del mayor de esa cuenta en contabilidad.</p> <p>El sistema compara los saldos de la cuenta provisión para vacaciones del módulo de nómina con el saldo del mayor de contabilidad, estos datos se muestran en la parte superior de la interfaz (Submayor de vacaciones).</p> <p>Si los datos son correctos se procede al procesamiento de la nómina, de lo contrario se verifican todas las cuentas realizadas a ese Submayor.</p>	
Validaciones	No procede.	
Post-condiciones	Se han conciliado los saldos.	
Post-requisito	No procede.	

2.9.8. Configuración del submayor de vacaciones

En esta opción se mostrará los datos necesarios que permitirán conocer el porcentaje en días e importe de las vacaciones de un trabajador y saber en qué cuenta se guardarán los mismos.

Especificación del requisito Configuración del submayor de vacaciones

Conceptos tratados	Conceptos	Atributos
	SM vacaciones.	% vacaciones, Cuenta.
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir en el maestro del Submayor de vacaciones al menos un trabajador que tenga días trabajados.	Emitir Movimiento (Gestionar trabajador).
Descripción	<p>Se selecciona la opción "Configuración del submayor de vacaciones".</p> <p>El sistema muestra un formulario para adicionar la información necesaria para</p>	

Capítulo 2. Modelamiento del Negocio

	<p>realizar la configuración del Submayor.</p> <p>En el formulario aparecerán los campos:</p> <p>Porcentaje de vacaciones: es el número que se va a multiplicar por el importe y los días trabajados de cada trabajador para sacar un porcentaje de ambos.</p> <p>Cuenta: es el lugar donde se va a guardar ese porcentaje que ya se ha calculado.</p> <p>Se presiona el botón aceptar.</p> <p>Los datos se guardan en la base de datos para ser usados posteriormente.</p>
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP- CH- Modelo Conceptual v1.7.doc.
Post-condiciones	Se ha realizado la Configuración del submayor de vacaciones.
Post-requisito	No procede.

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 13.](#)

2.9.9. Ajuste del submayor de vacaciones

En esta opción se permitirá realizar las modificaciones necesarias al Submayor, es decir si cuando se estaba realizando la apertura se cometieron errores, entonces en esta opción se podrán ajustar todos los datos que se deseen.

Especificación del requisito Ajuste del submayor de vacaciones

Conceptos tratados	Conceptos	Atributos
	SM vacaciones.	No. carnet, Nombre, Primer apellido, Segundo apellido, Ajuste días, Ajuste importe, Operación.
Precondiciones	Precondiciones	Pre-requisito

Capítulo 2. Modelamiento del Negocio

	Tiene que existir el submayor de vacaciones.	Apertura de submayor de vacaciones.
Descripción	<p>Se selecciona la opción Ajuste.</p> <p>Se muestra la interfaz (Ajuste de vacaciones) con los siguientes datos cargados:</p> <p>No. carnet, Nombre, Primer apellido, Segundo apellido.</p> <p>El resto de los campos que se muestran y hay que llenar para realizar el ajuste son:</p> <p>Ajuste días, Ajuste importe, Operación.</p> <p>Se presiona el botón aceptar y se ajustan los datos de Acumulado día período y Acumulado importe en la interfaz (Submayor de vacaciones).</p>	
Validaciones	Tiene que existir un acumulado total en días e importe de al menos un trabajador.	
Post-condiciones	No procede.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 14.](#)

2.9.10. Dar alta automática a un trabajador

Este requisito permite que una vez que el trabajador es insertado en la empresa, este pasa automáticamente al Submayor de vacaciones para las posteriores operaciones que se vayan a ir realizando con el paso del tiempo.

Especificación del Requisito Dar alta automática a un trabajador

Conceptos tratados	Conceptos	Atributos
	SM vacaciones.	Nombre,

Capítulo 2. Modelamiento del Negocio

		Primer apellido, Segundo apellido, Expediente, Cuenta, Acumulado día período, Acumulado importe período, Acumulado total en días, Acumulado total importe.
	Trabajador	Nombre, Primer apellido, Segundo apellido, CI (Expediente laboral).
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir el Submayor de vacaciones.	Apertura de submayor de vacaciones.
Descripción	<p>Cuando el trabajador es insertado en el sistema, para formar parte de la plantilla de los trabajadores de la empresa, el componente Trabajadores informa a los Submayores que se ha adicionado un nuevo trabajador.</p> <p>El Submayor verificará que el trabajador que fue adicionado aparezca en la interfaz (Submayor de vacaciones).</p> <p>Se validan los datos.</p> <p>Si hay error el sistema permite corregirlos, sino se muestra el trabajador que fue adicionado.</p>	
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP-CH-Modelo Conceptual v1.7.doc.	
Post-condiciones	Se ha dado de alta a un trabajador en el submayor de vacaciones.	
Post-requisito	No procede.	

2.9.11. Generar comprobante de operaciones dado una modificación

En esta opción se le enviará a nómina los datos necesarios para que realice el comprobante de operaciones.

Especificación del Requisito Generar comprobante de operaciones dado una modificación

Conceptos tratados	Conceptos	Atributos
	SM vacaciones.	Cuenta objeto, Cuenta contrapartida, Importe.
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir el submayor de vacaciones.	Apertura de submayor de vacaciones.
Descripción	<p>Cuando se realiza una modificación en el Submayor de vacaciones y después que se termina el procesamiento de la nómina, y se actualizan los Submayores, se envían los siguientes datos a nómina:</p> <p>Cuenta objeto, Cuenta contrapartida, Importe.</p> <p>Luego cuando nómina recibe estos datos se realiza en nómina el comprobante de operaciones.</p> <p>El sistema valida los datos.</p> <p>Si los datos son correctos se realiza el comprobante de operaciones y se notifica al usuario.</p> <p>Si los datos no son correctos se le notifica al usuario y se permite corregirlos.</p>	
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP-CH-Modelo Conceptual v1.7.doc.	
Post-condiciones	Se ha generado el comprobante de operaciones dado una modificación.	
Post-requisito	No procede.	

2.9.12. Gestionar submayor de retenciones

Las retenciones son adeudos del trabajador con el banco. Existen varios tipos de retenciones que se descuentan del salario devengado del trabajador según la prioridad que tengan las mismas. Algunos tipos de retenciones son: ley general de la vivienda, pensiones alimenticias y créditos a la población. Por cada tipo retención se abre un Submayor para el trabajador. Las deducciones se aplican en la nómina de pago de salario de los trabajadores, afectándose en nómina, salario, subsidio, vacaciones. También se deberá afectar en las nóminas adicionales que se emitan eventualmente.

Especificación del requisito Apertura del submayor de retenciones

Conceptos tratados	Conceptos	Atributos
	Trabajador.	Nombre, Primer apellido, Segundo apellido, CI (Expediente laboral).
	SM retenciones.	Se reflejarán en el submayor el No. carnet, Nombres, Primer apellido, Segundo apellido, Tipo de retención, Saldo de la deuda, Importe por plazo, Cantidad de plazo, Primera quincena, Segunda quincena, Fecha de retención, Sucursal bancaria. Los que se actualizarán con los descuentos que se efectúen por nómina.
Precondiciones	Precondiciones	Pre-requisito

Capítulo 2. Modelamiento del Negocio

	Tiene que existir en el maestro de trabajadores, al menos un trabajador, para poder adicionarle retenciones a algún trabajador.	Gestionar trabajadores.
	Tiene que estar definido el nomenclador tipo de retenciones a aplicar en la empresa.	Gestionar tipo de retención.
	Tiene que estar definido el nomenclador de programas de retenciones.	Programas de retenciones.
Descripción	<p>El especialista de nómina selecciona la opción Gestionar submayor de retenciones.</p> <p>Se muestra una interfaz con los siguientes campos:</p> <p>Expediente, Nombres, Primer apellido, Segundo apellido, Tipo de retención, Cuenta, Importe, Sucursal bancaria, Saldo de la deuda.</p> <p>La interfaz brinda la opción de adicionar, modificar y eliminar un registro de trabajador-retención. También brinda la opción de ver detalles de un registro de un trabajador en específico con el desglose de las operaciones que han afectado su Submayor, además muestra la opción de pago directo al banco que no es más que cuando el trabajador se dirige directamente al banco y paga por su cuenta sin esperar que en la empresa le descuenten, luego el banco manda un recibo de pago hacia la empresa y se le actualiza el Submayor a ese trabajador y el filtrar</p>	

que permite mostrar todas las retenciones que tiene un trabajador en específico o muestra todas las retenciones del mismo tipo con los datos correspondientes de los trabajadores que la tienen.

Se selecciona la opción Adicionar se muestra el formulario de búsqueda del trabajador (Interfaz Buscar trabajador) donde se filtrará por los campos que desee ya sea por:

- No de carnet,
- Nombre,
- Primer apellidos,
- Segundo apellido.

Se acepta la búsqueda realizada.

Se muestra el formulario Adicionar submayor de retenciones con los siguientes campos:

No. carnet, Nombres, Primer apellido, Segundo apellido del trabajador que se obtuvo como resultado de la búsqueda anterior en la parte superior del formulario.

Se selecciona el tipo de retención asociándole la misma al trabajador. Si el tipo de la retención es pensión alimenticia no saldrían los campos cantidad de plazos ni saldo total de la deuda.

Se adicionan los importes a pagar en el período. Son campos dinámicos según el programa de retenciones definido. Ejemplo. Si el período de pago es quincenal se define un programa 1ra quincena del mes y 2 quincena del mes por lo que aparecerán 2 campos para entrar los importes. Si es mensual aparecerá un solo campo para entrar el importe. Es decir la cantidad de campos que aparecerán para entrar el importe, dependerá del período de pago definido en la empresa.

Se muestra el importe del plazo mensual que sería la suma de los importes del programa que se utilice.

Se adiciona el saldo total de la deuda a descontar.

Se muestra la cantidad de plazos que es resultado de dividir el saldo de la deuda con el importe del plazo mensual.

Se adiciona la fecha en que se registró la retención.

Se acepta o se cancela la selección realizada. En caso de cancelar se cierra el

Capítulo 2. Modelamiento del Negocio

	<p>formulario y se regresa a la interfaz Gestionar submayor de retenciones.</p> <p>Si se acepta la selección se adiciona el trabajador con la retención asociada en la interfaz (Gestionar registro de retenciones).</p> <p>Al aceptar se validan los datos. Si hay errores el sistema permite corregirlos sino se adicionan los datos de la retención al trabajador.</p>
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP- CH- Modelo Conceptual v1.7.doc.
Post-condiciones	Se le ha realizado la apertura del Submayor de retenciones a un trabajador.
Post-requisito	No procede.

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 15.](#)

Especificación del requisito **Modificar registro retenciones**

Conceptos tratados	Conceptos	Atributos
	Trabajador.	Nombre, Primer apellido, Segundo apellido, CI (Expediente laboral).
	SM retenciones.	No. carnet, Nombres, Primer apellido, Segundo apellido, Tipo de retención, Saldo de la deuda, Importe por plazo, Cantidad de plazo, Primera quincena, Segunda quincena, Fecha de retención, Sucursal bancaria.
Precondiciones	Precondiciones	Pre-requisito

Capítulo 2. Modelamiento del Negocio

	Tiene que existir el Submayor de retenciones que se desea modificar.	Apertura del submayor de retenciones.
Descripción	<p>Dentro de la interfaz (Gestionar submayor de retenciones) se selecciona al trabajador que se le quiere modificar la retención.</p> <p>Se selecciona la opción modificar.</p> <p>Se muestra un formulario para modificar la retención asociada. Pero únicamente se podrán modificar los siguientes campos:</p> <p>Saldo de la deuda, Importe por plazo, Cantidad de plazos, Primera quincena, Segunda quincena.</p> <p>Se modifican los campos deseados y se acepta para actualizar.</p> <p>Se muestra un mensaje de confirmación " El submayor de retenciones fue modificado satisfactoriamente", este mensaje se muestra para saber que los datos fueron modificados satisfactoriamente.</p>	
Validaciones	No procede.	
Post-condiciones	Se ha modificado el Submayor de retenciones.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 16.](#)

Especificación del requisito Cierre del submayor de retenciones

Conceptos tratados	Conceptos	Atributos
	Trabajador.	Nombre, Primer apellido, Segundo apellido, CI (Expediente laboral).
Precondiciones	Precondiciones	Pre-requisito

Capítulo 2. Modelamiento del Negocio

	Tiene que existir el Submayor de retenciones que se desea eliminar.	Apertura de submayor de retenciones.
Descripción	<p>Se selecciona la opción Eliminar. Permitirá cerrarle el Submayor de retención a un trabajador.</p> <p>Se muestra un mensaje "Está seguro que desea eliminar la retención". Si se acepta se elimina el registro trabajador-retención y se muestra un mensaje de confirmación "El submayor de retenciones fue eliminado satisfactoriamente", para que el usuario sepa que se eliminó la retención sin ningún problema.</p>	
Validaciones	No procede.	
Post-condiciones	Se ha eliminado el Submayor de retenciones.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 17.](#)

2.9.13. Detalles del submayor de retenciones

En esta opción se mostrará el detalle de una retención específica del trabajador con todas las operaciones que han ido afectando la misma.

Especificación del requisito Detalles del submayor de retenciones

Conceptos	Conceptos	Atributos
tratados	SM retenciones.	Nombre, Primer apellido, Segundo apellido, No. carnet, Tipo de retención, Cuenta, Código de la nómina, Tipo de nómina, Fecha,

Capítulo 2. Modelamiento del Negocio

		Operación, Saldo acumulado, Debe retención, Haber retención, Saldo retención, Debe banco, Haber banco, Saldo banco.
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir el Submayor de retenciones.	Apertura de submayor de retenciones.
Descripción	<p>Se selecciona de la interfaz (Gestionar registro de retenciones) al trabajador que se desee.</p> <p>Se selecciona la opción ver detalles del trabajador seleccionado y se muestra un formulario con los siguientes campos cargados:</p> <p>Nombres, Primer apellidos, Segundo apellido, No. carnet, Tipo de retención, Cuenta.</p> <p>Se muestran desglosadas todas las operaciones que han afectado esa retención y que han ido modificando la misma. Los campos que se mostrarán son:</p> <p>Código de la nómina: identificación de la nómina donde se descontó el monto del período.</p> <p>Tipo de nómina: descripción del tipo de nómina.</p> <p>Fecha: fecha de la operación que modificó el monto de la retención.</p> <p>Operación: descripción de la operación realizada.</p> <p>Saldo acumulado: dinero que ha acumulado durante los días que lleva trabajando.</p> <p>Debe retención: lo que le desconté a cada trabajador.</p>	

Capítulo 2. Modelamiento del Negocio

	<p>Haber retención: lo que le tengo que descontar a cada trabajador.</p> <p>Saldo retención: es la resta del debe de la retención menos el haber de la retención.</p> <p>Debe banco: lo que le pagué al banco.</p> <p>Haber banco: lo que le tengo que pagar al banco.</p> <p>Saldo banco: Se le resta lo del debe al saldo que tiene el banco y esa diferencia es la que me va a quedar por pagarle al banco.</p>
Validaciones	Tiene que existir una retención asociada a un trabajador.
Post-condiciones	No procede.
Post-requisito	No procede.

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 18.](#)

2.9.14. Actualizar Submayor de Retención

Cuando se confirma la nómina el sistema procede a actualizar el Submayor de retenciones. Que no es más que hacer los cálculos correspondientes para que los datos que están en el Submayor coincidan con los de la nómina, y de esta forma quedará actualizado el Submayor del trabajador.

Especificación del requisito Actualizar submayor de retenciones

Conceptos tratados	Conceptos	Atributos
	SM retenciones.	Tipo de nómina, Saldo de la deuda, Importe por plazo, Cantidad de plazo, Sucursal, No. carnet, Cuenta, Código de la nómina, Fecha, Operación, Saldo acumulado,

Capítulo 2. Modelamiento del Negocio

		Debe retención, Haber retención, Saldo retención, Debe banco, Haber banco, Saldo banco.
	Nómina.	Código, Denominación, Fecha, Período, Estado, Tipo nómina, Id trabajador, Fondo tiempo, Horas trabajadas, Salario escala, Pagos adicionales, Otros pagos, Descuento, Salario bruto, Importe acumulado vacaciones, Retenciones, Salario neto, Elaborado por, Contabilizado por, Aprobado por, Contribución escala.
Precondiciones	Precondiciones	Pre-requisito
	Debe estar Confirmada la nómina.	Revisar nómina.

Capítulo 2. Modelamiento del Negocio

Descripción	<p>Cuando es confirmada la nómina el sistema comienza a actualizar los Submayores de retenciones, este proceso consiste en tomar de la nómina la cantidad retenida por cada concepto y hacer la actualización en cada Submayor de retenciones.</p> <p>El sistema valida los datos.</p> <p>Si los datos son correctos se realiza la actualización y se notifica al usuario.</p> <p>Si los datos no son correctos se notifica al usuario y se permite corregirlos.</p>
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP-CH-Modelo Conceptual v1.7.doc.
Post-condiciones	Se han actualizado los Submayores de retenciones correspondientes.
Post-requisito	No procede.

2.9.15. Conciliación con contabilidad del submayor de retenciones

En esta opción se permite la conciliación de los saldos con contabilidad, es decir deben de coincidir el saldo contable con el saldo del Submayor.

Especificación del requisito Conciliación con contabilidad del submayor de retenciones

Conceptos tratados	Conceptos	Atributos
	Cuenta.	Saldo contable.
Precondiciones	Precondiciones	Pre-requisito
	Que se haya realizado la carga inicial del Submayor de retenciones.	Apertura del submayor de retenciones.
Descripción	<p>Se muestra en la interfaz (Gestionar registro de retenciones) el saldo contable del Submayor que existe en contabilidad y el saldo del Submayor.</p> <p>Se verifican ambos saldos, porque deben de coincidir, si estos coinciden no se ha producido ningún error en las cuentas, en caso contrario se deberá de revisar todas las cuentas del Submayor.</p>	

Validaciones	No procede.
Post-condiciones	Se han conciliado los saldos.
Post-requisito	No procede.

2.9.16. Filtrar del submayor de retenciones

En esta opción se mostrará un listado de retenciones de un mismo tipo o un listado de un trabajador con todas las retenciones que tenga este.

Especificación del requisito Filtrar submayor de retenciones

Conceptos tratados	Conceptos SM retenciones.	Atributos Trabajador, Tipo de retención.
Precondiciones	Precondiciones Tiene que existir el Submayor de retenciones.	Pre-requisito Apertura de submayor de retenciones.
Descripción	<p>Se selecciona de la interfaz (Gestionar registro de retenciones) la opción de Filtrar.</p> <p>Se muestra una ventana con dos opciones que te permiten filtrar por: Trabajador o por Tipo de retención.</p> <p>Se selecciona la opción deseada y se presiona el botón aceptar.</p> <p>Se muestra en la interfaz (Gestionar registro de retenciones) todas las retenciones que tiene un trabajador asociadas o todas las retenciones de un mismo tipo. Esto depende de la opción que se seleccionó anteriormente.</p>	
Validaciones	Tiene que existir una retención asociada a un trabajador.	
Post-condiciones	No procede.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 19.](#)

2.9.17. Pago directo al banco del submayor de retenciones

En esta opción se le permitirá al trabajador pagarle directamente al banco, sin tener que esperar que se le descuente en la empresa por la retención, una vez que este paga el banco manda el recibo de pago a la empresa y el responsable en contabilidad actualiza los datos del Submayor.

Especificación del requisito Pago directo al banco del submayor de retenciones

Conceptos tratados	Conceptos	Atributos
	SM retenciones.	No. carnet, Nombre, Primer apellido, Segundo apellido, Tipo de retención, Importe.
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir el Submayor de retenciones.	Apertura de submayor de retenciones.
Descripción	<p>Se selecciona de la interfaz (Gestionar registro de retenciones) el trabajador que va a realizar el pago directo al banco.</p> <p>Se selecciona de la interfaz (Gestionar registro de retenciones) la opción de Pago.</p> <p>Se muestra una ventana con los siguientes datos cargados del trabajador que se seleccionó:</p> <p>No. carnet, Nombres, Primer apellido, Segundo apellido, Tipo de retención.</p> <p>Se llena el campo del importe que es el único campo que aparecerá sin datos.</p> <p>Se presiona el botón aceptar.</p> <p>Se actualiza el campo del saldo de la deuda en la interfaz (Gestionar registro de retenciones).</p>	
Validaciones	Tiene que existir una retención asociada a un trabajador.	

Post-condiciones	Se ha realizado el pago directo al banco de la retención.
Post-requisito	No procede.

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 20.](#)

2.9.18. Alerta de cumplido con el pago de las retenciones de los trabajadores

Este requisito permite mostrar todas las retenciones que han sido pagadas en el mes por los trabajadores, permitiendo que el trabajo del especialista sea más fácil, además de tener un mejor control de las retenciones pagadas.

Especificación del requisito Alerta de cumplido con el pago de la retención de los trabajadores

Conceptos tratados	Conceptos	Atributos
	SM retenciones.	Expediente, Nombre, Primer apellido Segundo apellido, Tipo de retención.
Precondiciones	Precondiciones	Pre-requisito
	Que se haya realizado la Carga inicial del Submayor de retenciones.	Apertura de submayor de retenciones.
Descripción	<p>Se selecciona de la interfaz (Gestionar registro de retenciones) la opción Alerta de cumplido.</p> <p>Se muestra una interfaz cargada con los siguientes datos de los trabajadores que ya han cumplido con el pago de la retención en ese mes.</p> <p>Los datos son:</p> <p>Expediente, Nombre, Primer apellido, Segundo apellido, Tipo de retención.</p>	
Validaciones	Tiene que existir una retención asociada a un trabajador.	

Post-condiciones	Se ha mostrado la alerta de cumplido con el pago de las retenciones de los trabajadores.
Post-requisito	No procede.

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 21.](#)

2.9.19. Gestionar recargo a la retención

Esta opción permite adicionar, modificar y eliminar un recargo a una retención determinada, para que el trabajador sepa y tenga registrado lo que tiene que pagar el próximo mes. Un recargo no es más que lo que tiene que pagar por atrasarse en el pago de una retención al banco.

Especificación del requisito Adicionar recargo a la retención

Conceptos tratados	Conceptos	Atributos
	SM retenciones.	Fecha, Cantidad de plazos, Id retención, Id recargo, Importe.
	Trabajador.	Nombre, Primer apellido, Segundo apellido, CI (Expediente laboral).
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir en el maestro de trabajadores, al menos un trabajador, para poder adicionarle recargos a algún trabajador.	Gestionar trabajadores.
Descripción	<p>Se selecciona de la interfaz (Gestionar registro de retenciones), la retención que se le va a aplicar el recargo.</p> <p>Luego de seleccionada la retención se selecciona la opción Gestionar recargo.</p> <p>Se muestra una interfaz donde se permite Adicionar, Modificar o Eliminar un recargo a esa retención.</p> <p>Se selecciona la opción Adicionar recargo.</p>	

Capítulo 2. Modelamiento del Negocio

	<p>Se muestra un formulario con los siguientes datos:</p> <p>Fecha cuando se aplica el recargo, Cantidad de plazos, Id retención, Id recargo, Importe.</p> <p>Se presiona el botón aceptar.</p> <p>Se validan los datos, si hay errores el sistema permite corregirlos sino se adicionan los datos del recargo a la retención.</p>
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP- CH- Modelo Conceptual v1.7.doc.
Post-condiciones	Se ha adicionado el recargo a la retención.
Post-requisito	No procede.

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 22.](#)

Especificación del requisito Modificar recargo a la retención

Conceptos tratados	Conceptos	Atributos
	SM retenciones.	Fecha, Cantidad de plazos, Id retención, Id recargo, Importe.
	Trabajador.	Nombre, Primer apellido, Segundo apellido, CI (Expediente laboral).
Precondiciones	Precondiciones	Pre-requisito

Capítulo 2. Modelamiento del Negocio

	Tiene que existir en el maestro de trabajadores, al menos un trabajador, para poder adicionarle recargos a algún trabajador.	Gestionar trabajadores.
Descripción	<p>Se selecciona de la interfaz (Gestionar registro de retenciones), la retención que se le va a aplicar el recargo.</p> <p>Luego de seleccionada la retención se selecciona la opción Gestionar recargo.</p> <p>Se muestra una interfaz donde se permite Adicionar, Modificar o Eliminar un recargo a esa retención.</p> <p>Se selecciona la opción Modificar recargo.</p> <p>Se muestra un formulario con los siguientes datos:</p> <p>Fecha cuando se aplica el recargo,</p> <p>Cantidad de plazos,</p> <p>Id retención,</p> <p>Id recargo,</p> <p>Importe.</p> <p>El Id retención y el Id recargo no se deben de modificar.</p> <p>Se presiona el botón aceptar.</p> <p>Se validan los datos, si hay errores el sistema permite corregirlos sino se modifican los datos del recargo a la retención.</p>	
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP- CH- Modelo Conceptual v1.7.doc.	
Post-condiciones	Se ha modificado el recargo a la retención.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 23.](#)

Especificación del requisito Eliminar recargo a la retención

Conceptos	Conceptos	Atributos
-----------	-----------	-----------

Capítulo 2. Modelamiento del Negocio

tratados	SM retenciones.	Fecha, Cantidad de plazos, Id retención, Id recargo, Importe.
	Trabajador.	Nombre, Primer apellido, Segundo apellido, CI (Expediente laboral).
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir en el maestro de trabajadores, al menos un trabajador, para poder adicionarle recargos a algún trabajador.	Gestionar trabajadores.
Descripción	<p>Se selecciona de la interfaz (Gestionar registro de retenciones), la retención que se le va a aplicar el recargo.</p> <p>Luego de seleccionada la retención se selecciona la opción Gestionar recargo.</p> <p>Se muestra una interfaz donde se permite Adicionar, Modificar o Eliminar un recargo a esa retención.</p> <p>Se selecciona la opción Eliminar recargo.</p> <p>Se presiona el botón aceptar.</p> <p>Se muestra un mensaje de confirmación "Se ha eliminado el recargo de la retención".</p>	
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP- CH- Modelo Conceptual v1.7.doc.	
Post-condiciones	Se ha eliminado el recargo de la retención.	
Post-requisito	No procede.	

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 24.](#)

2.9.20. Descuento adelantado de las retenciones

Este requisito permite descontarle a un trabajador una retención antes que este vaya a salir de vacaciones u otro problema que se le presente para el próximo mes y no se vaya a encontrar en la empresa, permitiendo que la retención sea pagada en tiempo y no ocurra ningún atraso en el pago de la misma.

Especificación del requisito Descuento adelantado de las retenciones

Conceptos tratados	Conceptos	Atributos
	SM retenciones.	Tipo de nómina, Saldo de la deuda, Importe por plazo, Cantidad de plazo, Sucursal, No. carnet, Cuenta, Código de la nómina, Fecha, Operación, Saldo acumulado, Debe retención, Haber retención, Saldo retención, Debe banco, Haber banco, Saldo banco.
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir el Submayor de retenciones.	Apertura de submayor de retenciones.
Descripción	Cuando es confirmada la nómina, el sistema comienza a actualizar los datos y en el Submayor de retenciones aparecerán todas aquellas retenciones que van a ser descontadas de forma adelantadas por los especialistas de nómina.	

	<p>El sistema valida los datos.</p> <p>Si la actualización se hizo de forma correcta, se le notifica al usuario sobre el descuento y se procede a descontar del salario del trabajador.</p> <p>Si los datos no son correctos se notifica al usuario y se permite corregirlos.</p>
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP- CH- Modelo Conceptual v1.7.doc.
Post-condiciones	Se ha realizado el descuento adelantado de las retenciones.
Post-requisito	No procede.

2.9.21. Mostrar fichero

Este requisito permite mostrar a todos los trabajadores existentes en la empresa con todo el salario que estos han ido devengando en los meses que llevan trabajando.

Especificación del requisito Mostrar fichero

Conceptos tratados	Conceptos	Atributos
	Trabajador.	Nombre, Primer apellido, Segundo apellido, CI (Expediente laboral).
Precondiciones	Precondiciones	Pre-requisito
	Tienen que existir trabajadores en la empresa.	Emitir movimiento de alta.
Descripción	<p>Se selecciona en la interfaz (Fichero histórico devengado) la opción Mostrar fichero.</p> <p>Se muestra una interfaz que te permite buscar un trabajador por distintos criterios de búsqueda, se acepta y se escoge el trabajador que se desee.</p> <p>Se muestra una interfaz con los siguientes datos cargados del trabajador seleccionado de la búsqueda:</p> <p>Expediente laboral,</p>	

	Nombre, Primer apellido, Segundo apellido, Importe por mes, que no es más que lo que ha ido acumulando cada trabajador por los días que ha laborado en cada uno de los meses.
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP-CH-Modelo Conceptual v1.7.doc.
Post-condiciones	No procede.
Post-requisito	No procede.

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 25.](#)

2.9.22. Modificar fichero

Este requisito permite realizar las distintas modificaciones del importe en el fichero de los trabajadores. Por haberse cometido algún error a la hora de entrar los datos en los Submayores de cada trabajador.

Especificación del requisito Modificar fichero

Conceptos tratados	Conceptos	Atributos
	Trabajador.	Nombre, Primer apellido, Segundo apellido, CI (Expediente laboral).
Precondiciones	Precondiciones	Pre-requisito
	Tienen que existir trabajadores en la empresa.	Emitir movimiento de alta.
Descripción	Se busca al trabajador. Se selecciona el trabajador que va a ser modificado. Se selecciona en la interfaz (Fichero histórico devengado) la opción Modificar fichero. Se muestra un formulario con los siguientes datos cargados y deshabilitados: Expediente laboral,	

Capítulo 2. Modelamiento del Negocio

	<p>Nombre, Primer apellido, Segundo apellido, El Importe por mes, es el único campo que permitirá ser modificado. Se validan los datos. Se muestra un mensaje de confirmación” El fichero fue modificado satisfactoriamente”, este mensaje se muestra para saber que los datos fueron modificados satisfactoriamente.</p>
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP-CH-Modelo Conceptual v1.7.doc.
Post-condiciones	Se ha modificado el fichero devengado de los trabajadores.
Post-requisito	No procede.

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 26.](#)

2.9.23. Alta automática al fichero

Esta opción permite que una vez que el trabajador sea insertado en la empresa, este aparezca automáticamente en el Fichero histórico devengado.

Especificación del requisito Alta automática al fichero

Conceptos tratados	Conceptos	Atributos
	Trabajador.	Nombre, Primer apellido, Segundo apellido, CI (Expediente laboral).
Precondiciones	Precondiciones	Pre-requisito
	Tiene que existir un trabajador en la empresa.	Emitir movimiento de acta.
Descripción	Cuando el trabajador es insertado en el sistema, para formar parte de la plantilla de los trabajadores de la empresa, el componente Trabajadores informa al fichero histórico devengado.	

Capítulo 2. Modelamiento del Negocio

	<p>En la interfaz (Fichero histórico devengado) se muestra el trabajador que fue adicionado junto con los que ya existían en la empresa.</p> <p>Se validan los datos.</p> <p>Si hay error el sistema permite corregirlos, sino se muestra el trabajador que fue adicionado.</p>
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP-CH-Modelo Conceptual v1.7.doc.
Post-condiciones	Se ha dado de alta automática a un trabajador en el fichero histórico devengado.
Post-requisito	No procede.

2.9.24. Resumen devengado del trabajador en meses anteriores

Este requisito permite mostrar una cierta cantidad del salario devengado de un trabajador, dependiendo de la cantidad de meses que se señalen por el especialista.

Especificación del requisito Resumen devengado del trabajador en meses anteriores

Conceptos tratados	Conceptos	Atributos
	Trabajador.	Nombre, Primer apellido, Segundo apellido, CI (Expediente laboral).
Precondiciones	Precondiciones	Pre-requisito
	Tienen que existir trabajadores en la empresa.	Emitir movimiento de acta.
Descripción	<p>Se selecciona de la interfaz (Fichero histórico devengado), al trabajador que se quiere mostrar el resumen devengado.</p> <p>Se presiona la opción filtrar, donde se filtrará por fecha inicio y fecha fin.</p> <p>Se muestra una interfaz como resultado del filtrar, mostrándose el total del salario de ese trabajador en el tiempo señalado, se mostrará además el Nombre,</p>	

Capítulo 2. Modelamiento del Negocio

	Primer apellido, Segundo apellido, CI. Se validan los datos.
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual: ERP-CH-Modelo Conceptual v1.7.doc.
Post-condiciones	Se ha mostrado el resumen histórico devengado de él trabajador.
Post-requisito	No procede.

Prototipo de interfaz de usuario del requisito descrito. [Ver anexo 27.](#)

Conclusiones parciales del capítulo

En el presente capítulo se realizó la descripción del negocio, se identificaron y describieron los procesos, se construyó el Modelo conceptual y se detallaron las entidades que se relacionan, se especificaron los requisitos funcionales con el objetivo de satisfacer las necesidades de los clientes, se hizo referencia a los prototipos de interfaz de usuario para dar una primera versión del sistema que se quiere realizar. Se especificaron las técnicas para el levantamiento de requisitos y los patrones utilizados para la agrupación de los mismos.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

En el presente capítulo se realizarán los distintos diagramas del diseño entre los que se encuentran el diagrama de clases del diseño que mostrará la estructura del sistema y el diagrama de componentes que visualizará las distintas dependencias que existirán entre los distintos componentes, ambos permitirán tener un mejor entendimiento del sistema que se quiere construir a los futuros programadores. Se especificarán además los patrones de asignación de responsabilidades y los patrones del diseño utilizados.

3.1. Modelo de Diseño

El modelo de diseño es un modelo de objetos que describe la realización de los requisitos y sirve como una abstracción del modelo de implementación y de código fuente, pues es utilizado como una entrada esencial en las actividades de implementación y prueba (5). Es usado además para concebir en adición a la documentación del diseño del sistema software. De manera general es un abarcador y compuesto artefacto que encapsula todas las clases de diseño, subsistemas, paquetes, colaboraciones, y las relaciones entre ellos.

3.2. Patrones arquitectónicos

Los patrones de arquitectura especifican un conjunto predefinido de sistemas con sus responsabilidades y una serie de recomendaciones para organizar los distintos componentes.

3.2.1. Patrón Modelo-Vista-Controlador (MVC)

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. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el controlador es el Sistema de Gestión de Base de Datos y el modelo es el modelo de datos.

El Modelo es el responsable de acceder a la capa de almacenamiento de datos, permite definir las reglas de negocio (la funcionalidad del sistema), llevar un registro de las Vistas y Controladores del sistema, notificar a las Vistas los cambios que en los datos pueda producir un agente.

El Controlador es responsable de recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.). Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al Modelo o a las Vistas. Una de estas peticiones a las Vistas puede ser una llamada a un método.

Las Vistas son responsables de recibir datos del Modelo y mostrarlo al usuario, tienen un registro de su Controlador asociado (normalmente, porque además lo instancia). Pueden dar el servicio de "Actualización", para que sea invocado por el Controlador o por el Modelo (cuando es un Modelo activo que informa de los cambios en los datos producidos por otros agentes). (24)

Por las características expuestas anteriormente y por permitir un mejor acoplamiento entre la capa de presentación, la lógica del negocio y la de acceso a datos, es que fue utilizado este patrón para conformar los diagramas de clases del diseño que se realizaron a lo largo de este trabajo.

3.3. Patrones de asignación de responsabilidades (Patrones GRASP)

“Los patrones GRASP (Patrones Generales de Software para Asignar Responsabilidades) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones” (21). La puesta en práctica de estos patrones que se muestran a continuación se evidencia en el próximo capítulo a través de las métricas orientadas a clases.

Experto: Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

Creador: Asignar responsabilidades para la creación de objetos, es decir asignarle a la clase B la responsabilidad de crear una instancia de clase A

Bajo acoplamiento: Es la meta principal que es preciso tener presente durante el diseño. Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño. Estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento.

Alta cohesión: Una clase tiene responsabilidades moderadas en un área funcional y colabora con las otras para llevar a cabo las tareas. Es un principio que se debe tener presente en todas las decisiones del

diseño, es la meta principal que ha de buscarse en todo momento. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones del diseño. (21)

3.4. Patrones de diseño

“Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan.” (25)

3.4.1. Patrones GOF

Patrones estructurales

Fachada: Provee de una interfaz unificada simple para acceder a una interfaz o grupo de interfaces de un subsistema. (21)

Este es utilizado en varios de los diagramas de clases realizados en el diseño. Un ejemplo de su uso es en los servicios, donde la relación existente entre las clases controladoras y los servicios permite acceder a métodos que no están implementados en el componente Submayores y que se encuentran, tanto en otros componentes pertenecientes al subsistema Capital Humano, como en otros subsistemas externos.

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. (21)

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.

3.5. Diagrama de clases

“Una clase es una descripción de un conjunto de objetos que comparten las mismas responsabilidades, relaciones, operaciones, atributos y significado semántico. Una clase de diseño representa una abstracción de una o más clases en la implementación de un sistema. Exactamente la correspondencia depende del lenguaje de implementación, al igual que su tamaño y sus objetos.” (21) Las clases definen objetos, los cuales a su vez implementan los requerimientos; si las mismas son buenas o no, dependen profundamente del entorno de implementación. De manera general las clases deben mapearse en un fenómeno en particular en el lenguaje de implementación y deben estar estructuradas de manera que representen los resultados en un buen código. A continuación se muestran los 9 diagramas de clases realizados en el diseño.

3.5.1. Diagramas de clases del diseño

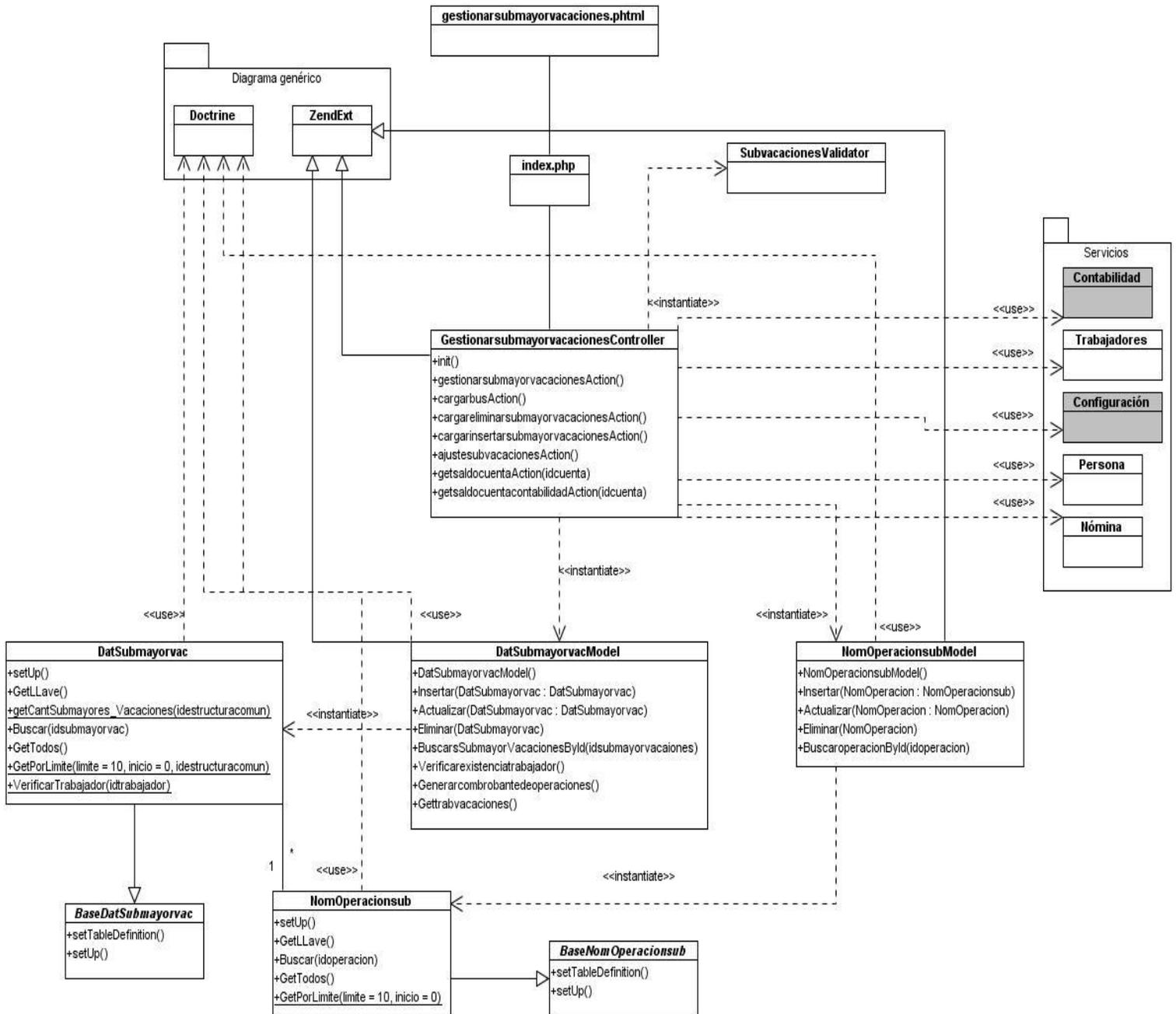


Fig.1. Diagrama de clases del Gestionar submayor de vacaciones. [Ver descripción.](#)

Capítulo 3. Análisis y Diseño del sistema

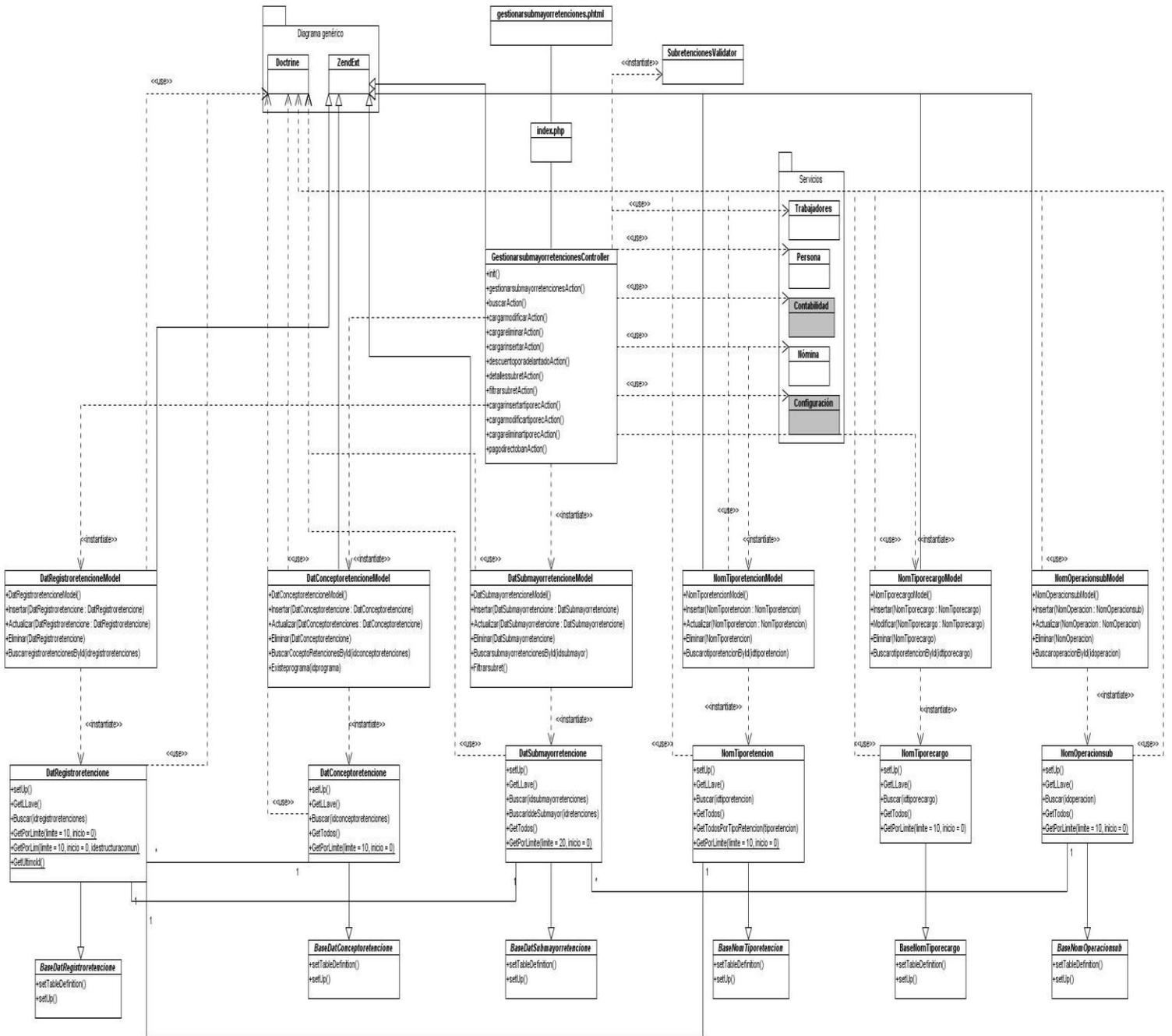


Fig.2. Diagrama de clases del Gestionar submayor de retenciones. [Ver descripción.](#)

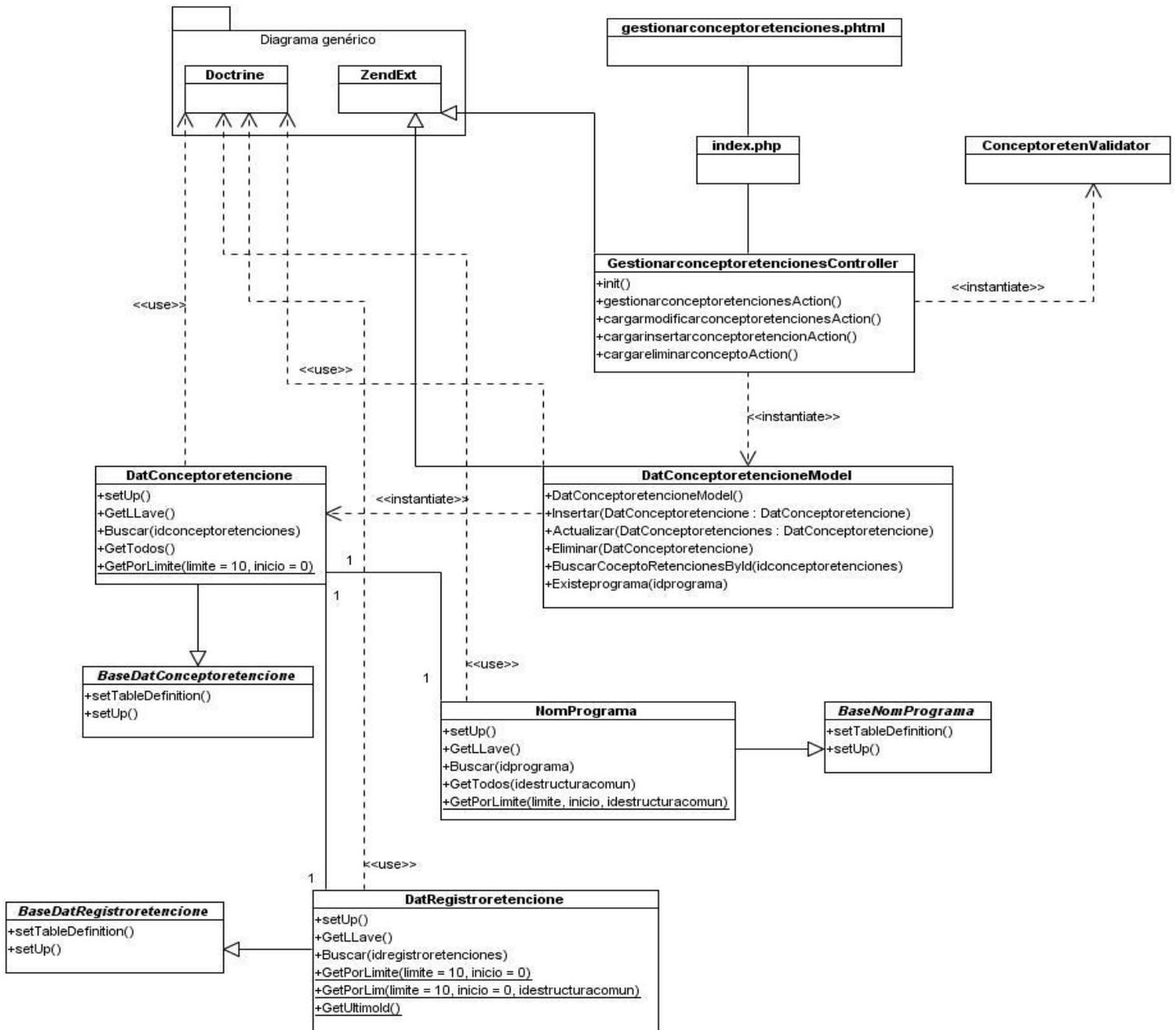


Fig.3. Diagrama de clases del Gestionar concepto de retención. [Ver descripción.](#)

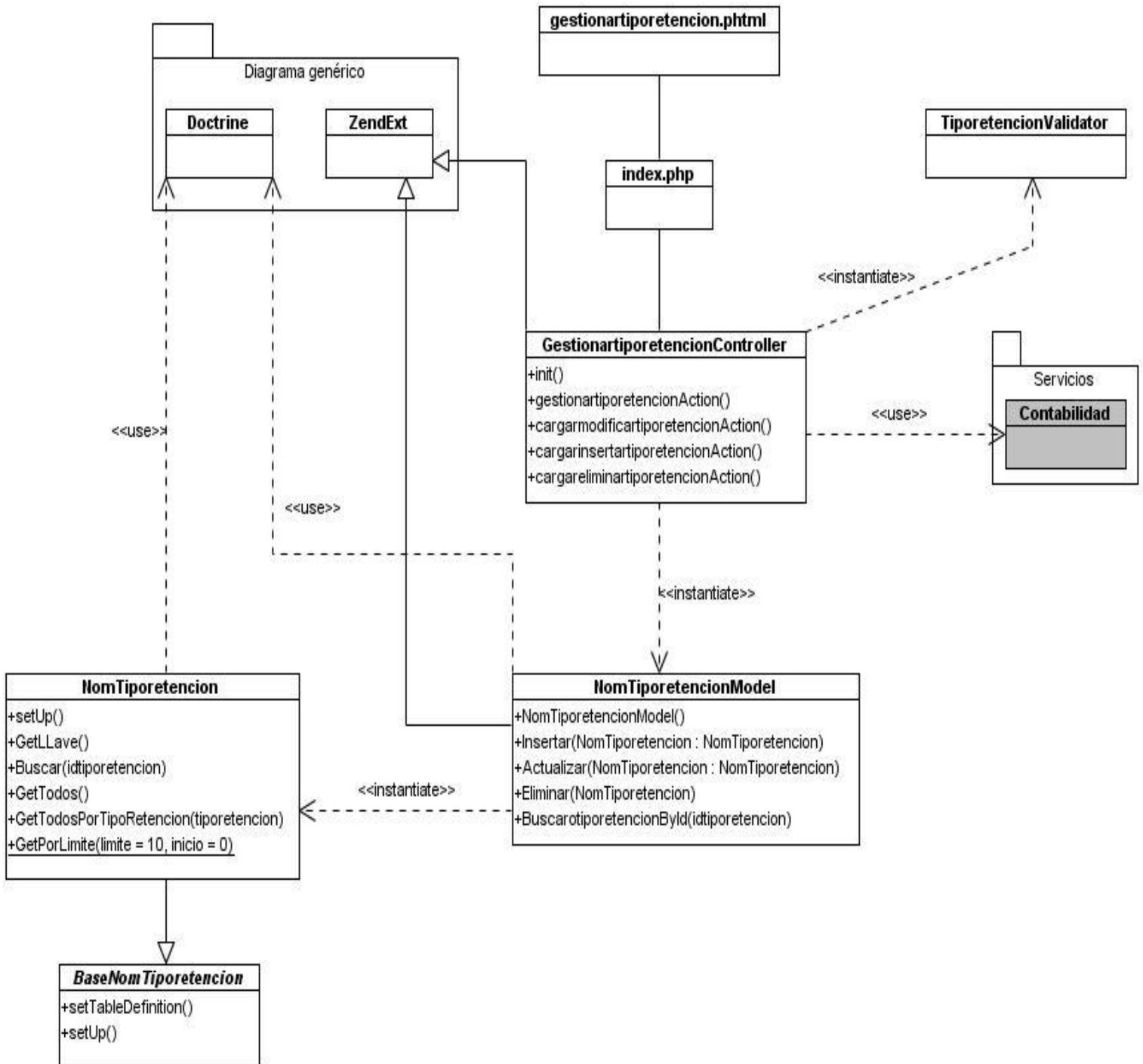


Fig.5. Diagrama de clases del Gestionar tipo de retenciones. [Ver descripción.](#)

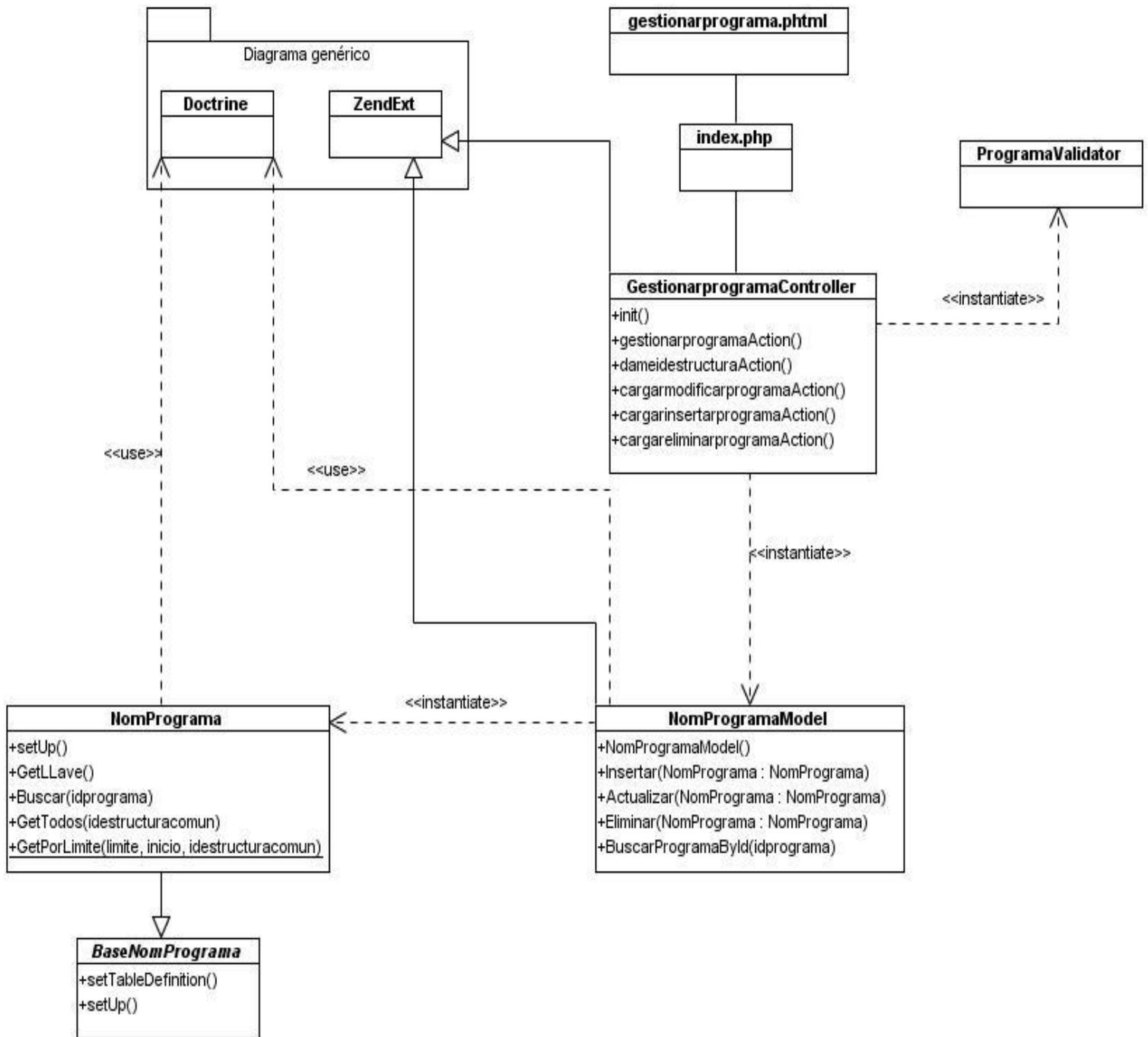


Fig.6. Diagrama de clases del Gestionar nomenclador de programas. [Ver descripción.](#)

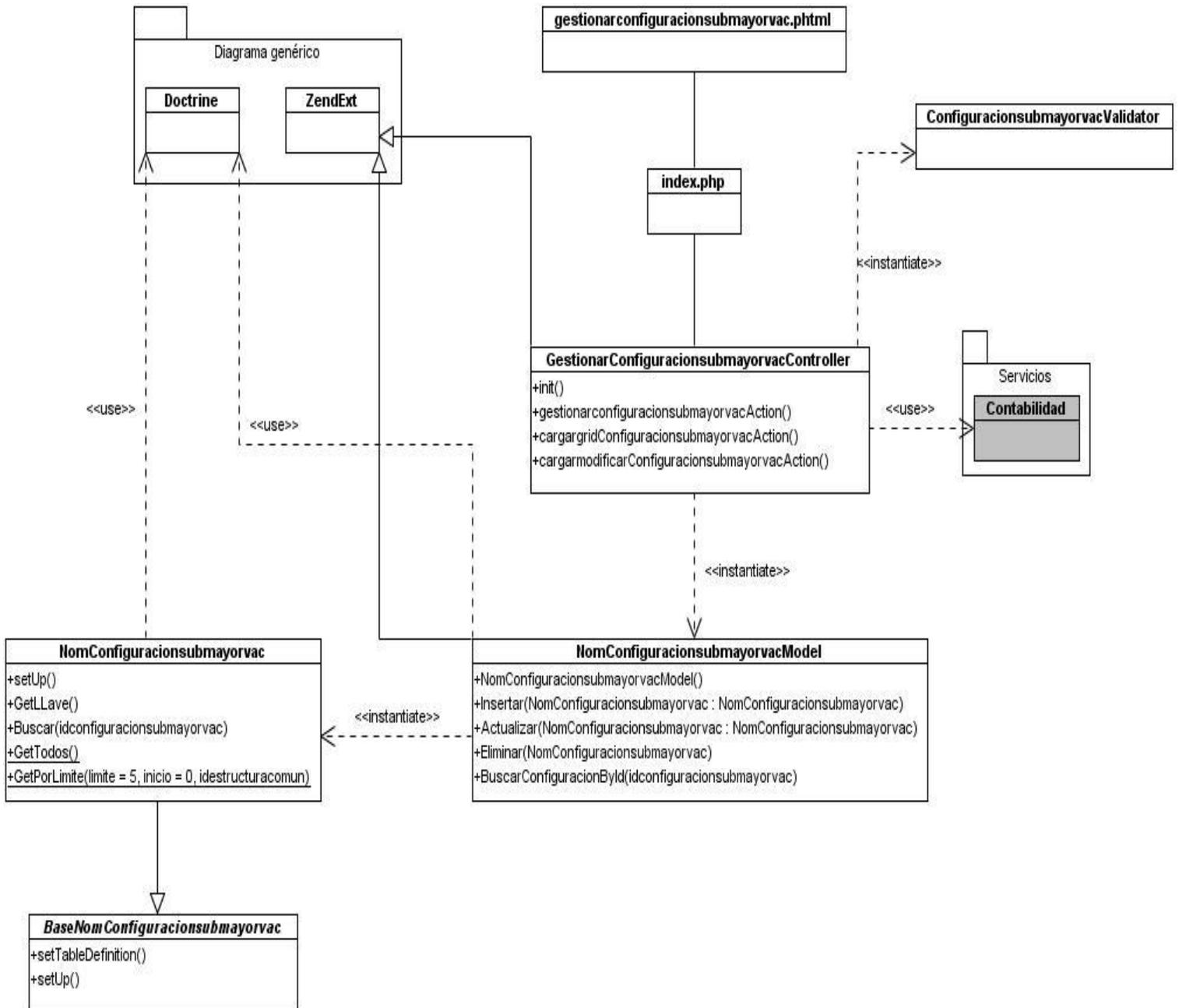


Fig.7. Diagrama de clases de la Configuración del submayor de vacaciones. [Ver descripción.](#)

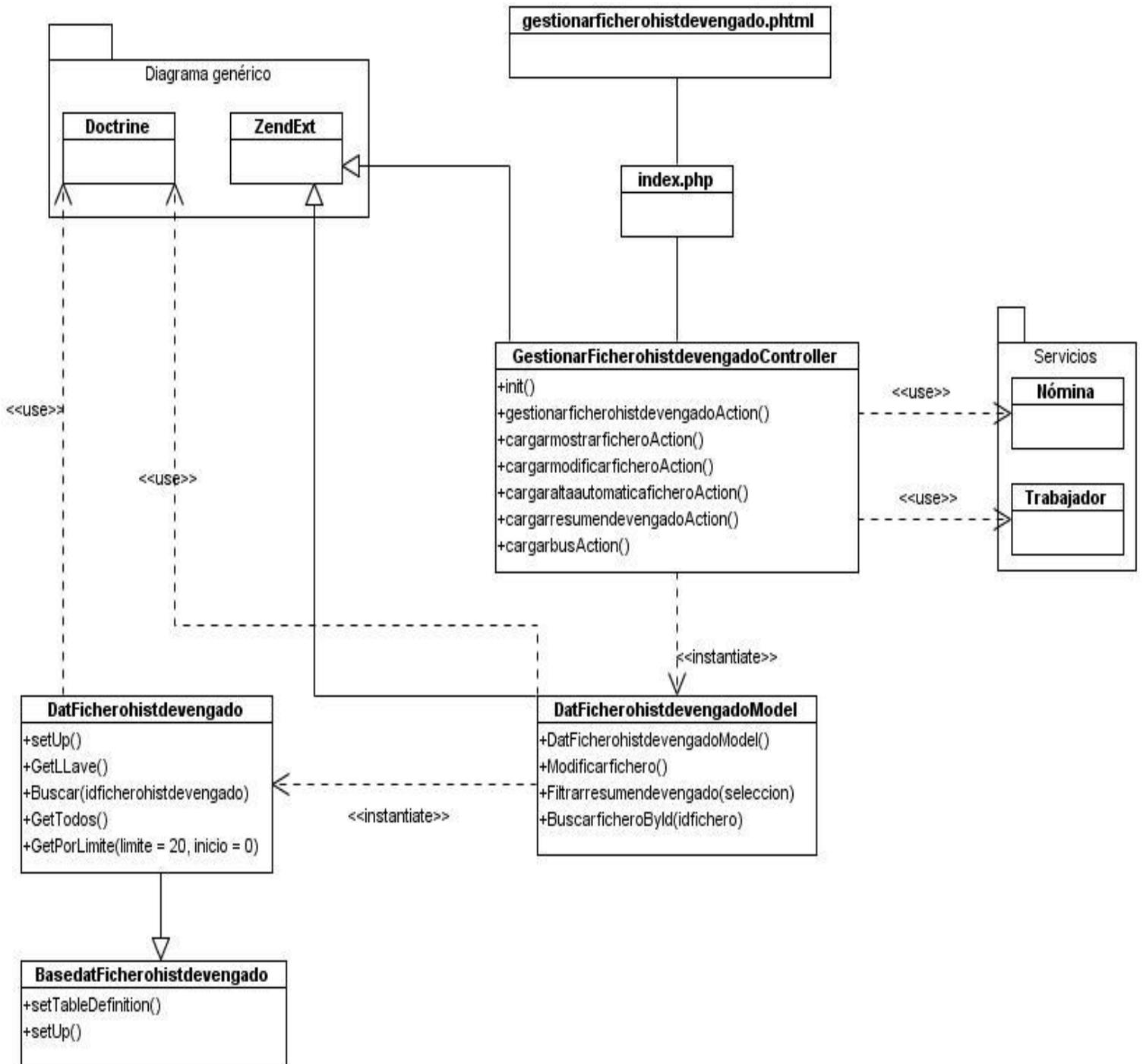


Fig.8. Diagrama de clases del Fichero histórico devengado. [Ver descripción.](#)

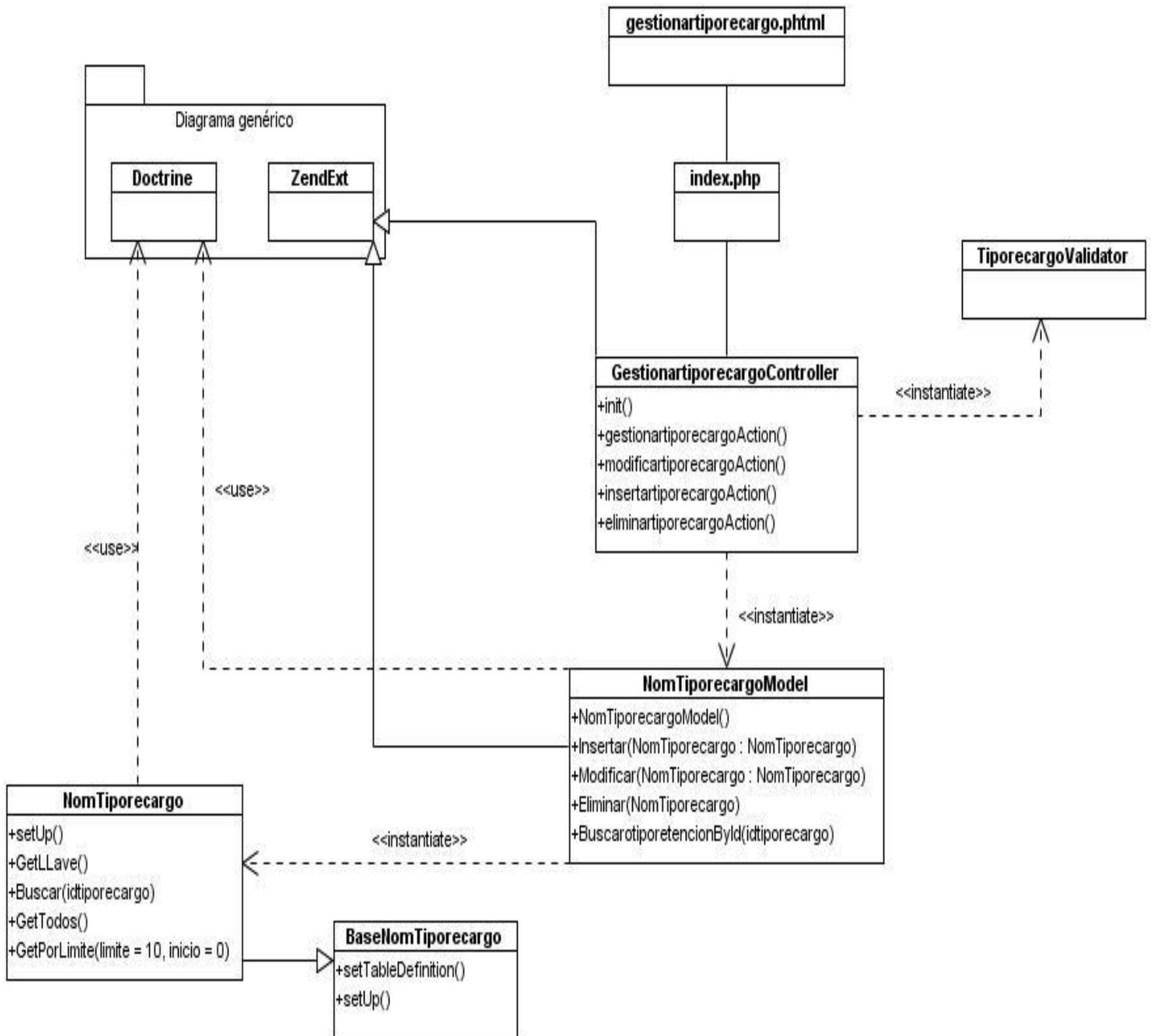


Fig.9. Diagrama de clases del Gestionar tipo de recargo. [Ver descripción.](#)

3.6. Diagrama de componente

Un diagrama de componente es una parte modular de un sistema, desplegable y reemplazable que encapsula implementación y un conjunto de interfaces y proporciona la realización de los mismos. Un componente típicamente contiene clases y puede ser implementado por uno o más artefactos (archivos ejecutables, binarios, tablas, archivos, documentos y páginas Web.). A continuación se muestra el diagrama de componentes de los Submayores del subsistema Capital Humano del sistema Cedrux.

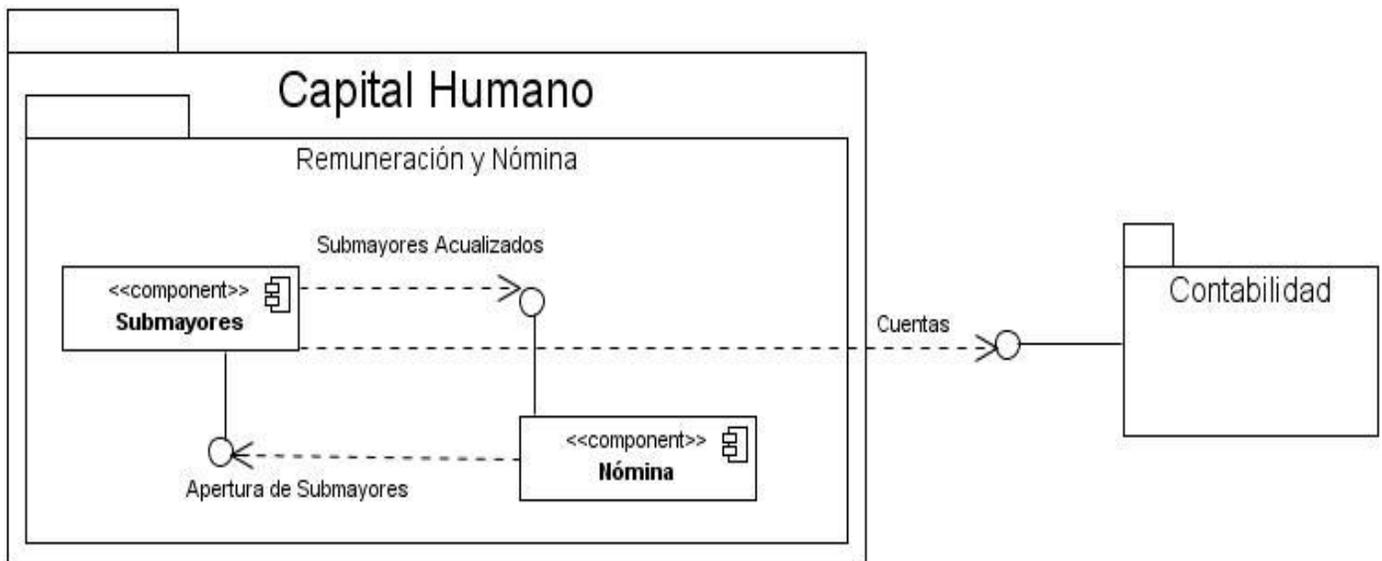


Fig.11. Diagrama de componentes de Submayores.

El diagrama de componentes que se muestra en la figura 11, está compuesto por el subsistema Capital Humano integrado por los procesos de Remuneración y nómina, donde se encontrarán los componentes de los Submayores y de nómina. Ambos componentes se relacionan como se explica a continuación:

	Nómina	Submayores	Subsistema Contabilidad
Entrada	Apertura de los submayores	Submayores actualizados	
Salida	Submayores actualizados	Apertura de los submayores	Cuentas

Conclusiones parciales del capítulo

En el presente capítulo se construyeron los diferentes diagramas del diseño, entre los que se encuentran el diagrama de clases que muestra la forma en que van a estar compuesto estructuralmente el sistema y se representó gráficamente el diagrama de componentes donde se visualiza claramente las entradas y las salidas de cada componente y se explicaron y aplicaron además los diferentes patrones del diseño, los de asignación de responsabilidades y el patrón de arquitectura modelo vista controlador utilizados para la confección de los diagramas de clases.

CAPÍTULO 4: VALIDACIÓN DE REQUISITOS

En este capítulo se mostrarán las técnicas de validación que fueron aplicadas para demostrar que las descripciones de los requisitos definen lo que el cliente necesita. Se aplicarán algunas métricas a los requisitos para saber si la especificación de los mismos puede ser aceptada o rechazada, así como la aplicación de las métricas para medir la efectividad de los diferentes patrones aplicados en la construcción de las clases del diseño.

4.1. Técnicas para la validación de los requisitos

La validación de requisitos tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea. Es necesario asegurar que el análisis realizado y los resultados obtenidos de la etapa de definición de los requisitos son correctos.

Pocas son las propuestas existentes que ofrecen técnicas para la realización de la validación y muchas de ellas consisten en revisar los modelos obtenidos en la definición de requisitos con el usuario para detectar errores o inconsistencias.

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

Entre las técnicas de validación de requerimientos se encuentran:

- Listas de Chequeo
- Construcción de prototipos
- Generación de Casos de Pruebas

Para la validación de los requisitos del presente trabajo se utilizaron dos de estas técnicas:

La construcción de prototipos: porque estos son una versión inicial de un sistema de software que se utilizan para demostrar los conceptos, probar las opciones de diseño y entender mejor el problema y su solución, permiten revelar errores u omisiones en los requerimientos propuestos, favorecen la comunicación entre clientes y desarrolladores, y dan una primera visión del producto. [Ver anexos](#)

Los casos de prueba: porque estos permiten definir las entradas, las salidas del software y acciones del usuario para que se pueda completar lo que expresa el requisito. Además como los requisitos deben ser posibles de probar, se le debe de asociar un caso de prueba por cada requerimiento. A continuación se muestran dos casos de pruebas correspondientes a los requisitos más importantes. Para conocer el resto de los caso de prueba dirigirse a los [Anexos](#).

4.1.1. Construcción de prototipos de interfaz usuario

Un prototipo de interfaz usuario es una representación limitada del diseño de un producto que permite a las partes responsables de su creación experimentar, probarlo en situaciones reales y explorar su uso. Son útiles para comunicar, discutir y definir ideas entre los diseñadores y las partes responsables. Los prototipos apoyan el trabajo evaluando productos, clarificando requisitos de usuario y definiendo alternativas.

4.1.2. Diseño de casos de prueba

En la ingeniería del software, los casos de prueba son un conjunto de condiciones o variables bajo las cuales el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio.

Se pueden realizar muchos casos de prueba para determinar que un requisito es completamente satisfactorio. Con el propósito de comprobar que todos los requisitos de una aplicación son revisados, debe haber al menos un caso de prueba para cada requisito a menos que un requisito tenga requisitos secundarios. En ese caso, cada requisito secundario deberá tener por lo menos un caso de prueba. Si la aplicación es creada sin requisitos formales, entonces los casos de prueba se escriben basados en la operación normal de programas de una clase similar.

Lo que caracteriza un escrito formal de caso de prueba es que hay una entrada conocida y una salida esperada, los cuales son formulados antes de que se ejecute la prueba. La entrada conocida debe probar una precondición y la salida esperada debe probar una postcondición.

Bajo circunstancias especiales, podría haber la necesidad de ejecutar la prueba, producir resultados, y luego un equipo de expertos evaluaría si los resultados se pueden considerar como "Correctos". Esto sucede a menudo en la determinación del número del rendimiento de productos nuevos. La primera prueba se toma como línea base para los subsecuentes ciclos de pruebas/lanzamiento del producto.

Capítulo 4. Validación de los Requisitos

Los casos de prueba escritos, incluyen una descripción de la funcionalidad que se probará, la cuál es tomada ya sea de los requisitos o de los Casos de Uso, y la preparación requerida para asegurarse de que la prueba pueda ser dirigida.

Las variaciones de los casos de prueba son comúnmente utilizadas en pruebas de aceptación. La prueba de aceptación es realizada por un grupo de usuarios finales o los clientes del sistema, para asegurarse que el sistema desarrollado cumple sus requisitos. La prueba de aceptación de usuario se distingue generalmente por la incorporación de un trayecto feliz o casos de prueba positivos. (26)

4.1.2.a. Caso de prueba del Gestionar Submayor de vacaciones

Apertura del submayor de vacaciones

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 Submayores.
- Se debe seleccionar el menú Submayor de vacaciones.

Requisitos a probar

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

Capítulo 4. Validación de los Requisitos

1. Carga inicial del submayor de vacaciones.	Se realiza la apertura del submayor de vacaciones a un trabajador que ingresa a la empresa.	EP 1.1: Adicionar un submayor de vacaciones introduciendo los datos correctamente.	<ul style="list-style-type: none">- Se debe seleccionar la opción Carga inicial.- Se debe buscar el trabajador al que se le desea realizar la operación, introduciendo los datos requeridos.- Se debe presionar el botón Buscar.- Se debe seleccionar el trabajador encontrado.- Se debe presionar el botón Aceptar.- Se deben introducir en el formulario los datos correctamente.- Se presiona el botón Aceptar.
--	---	--	--

Capítulo 4. Validación de los Requisitos

		<p>EP 1.2: Adicionar un submayor de vacaciones dejando campos requeridos en blanco.</p>	<ul style="list-style-type: none">- Se debe seleccionar la opción Carga inicial.- Se debe buscar el trabajador al que se le desea realizar la operación, introduciendo los datos requeridos.- Se debe presionar el botón Buscar.- Se debe seleccionar el trabajador encontrado.- Se debe presionar el botón Aceptar.- Se deben introducir en el formulario los datos dejando campos requeridos en blanco.- Se presiona el botón
--	--	---	--

Capítulo 4. Validación de los Requisitos

		<p>EP 1.3: Adicionar un submayor de vacaciones introduciendo errores en los datos.</p>	<ul style="list-style-type: none">- Se debe seleccionar la opción Carga inicial.- Se debe buscar el trabajador al que se le desea realizar la operación, introduciendo los datos requeridos.- Se debe presionar el botón Buscar.- Se debe seleccionar el trabajador encontrado.- Se debe presionar el botón Aceptar.- Se deben introducir en el formulario los datos introduciendo datos erróneos.- Se presiona el botón Aceptar.
--	--	--	---

Capítulo 4. Validación de los Requisitos

		<p>EP 1.4: Cancelar operación.</p>	<ul style="list-style-type: none"> - Se debe seleccionar la opción Carga inicial. - Se debe buscar el trabajador al que se le desea realizar la operación, introduciendo los datos requeridos. - Se debe presionar el botón Buscar. - Se debe seleccionar el trabajador encontrado. - Se debe presionar el botón Aceptar. - Se deben introducir o no en el formulario los datos. - Se debe presionar el botón Cancelar.
--	--	------------------------------------	--

Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Acumulado días.	Campo de texto.	NO	Combinación números.
2	Acumulado importe.	Campo de texto.	NO	Combinación números.

Capítulo 4. Validación de los Requisitos

3	Operación.	Lista desplegable.	NO	Selección.
4	Cuenta.	Lista desplegable.	NO	Selección.

Juegos de datos a probar

Id del escenario	Escenario	Acumulado días	Acumulado importe	Operación	Cuenta	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar un submayor de vacaciones introduciendo los datos correctamente.	V(0)	V(12)	V(Carga Inicial)	V(Salarios)	Se muestra un mensaje: "El submayor de vacaciones fue insertado satisfactoriamente."	
EP 1.2	Adicionar un submayor de vacaciones dejando campos requeridos en blanco.	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)."	
		V(0)	I(vacío)	I(vacío)	V(Salarios)		
		I(vacío)	V(12)	V(Carga Inicial)	I(vacío)		
		V(0)	V(12)	V(Carga Inicial)	I(vacío)		

Capítulo 4. Validación de los Requisitos

EP 1.3	Adicionar un submayor de vacaciones introduciendo errores en los datos.	I(as)	V(12)	NA	NA	Se muestra un mensaje de error: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s).".	
		V(0)	I(ase)	NA	NA		
		I(df33)	I(\$%GGH)	NA	NA		
EP 1.4	Cancelar la operación.	NA	NA	NA	NA	Se cancela la operación y se cierra la ventana.	

Ajuste del submayor de vacaciones

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 Submayores.
- Se debe seleccionar el menú Submayor de vacaciones.
- Se debe seleccionar el botón **Ajuste**.

Requisitos a probar

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

Capítulo 4. Validación de los Requisitos

1. Ajuste del submayor de vacaciones.	Se actualizan los datos del submayor de vacaciones de un trabajador.	EP 1.1: Ajustar un submayor de vacaciones introduciendo los datos correctamente.	<ul style="list-style-type: none"> - Se debe seleccionar el submayor de vacaciones a actualizar. - Se deben introducir en el formulario los datos correctamente. - Se debe presionar el botón Aceptar.
		EP 1.2: Ajustar un submayor de vacaciones dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se debe seleccionar el submayor de vacaciones a actualizar. - Se deben introducir en el formulario los datos dejando campos requeridos en blanco. - Se debe presionar el botón Aceptar.
		EP 1.3: Ajustar un submayor de vacaciones introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se debe seleccionar el submayor de vacaciones a actualizar. - Se deben introducir en el formulario los datos introduciendo datos erróneos. - Se debe presionar el botón Aceptar.

Capítulo 4. Validación de los Requisitos

		EP 1.4: Cancelar operación.	<ul style="list-style-type: none"> - Se debe seleccionar el submayor de vacaciones a actualizar. - Se deben introducir o no en el formulario los datos. - Se debe presionar el botón Cancelar.
--	--	-----------------------------	--

Descripción de la variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Acumulado días.	Campo de texto.	NO	Combinación números.
2	Acumulado importe.	Campo de texto.	NO	Combinación números.
3	Operación.	Lista desplegable.	NO	Selección.

Juegos de datos a probar

Id del escenario	Escenario	Acumulado días	Acumulado importe	Operación	Respuesta del sistema	Resultado de la prueba
EP 1.1	Ajustar un submayor de vacaciones introduciendo los datos correctamente.	V(0)	V(12)	V(Carga Inicial)	Se muestra un mensaje: "El submayor de vacaciones fue ajustado satisfactoriamente."	

Capítulo 4. Validación de los Requisitos

EP 1.2	Ajustar un submayor de vacaciones dejando campos requeridos en blanco.	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)".
		V(0)	I(vacío)	I(vacío)	
		I(vacío)	V(12)	V(Carga Inicial)	
		V(0)	V(12)	V(Carga Inicial)	
EP 1.3	Ajustar un submayor de vacaciones introduciendo errores en los datos.	I(as)	V(12)	NA	Se muestra un mensaje de error: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)".
		V(0)	I(ase)	NA	
		I(df33)	I(\$%GGH)	NA	
EP 1.4	Cancelar la operación.	NA	NA	NA	Se cancela la operación y se cierra la ventana.

Cierre del submayor de vacaciones

Requisitos a probar

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

Capítulo 4. Validación de los Requisitos

1. Cierre del submayor de vacaciones.	del de	Se elimina un submayor de vacaciones.	EP 1.1: Cierre del submayor de vacaciones correctamente.	<ul style="list-style-type: none"> - Se debe seleccionar el submayor de vacaciones a eliminar. - Se debe presionar el botón Cierre. - Se debe presionar el botón Aceptar.
			EP 1.2: Cancelar operación.	<ul style="list-style-type: none"> - Se debe seleccionar el submayor de vacaciones a eliminar. - Se debe presionar el botón Cierre. - Se debe presionar el botón Cancelar.

Descripción de la variable

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

Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Cierre del submayor de vacaciones correctamente.	Se muestra un mensaje: "El submayor de vacaciones fue cerrado satisfactoriamente."	
EP 1.2	Cancelar operación.	Se cancela la operación y se cierra la ventana.	

4.1.2.b. Casos de prueba del Gestionar submayor de retenciones

Apertura del submayor de retenciones

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 Submayores.
- Se debe seleccionar el menú Gestionar submayor de retenciones.
- Se debe seleccionar el botón **Adicionar**.
- Se debe buscar un trabajador.

Requisitos a probar

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

Capítulo 4. Validación de los Requisitos

		<p>EP 1.2: Adicionar un submayor de retenciones dejando campos requeridos en blanco.</p>	<p>Se deben introducir los datos del submayor de retenciones en el formulario dejando al menos un campo requerido en blanco.</p> <p>Se debe presionar el botón Aplicar o Aceptar.</p>
		<p>EP 1.3: Adicionar un submayor de retenciones introduciendo errores en los datos.</p>	<p>Se deben introducir los datos del submayor de retenciones introduciendo errores en los datos.</p> <p>Se debe presionar el botón Aplicar o Aceptar.</p>
		<p>EP 1.4: Cancelar operación.</p>	<p>Se deben introducir o no los datos a adicionar.</p> <p>Se debe presionar el botón Cancelar.</p>

Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
----	-----------------	---------------	----------------	-------------

Capítulo 4. Validación de los Requisitos

1	Tipo de retención.	Lista desplegable.	NO	Selección.
2	Saldo de la deuda.	Campo de texto.	NO	Combinación de números.
3	Importe por plazo.	Campo de texto.	NO	Combinación de números.
4	Cantidad de plazos.	Campo de texto.	NO	Combinación de números.
5	Fecha de retención.	Campo fecha.	NO	Selección.
6	Sucursal bancaria.	Campo de texto.	NO	Combinación de letras.
7	Primera quincena.	Campo de texto.	NO	Combinación de números.
8	Segunda quincena.	Campo de texto.	NO	Combinación de números.

Juegos de datos a probar

Id del escenario	Escenario	Tipo de retención	Saldo de la deuda	Importe por plazo	Cantidad de plazos	Fecha de retención	Sucursal bancaria	Primera quincena	Segunda quincena	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar un submayor de retenciones introduciendo los datos correctamente.	V(Vivienda vinculada)	V(1000)	V(75)	V(10)	V(02/05/2008)	V(Metropolitanos)	V(50)	V(100)	Se muestra un mensaje: "El submayor de retenciones fue insertado satisfactoriamente."	

Capítulo 4. Validación de los Requisitos

EP 1.2	Adicionar un submayor de retenciones dejando campos requeridos en blanco.	I(Vacío)	I(Vacío)	I(Vacío)	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(200)	V(45)	V(5)	I(Vacío)	I(Vacío)	V(54)	V(200)	
		V(Vivienda vinculada)	I(Vacío)	I(Vacío)	V(12)	V(03/12/2008)	V(Popular)	V(95)	V(75)	
		I(Vacío)	V(100)	V(75)	V(10)	V(02/05/2008)	V(Metropolitanos)	V(50)	V(100)	
		V(Vivienda vinculada)	I(Vacío)	V(75)	V(10)	V(02/05/2008)	V(Metropolitanos)	V(50)	V(100)	
		V(Vivienda vinculada)	V(100)	V(75)	V(10)	V(02/05/2008)	V(Metropolitanos)	V(50)	V(100)	
		V(Vivienda vinculada)	V(100)	I(Vacío)	V(10)	I(Vacío)	V(Metropolitanos)	V(50)	V(125)	
		V(Vivienda vinculada)	V(100)	V(75)	I(Vacío)	V(02/05/2008)	I(Vacío)	V(50)	V(300)	

Capítulo 4. Validación de los Requisitos

		V(Vivienda vinculada)	I(Vacío)	V(75)	V(10)	V(02/05/2008)	V(Metropolitanos)	I(Vacío)	V(100)		
		V(Vivienda vinculada)	V(100)	V(75)	V(10)	I(Vacío)	V(Metropolitanos)	V(50)	I(Vacío)		
		I(Vacío)	V(100)	V(75)	V(10)	V(02/05/2008)	I(Vacío)	V(50)	V(100)		
EP 1.3	Adicionar un submayor de retenciones introduciendo errores en los datos.	I(234)	I(fff)	I(ff)	I(erer)	I(ere-ere)	I(456)	I(dfdf)	I(sdfs)	Se muestra un mensaje de error: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)".	
		I(i".\$)	V(200)	V(45)	V(5)	I(02-12-08)	I(.\$%TG)	V(54)	V(200)		
		V(Vivienda vinculada)	I(DES DES)	I(qwsa".)	V(12)	V(03/12/2008)	V(Popular)	V(95)	V(75)		
		I(123".)	V(100)	V(75)	V(10)	V(02/05/2008)	V(Metropolitanos)	V(50)	V(100)		
		V(Vivienda vinculada)	I(de34)	V(75)	V(10)	V(02/05/2008)	V(Metropolitanos)	V(50)	V(100)		
		V(Vivienda vinculada)	V(100)	V(75)	V(10)	V(02/05/2008)	V(Metropolitanos)	V(50)	V(100)		

Capítulo 4. Validación de los Requisitos

		V(Vivienda vinculada)	V(1000)	I(-.,)	V(10)	I(ki-12-12)	V(Metropolitanos)	V(50)	V(125)		
		V(Vivienda vinculada)	V(1000)	V(75)	I(*34)	V(02/05/2008)	I(.\$Des323)	V(50)	V(300)		
		V(Vivienda vinculada)	I(Vacío)	V(75)	V(10)	V(02/05/2008)	V(Metropolitanos)	I(Vacío)	V(100)		
		V(Vivienda vinculada)	V(1000)	V(75)	V(10)	I(3/5/08)	V(Metropolitanos)	V(50)	I(34de)		
		I(Vacío)	V(1000)	V(75)	V(10)	V(02/05/2008)	I(V1c89)	V(50)	V(100)		
EP 1.4	Cancelar operación.	NA	NA	NA	NA	NA	NA	NA	NA	Se cancela la operación y se cierra la ventana.	

Modificar Submayor de retenciones

Condiciones de ejecución

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

Capítulo 4. Validación de los Requisitos

- Se debe seleccionar el menú Capital Humano.
- Se debe seleccionar el menú Remuneración y nómina.
- Se debe seleccionar el menú Submayores.
- Se debe seleccionar el menú Gestionar submayor de retenciones.

Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Modificar submayor de retenciones.	Se modifica un submayor de retenciones.	EP 1.1: Modificar submayor de retenciones introduciendo los datos correctamente.	<ul style="list-style-type: none">- Se debe seleccionar el submayor de retenciones a modificar.- Se debe presionar el botón Modificar.- Se deben introducir los nuevos datos del submayor de retenciones en el formulario.- Se debe presionar el botón Aceptar.

Capítulo 4. Validación de los Requisitos

		<p>EP 1.2: Modificar submayor de retenciones dejando campos requeridos en blanco.</p>	<ul style="list-style-type: none"> - Se debe seleccionar el submayor de retenciones a modificar. - Se debe presionar el botón Modificar. - Se deben introducir los datos del submayor de retenciones en el formulario dejando al menos un campo requerido en blanco. - Se debe presionar el
		<p>EP 1.3: Modificar submayor de retenciones introduciendo errores en los datos.</p>	<ul style="list-style-type: none"> - Se debe seleccionar el submayor de retenciones a modificar. - Se debe presionar el botón Modificar. - Se deben introducir los datos del submayor de retenciones en el formulario introduciendo algunos datos inválidos. - Se debe presionar el

Capítulo 4. Validación de los Requisitos

			botón Aceptar .
		EP 1.4: Cancelar operación.	<ul style="list-style-type: none">Se debe seleccionar el submayor de retenciones a modificar.Se debe presionar el botón Modificar.Se deben introducir o no los datos del submayor de retenciones en el formulario.Se debe presionar el botón Cancelar.

Capítulo 4. Validación de los Requisitos

Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Tipo de retención.	Campo de texto.	NO	No permite escribir.
2	Saldo de la deuda.	Campo de texto.	NO	Combinación de números.
3	Importe por plazo.	Campo de texto.	NO	Combinación de números.
4	Cantidad de plazos.	Campo de texto.	NO	Combinación de números.
5	Fecha de retención.	Campo fecha.	NO	No permite escribir.
6	Sucursal bancaria.	Campo de texto.	NO	No permite escribir.
7	Primera quincena.	Campo de texto.	NO	Combinación de números.
8	Segunda quincena.	Campo de texto.	NO	Combinación de números.

Juegos de datos a probar

Id del escenario	Escenario	Tipo de retención	Saldo de la deuda	Importe por plazo	Cantidad de plazos	Fecha de retención	Sucursal bancaria	Primera quincena	Segunda quincena	Respuesta del sistema	Resultado de la prueba
------------------	-----------	-------------------	-------------------	-------------------	--------------------	--------------------	-------------------	------------------	------------------	-----------------------	------------------------

Capítulo 4. Validación de los Requisitos

EP 1.1	Modificar submayor de retenciones introduciendo los datos correctamente.	V(Vivienda vinculada)	V(1000)	V(75)	V(10)	V(02/05/2008)	V(Metropolitanos)	V(50)	V(100)	Se muestra un mensaje: "El submayor de retenciones fue actualizado satisfactoriamente."	
EP 1.2	Modificar submayor de retenciones dejando campos requeridos en blanco.	I(Vacío)	I(Vacío)	I(Vacío)	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(2000)	V(45)	V(5)	I(Vacío)	I(Vacío)	V(54)	V(200)		
		V(Vivienda vinculada)	I(Vacío)	I(Vacío)	V(12)	V(03/12/2008)	V(Popular)	V(95)	V(75)		
		I(Vacío)	V(1000)	V(75)	V(10)	V(02/05/2008)	V(Metropolitanos)	V(50)	V(100)		
		V(Vivienda vinculada)	I(Vacío)	V(75)	V(10)	V(02/05/2008)	V(Metropolitanos)	V(50)	V(100)		
		V(Vivienda vinculada)	V(1000)	V(75)	V(10)	V(02/05/2008)	V(Metropolitanos)	V(50)	V(100)		

Capítulo 4. Validación de los Requisitos

		V(Vivienda vinculada)	V(1000)	I(Vacío)	V(10)	I(Vacío)	V(Metropolitanos)	V(50)	V(125)		
		V(Vivienda vinculada)	V(1000)	V(75)	I(Vacío)	V(02/05/2008)	I(Vacío)	V(50)	V(300)		
		V(Vivienda vinculada)	I(Vacío)	V(75)	V(10)	V(02/05/2008)	V(Metropolitanos)	I(Vacío)	V(100)		
		V(Vivienda vinculada)	V(1000)	V(75)	V(10)	I(Vacío)	V(Metropolitanos)	V(50)	I(Vacío)		
		I(Vacío)	V(1000)	V(75)	V(10)	V(02/05/2008)	I(Vacío)	V(50)	V(100)		
EP 1.3	Modificar submayor de retenciones introduciendo errores en los datos.	I(234)	I(fff)	I(ff)	I(erer)	I(ere-ere)	I(456)	I(dfdf)	I(sdfs)	Se muestra un mensaje de error: "Por favor verifique nuevamente que hay campo(s) con valor(es)	
		I(i".\$)	V(2000)	V(45)	V(5)	I(02-12-08)	I(.\$%TG)	V(54)	V(200)		
		V(Vivienda vinculada)	I(DESDES)	I(qwsa".)	V(12)	V(03/12/2008)	V(Popular)	V(95)	V(75)		
		I(123".)	V(1000)	V(75)	V(10)	V(02/05/2008)	V(Metropolitanos)	V(50)	V(100)		

Capítulo 4. Validación de los Requisitos

	V(Vivienda vinculada)	I(de34)	V(75)	V(10)	V(02/05/2008)	V(Metropolitanos)	V(50)	V(100)	incorrecto(s) ..
	V(Vivienda vinculada)	V(100)	V(75)	V(10)	V(02/05/2008)	V(Metropolitanos)	V(50)	V(100)	
	V(Vivienda vinculada)	V(100)	I(-,)	V(10)	I(ki-12-12)	V(Metropolitanos)	V(50)	V(125)	
	V(Vivienda vinculada)	V(100)	V(75)	I(*34)	V(02/05/2008)	I(-.\$Des323)	V(50)	V(300)	
	V(Vivienda vinculada)	I(Vacío)	V(75)	V(10)	V(02/05/2008)	V(Metropolitanos)	I(Vacío)	V(100)	
	V(Vivienda vinculada)	V(100)	V(75)	V(10)	I(3/5/08)	V(Metropolitanos)	V(50)	I(34de)	
	I(Vacío)	V(100)	V(75)	V(10)	V(02/05/2008)	I(V1c89)	V(50)	V(100)	

Capítulo 4. Validación de los Requisitos

EP 1.4	Cancelar operación.	NA	Se cancela la operación y se cierra la ventana.								
-----------	------------------------	----	----	----	----	----	----	----	----	---	--

Cierre del submayor de retenciones

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 Submayores.
- Se debe seleccionar el menú Gestionar submayor de retenciones.

Requisitos a probar

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

Capítulo 4. Validación de los Requisitos

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

Descripción de variable

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

Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar submayor de retenciones correctamente.	Se muestra un mensaje: "El submayor de retenciones fue eliminado satisfactoriamente."	
EP 1.2	Cancelar operación.	Se cancela la operación y se cierra la ventana.	

4.2. Métricas para la validación de los requisitos

La medición permite tener una visión más profunda proporcionando un mecanismo para la evaluación objetiva. Lord Kelvin en una ocasión dijo:

"Cuando pueda medir lo que estoy diciendo y expresarlo con números, ya conoce algo sobre ello; cuando no pueda medir, cuando no pueda expresar lo que dice con números, su conocimiento es precario y

deficiente: puede ser el comienzo del conocimiento, pero en sus pensamientos, apenas está avanzando hacia el escenario de la ciencia.” (23)

4.2.1. Criterios para la evaluación y aceptación de los requisitos

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

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

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

Observaciones de consistencia: Son las observaciones realizadas por contradicciones existentes entre los requisitos. (27)

Para aceptar una Especificación de requisitos debe cumplirse que:

- La correctitud sea menor 0, 10.
- La completitud sea mayor que 0,90.
- La consistencia sea menor que 0,20.

4.2.1.a. Correctitud

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

$$X = D/T$$

X – Correctitud.

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

T – Total de requisitos revisados.

La métrica se aplicará en la actividad de validación previa a la aprobación de los requisitos.

Un menor valor de X determina una mayor correctitud de la Especificación de requisitos.

4.2.1.b. Completitud

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

$$X = 1 - O/S$$

X – Completitud

O – Total de observaciones de formato.

S – Total de secciones del documento.

La métrica se aplicará en la actividad de validación previa a la aprobación de los requisitos.

Mientras más cercano a uno el valor de X más completa será la Especificación de requisitos.

4.2.1.c. Consistencia

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

$$X = C/T$$

X – Consistencia

C – Total de observaciones de consistencia.

T – Total de requisitos revisados.

La métrica se aplicará en la actividad de validación previa a la aprobación de los requisitos.

Un menor valor de X determina una mayor consistencia de la Especificación de requisitos.

Teniendo en cuenta lo antes mencionado se presentan a continuación cada una de estas métricas aplicadas a los requisitos definidos en este trabajo.

Variables necesarias para llevar a cabo cada una de las métricas:

- Observaciones de negocio de los requisitos (D) = 3.
- Observaciones de formato de los requisitos (O) = 0.
- Observaciones de consistencia de los requisitos (C) = 6.
- Total de requisitos revisados (T) = 35.
- Total de secciones del documento (S) = 6.

La tabla que muestra el listado de requisitos con las observaciones realizadas en la actividad de validación se muestra en los [Anexos](#).

Correctitud

$$X = D/T$$

$$X = 3/35$$

$$X = 0.09$$

Compleitud

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

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

$$X = 1$$

Consistencia

$$X = C/T$$

$$X = 6/35$$

$$X = 0.17$$

Después de obtener los resultados que se muestran en cada una de las ecuaciones anteriores, se puede llegar a la conclusión de que como la Correctitud es menor que 0.10, la Compleitud es mayor que 0.90 y que la Consistencia es menor que 0.20, entonces la especificación de los requisitos puede ser aceptada satisfactoriamente, ya que tienen una buena Correctitud, Compleitud y Consistencia.

4.3. Métricas orientadas a clases

4.3.1. 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. (28)

Capítulo 4. Validación de los Requisitos

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 junto con los umbrales seleccionados para su clasificación.

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

Tabla 2. Valores de los umbrales para TC

A continuación se ilustran las clases del sistema aplicándole la métrica seleccionada.

No	Nombre	Cantidad de atributos	Cantidad de operaciones	Tamaño
1	Gestionar Submayor de vacaciones Controller	0	8	Pequeño
2	Dat Submayor de vacaciones Model	0	8	Pequeño
3	Dat Submayor de vacaciones	0	7	Pequeño
4	Gestionar Submayor de retenciones Controller	0	13	Pequeño
5	Dat Submayor de retenciones Model	0	6	Pequeño
6	Dat Subamayor retencione	0	5	Pequeño
7	Gestionar Programa Controller	0	6	Pequeño
8	Nom Programa Model	0	5	Pequeño
9	Nom Programa	0	5	Pequeño
10	Gestionar Configuración submayor de vacaciones Controller	0	4	Pequeño
11	Nom Configuración submayor vacaciones Model	0	5	Pequeño
12	Nom Configuración submayor vacaciones	0	5	Pequeño
13	Gestionar Concepto retenciones Controller	0	5	Pequeño
14	Dat Concepto retenciones Model	0	6	Pequeño

Capítulo 4. Validación de los Requisitos

15	Dat Concepto retenciones	0	5	Pequeño
16	Gestionar Operación Controller	0	5	Pequeño
17	Nom Operación sub Model	0	5	Pequeño
18	Nom Operaciones sub	0	5	Pequeño
19	Gestionar Tipo recargo Controller	0	5	Pequeño
20	Nom Tipo recargo Model	0	5	Pequeño
21	Nom Tipo recargo	0	5	Pequeño
22	Gestionar Fichero histórico devengado Controller	0	6	Pequeño
23	Dat Fichero histórico devengado Model	0	4	Pequeño
24	Dat Fichero histórico devengado	0	6	Pequeño
25	Dat Registro de retenciones Model	0	5	Pequeño
26	Dat Registro de retenciones	0	7	Pequeño
27	Gestionar Tipo retención Controller	0	5	Pequeño
28	Nom Tipo retención Model	0	5	Pequeño
29	Nom Tipo retención	0	6	Pequeño

Tabla 3. Tamaño de las clases

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. Todas 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.

Clasificación	Cantidad Clases
Pequeño	29
Medio	0
Grande	0

Tabla 4. Cantidad de clases por clasificación

Cantidad Clases	Promedio Atributos	Promedio Operaciones
29	0	5.76

Tabla 5. Resultado la métrica TC

4.3.2. 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 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. (28)

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.

4.3.3. 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. (28)

No	Nombre	Relaciones de uso
1	Gestionar Submayor de vacaciones Controller	5
2	Dat Submayor de vacaciones Model	0
3	Dat Submayor de vacaciones	0
4	Gestionar Submayor de retenciones Controller	5
5	Dat Submayor de retenciones Model	0

Capítulo 4. Validación de los Requisitos

6	Dat Subamayor retencione	0
7	Gestionar Programa Controller	0
8	Nom Programa Model	0
9	Nom Programa	0
10	Gestionar Configuración submayor de vacaciones Controller	1
11	Nom Configuración submayor vacaciones Model	0
12	Nom Configuración submayor vacaciones	0
13	Gestionar Concepto retenciones Controller	0
14	Dat Concepto retenciones Model	0
15	Dat Concepto retenciones	0
16	Gestionar Operación Controller	0
17	Nom Operación sub Model	0
18	Nom Operaciones sub	0
19	Gestionar Tipo recargo Controller	0
20	Nom Tipo recargo Model	0
21	Nom Tipo recargo	0
22	Gestionar Fichero histórico devengado Controller	2
23	Dat Fichero histórico devengado Model	0
24	Dat Fichero histórico devengado	0
25	Dat Registro de retenciones Model	0
26	Dat Registro de retenciones	0
27	Gestionar Tipo retención Controller	1
28	Nom Tipo retención Model	0
29	Nom Tipo retención	0

Tabla 6. Cantidad de relaciones de uso entre las clases

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

Categoría	Relaciones de uso	Cantidad de Clases
-----------	-------------------	--------------------

Capítulo 4. Validación de los Requisitos

Ninguno	0	24
Bajo	1	2
Medio	2	1
Alto	>2	2

Tabla 7. Acoplamiento

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 0.48.

Categoría	Criterio	Cantidad de Clases
Baja	\leq Promedio.	24
Media	$>$ Promedio. Y $\leq 2^*$ Promedio.	0
Alta	$> 2^*$ Promedio.	5

Tabla 8. Cantidad de prueba y complejidad de mantenimiento

Categoría	Criterio	Cantidad de Clases
Baja	$> 2^*$ Promedio.	5
Media	$>$ Promedio. Y $\leq 2^*$ Promedio.	0
Alta	\leq Promedio.	24

Tabla 9. Reutilización

De manera general los resultados obtenidos de esta métrica son positivos. El acoplamiento existente entre las clases es bajo, el 82.75% no tiene ningún acoplamiento, el 6.9% es bajo, el 3.4% es medio, y solo el 6.9% tiene alto acoplamiento. El nivel de reutilización de las clases es alto, puesto que el 82.75% de las clases pueden ser reutilizadas. De igual forma la cantidad de pruebas y la complejidad de mantenimiento tienen que el 82.75% de las clases son fáciles de reparar y la cantidad de pruebas a realizar es relativamente corta. Estos resultados se visualizan en el siguiente gráfico:

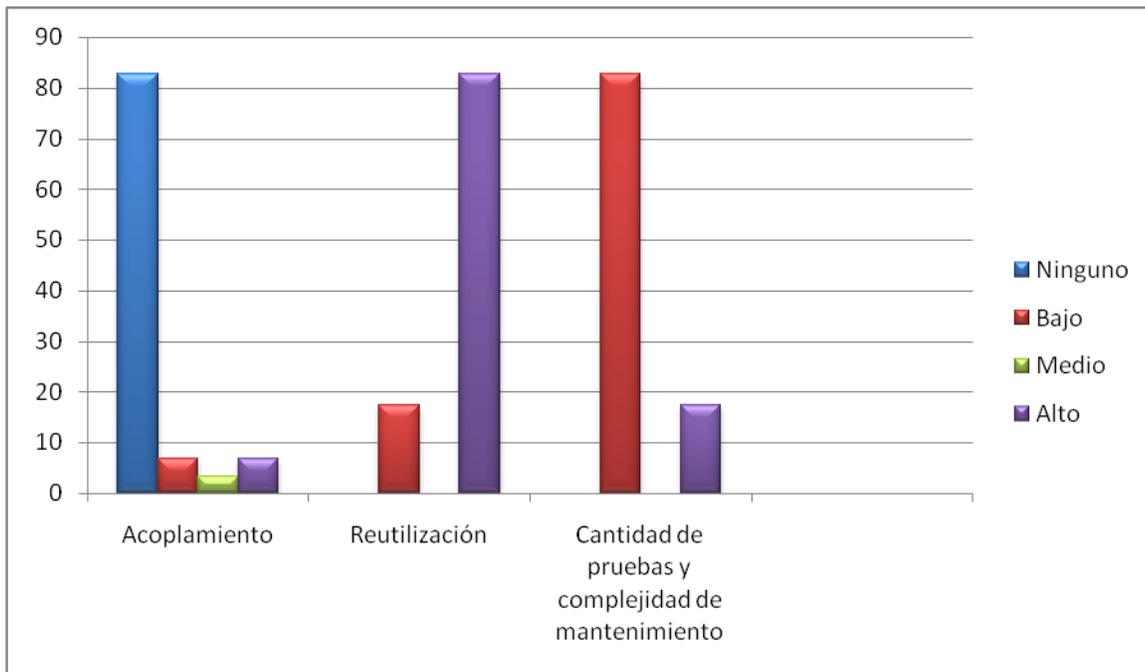


Fig.12. Resultados de la métrica Relaciones entre clases

Conclusiones parciales del capítulo

En este capítulo se aplicaron las técnicas de validación de los requisitos demostrando que los mismos cumplen con lo que realmente el usuario necesita o desea, se mostraron las métricas que arrojaron como resultado que las especificaciones de los requisitos pueden ser aceptadas satisfactoriamente, además de

Capítulo 4. Validación de los Requisitos

las métricas para conocer la efectividad con que fueron aplicados los distintos patrones del diseño en la construcción de los diferentes diagramas de clases.

CONCLUSIONES

Con la realización del presente trabajo se obtuvo la documentación necesaria referente al análisis y el diseño del componente Submayores del subsistema Capital Humano del sistema Cedrux, de modo que se crearon las bases para dar paso a la implementación de dicho componente.

A lo largo de esta investigación se realizó un estudio de los software que existen tanto en Cuba como a nivel internacional que gestionan los procesos contables en las empresas. Se hizo un análisis de cada uno de estos sistemas y se tomaron las ideas positivas que permitieron enriquecer la solución propuesta. Además se describieron las principales herramientas que se usaron para el modelado del análisis y el diseño y se crearon los diferentes artefactos que propone el Modelo de desarrollo orientado a componentes.

Con las técnicas seleccionadas para el levantamiento de los requisitos fueron definidos todos los procesos del negocio que se llevan a cabo para la creación de los Submayores en las entidades presupuestadas y empresariales, logrando de esta forma una buena especificación de cada uno de los requisitos del sistema. La correctitud, completitud y consistencia, arrojaron resultados positivos que permitieron demostrar que la especificación de los requisitos puede ser aceptada por los involucrados en el negocio.

Se confeccionaron y describieron los diagramas del diseño, donde se aplicaron diferentes patrones tanto del diseño como de asignación de responsabilidades. Además de la validación realizada a los diagramas de clases a través de un conjunto de métricas, las cuales mostraron resultados satisfactorios.

Con la solución propuesta se minimizará el trabajo que se realiza de forma manual, se facilitará el manejo de los datos y se contribuirá a la seguridad e integridad de la información en las entidades, obteniendo una mayor eficiencia y calidad en el trabajo. Logrando de esta manera el cumplimiento a los objetivos propuestos en el presente trabajo de diploma.

RECOMENDACIONES

Basándose en la investigación realizada y en la experiencia acumulada durante la realización de este trabajo de diploma se recomienda:

- Implementar la solución propuesta.
- Continuar investigando las necesidades de las distintas entidades, con el propósito de agregarle nuevas funcionalidades al sistema.

REFERENCIA BIBLIOGRÁFICA

1. **bsigroup**. www.bsigroup.com.mx. *www.bsigroup.com.mx*. [Online] The British Standards Institution, 2009. <http://www.bsigroup.com.mx>.
2. **assets**. assets.co.cu . *assets.co.cu* . [Online] Assets, 2004. [Cited: Octubre 25, 2008.] <http://assets.co.cu>.
3. **sagemas**. [sagemas.com/products/sagemas500](http://www.sagemas.com/products/sagemas500). *sagemas.com*. [Online] 2008. [Cited: Noviembre 20, 2008.] <http://www.sagemas.com/products/sagemas500>.
4. **seven**. [seven2k.com/index.html](http://www.seven2k.com/index.html). *www.seven2k.com/index.html*. [Online] 2008. [Cited: Noviembre 20, 2008.] <http://www.seven2k.com/index.html>.
5. **desoft**. [desoft.cu](http://www.desoft.cu). [Online] Desoft S.A., 2008. www.desoft.cu.
6. **rodasxxi**. [rodasxxi.cu](http://www.rodasxxi.cu). [Online] CITMATEL, 2009. www.rodasxxi.cu.
7. **3000, Norma cubana**. *Norma cubana 3000*. 2007.
8. **Maribel Ariza Rojas, Juan Carlos Molina García**. *Modelo de desarrollo orientado a componente*. 2004.
9. **Ivar Jacobson, Grady Booch, James Rumbaugh**. *El lenguaje unificado de modelado. Manual de referencia*. s.l. : Addison-Wesley, 2000.
10. **White, Stephen A**. *Introduction to BPMN*. s.l. : IBM Corporation.
11. **Alarcón, Raúl**. *Diseño Orientado a Objetos con UML*. s.l. : Grupo EIDOS, 2000.
12. **Paradigm, Visual**. visual-paradigm.com/product/vpuml/. *visual-paradigm.com*. [Online] Visual Paradigm, 2008. [Cited: Diciembre 20, 2008.] <http://www.visual-paradigm.com/product/vpuml/>.
13. **lenjuageprog**. [desarrolloweb.com](http://www.desarrolloweb.com). *desarrolloweb.com*. [Online] 2009. [Cited: Enero 12, 2009.] <http://www.desarrolloweb.com>.
14. **PHP**. *Manual de PHP*. 2007.
15. **maestrosdelweb**. [aestrosdelweb.com](http://www.maestrosdelweb.com). *maestrosdelweb.com*. [Online] 2008. [Cited: Enero 12, 2009.] <http://www.maestrosdelweb.com>..
16. **Madrid, Universidad Complutense de**. *Introducción a las base de datos*. España : s.n.
17. **postgre**. http-peru.com. *http-peru.com*. [Online] 2008. [Cited: Diciembre 23, 2008.] <http://www.http-peru.com>..
18. **masadelante**. masadelante.com/faqs/servidor. *masadelante.com*. [Online] Web Design Spain, 2009. [Cited: Mayo 3, 2009.] <http://masadelante.com/faqs/servidor>.

19. **apache**. apache.com. *apache.com*. [Online] [Cited: Enero 24, 2009.] <http://www.apache.com>.
20. *www.intersedes.ucr.ac.cr*. **intersedes**. 10, s.l. : Revista InterSedes © Universidad de Costa Rica, 2005, Vol. VI.
21. **Larman, Craig**. *UML y Patrones. Una introducción al análisis y el diseño orientado a objetos y al proceso unificado*. México : Cámara Nacional de la Industria Editorial Mexicana, 1999. 1524.
22. **netwok, Software and Systems Process Improvement**. latinspin.org. [Online] SPIN, 2006. www.latinspin.org.
23. **Preesman, Roger S**. *Ingenieria de Software. Un enfoque practico*.
24. **buenmaster**. buenmaster.com. *buenmaster.com*. [Online] 2009. [Cited: Diciembre 23, 2008.] <http://buenmaster.com>.
25. **Gamma, Erich**. *Design Patterns: Elements of Reusable Object-Oriented Software by ErichGamma*. s.l. : Pearson Education, 1994.
26. **hgv9651software**. hgv9651software2.blogspot.com. *hgv9651software2.blogspot.com*. [Online] 2009. [Cited: Octubre 25, 2008.] <http://hgv9651software2.blogspot.com>.
27. **IEEE-STD-830-1998**. IEEE-STD-830-1998 : ESPECIFICACIONES DE LOS REQUISITOS DEL.
28. **Ivar Jacobson, Grady Booch, James Rumbaugh**. *El proceso unificado de desarrollo de software*. España : Addison-Wesley, 2000.

BILIOGRAFÍA

1. **social, Ministerio de trabajo y seguridad.** *Decreto Ley 91.* Ciudad Habana : s.n., 2000.
2. **havasoft.** www.havasoft.minaz.cu . www.havasoft.minaz.cu . [Online]
3. **Schmuller, Joseph.** *Aprendiendo UML en 24 horas.* Mexico : Prentice Hall, 2000.
4. **requerimientos., Conferencia 4 Flujo de trabajo de.** teleformacion.uci.cu. teleformacion.uci.cu/file.php/102/Curso_2008-. [Online] uci, 2007. [Cited: 5 13, 2009.] <http://teleformacion.uci.cu>.
5. **paradigm, Visual.** Visual Paradigm para UML. freedownloadmanager.org. [Online] 2007. www.freedownloadmanager.org.
6. citmatel.cu. citmatel.cu. [Online] CITMATEL, 2005. [Cited: Abril 2, 2009.] <http://www.citmatel.cu>.
7. Resolución No. 27/06. Cuba : s.n., 2006.
8. **Telecomunicaciones, E.T.S.I. Informática y.** lsi.ugr.es. lsi.ugr.es. [Online] Universidad de Granada, Noviembre 19, 2008. [Cited: Abril 2, 2009.] <http://lsi.ugr.es>.
9. **Computación, Departamento de Ingeniería de Sistemas y.** agamenon.uniandes.edu.co. agamenon.uniandes.edu.co. [Online] Universidad de los Andes , Mayo 15, 2009. [Cited: Mayo 16, 2009.] <http://agamenon.uniandes.edu.co>.
10. **Gracia, Joaquin.** ingenierosoftware.com. ingenierosoftware.com. [Online] 2003. [Cited: Marzo 12, 2009.] <http://www.ingenierosoftware.com>.
11. teleformacion.uci.cu/file.php/102/Curso_2008-2009/Materiales_Basicos/Materiales_Basicos_Conf_5/Conferencia_Requerimientos_2007-2008.pdf. teleformacion.uci.cu. [Online] 2009. <http://teleformacion.uci.cu>.
12. **requerimientos., Conferencia 4 Fase de Inicio. Flujo de trabajo de.** teleformacion.uci.cu. teleformacion.uci.cu. [Online] [Cited: 5 13, 2009.] http://teleformacion.uci.cu/file.php/102/Curso_2008-2009/Materiales_Basicos/Materiales_Basicos_Conf_4/Conf_4_FT_Requerimiento.doc.
13. **diseño., Conferencia 2 Arquitectura y Patrones de.** teleformacion.uci.cu. teleformacion.uci.cu. [Online] 2009. [Cited: Mayo 13, 2009.] <http://teleformacion.uci.cu>.
14. **implementación, Conferencia Flujo de.** teleformacion.uci.cu/file.php/259/CURSO_2008-2009/Materiales_Basicos/Semana_7/Conf/Conferencia_Implementacion.pdf . teleformacion.uci.cu. [Online] 2009. [Cited: Mayo 15, 2009.] <http://teleformacion.uci.cu>.
15. **Española, Real Academia.** rae.es. rae.es. [Online] <http://www.rae.es>.

16. **Brito, Henry Raúl González.** ERP cubano, un paso estratégico para la consolidación del Software Libre en Cuba. Cuba : s.n., 2006.
17. **Azúcar, Ministerio del.** havasoft.minaz.cu. *havasoft.minaz.cu*. [Online] HavaSoft, 2009. [Cited: Abril 3, 2009.] <http://www.havasoft.minaz.cu>.
18. **Lucas, Robert.** *On the Mechanics of Development Planning*. s.l. : Journal of Monetary Economics, 1988.
19. **Sala-i-Martin, Xavier.** *Apuntes de crecimiento económico*. barcelona : Antoni Bosch, 2000.
20. **Precios, Ministerio de Finanzas y.** Resolución No. 105-2007. *Resolución No. 105-2007*. Cuba : s.n., 2007.
21. **diseño., Conferencia 2 Arquitectura y Patrones de.** teleformacion.uci.cu/file.php/259/CURSO_2008-2009/Materiales_Basicos/Semana_3/Conf/Conferencia_2_de_Arquitectura_2009ok.doc. *teleformacion.uci.cu*. [Online] uci, 2007. [Cited: 5 13, 2009.] http://teleformacion.uci.cu/file.php/259/CURSO_2008-2009/Materiales_Basicos/Semana_3/Conf/Conferencia_2_de_Arquitectura_2009ok.doc.
21. **White, Stephen A.** bpmn.org. *bpmn.org*. [En línea] <http://www.bpmn.org>.
22. **Alvarez, Katya Olivares.** Revista electrónica de derecho informático. *blogs.antartec.com*. [Online] [Cited: Abril 23, 2009.] <http://blogs.antartec.com>.
23. **Alvarez, Miguel Angel.** desarrolloweb.com. *desarrolloweb.com*. [Online] [Cited: Marzo 12, 2009.] <http://www.desarrolloweb.com..>
24. **Henst, Christian Van Der.** maestrosdelweb.com. *maestrosdelweb.com*. [Online] [Cited: Febrero 19, 2009.] <http://www.maestrosdelweb.com>.
25. **ASPgems, Loogic.** pymecrunch.com. *pymecrunch.com*. [Online] Neuseed Creative Solutions. [Cited: Diciembre 1, 2008.]
26. **Rica, Universidad de Costa.** intersedes.ucr.ac.cr. *intersedes.ucr.ac.cr*. [Online] Julio 26, 2007. [Cited: Octubre 12, 2008.] <http://www.intersedes.ucr.ac.cr>.
27. **Precios, Ministerio de Finanzas Y.** *Resolución No. 105-2008 (Contribución especial de los trabajadores beneficiarios de la Seguridad Social)*. Cuba : Ministerio de Finanzas Y Precios, 2008.
28. —. *Resolución No. 116-2002 (La Contribución a la Seguridad Social, a la cual están obligadas todas las entidades)*. Cuba : Ministerio de Finanzas Y Precios, 2002.
29. —. *Resolución No. 198-2006 (Escala de salario del MES)*. Cuba : Ministerio de Finanzas Y Precios, 2006.

30. **Precios, Ministerio de Finanzas y.** *Resolución 13 del 2007 (Objetivo de la nómina, campos obligatorios)*. Cuba : Ministerio de Finanzas y Precios, 2007.
31. **Social, Ministerio de Trabajo y Seguridad.** *Resolución 27 del 2006 (Reglamento general organización del salario)*. Cuba : Ministerio de Trabajo y Seguridad Social, 2006.
32. **social, Ministerio de trabajo y seguridad.** *Ley No. 49 de 1985 (Se establecen las regulaciones relativas a las Vacaciones Anuales Pagadas.)*. Cuba : Ministerio de trabajo y seguridad social, 1985.
33. **Precios, Ministerio de Finanzas y.** *Ley 73 de 1994 (Sistema Tributario)*. Cuba : Ministerio de Finanzas y Precios, 1994.
34. **network, Software and Systems Process Improvement.** [Online]

ANEXOS

Prototipos de interfaz de usuario

Anexo1: Adicionar tipo de retención

Adicionar tipo de retención

Código

Denominación

Fecha inicio

Prioridad

Condición de parada

Cuenta

Cancelar Aplicar Aceptar

Anexo 2: Modificar tipo de retención

Modificar tipo de retención

Código

Denominación

Fecha inicio

Prioridad

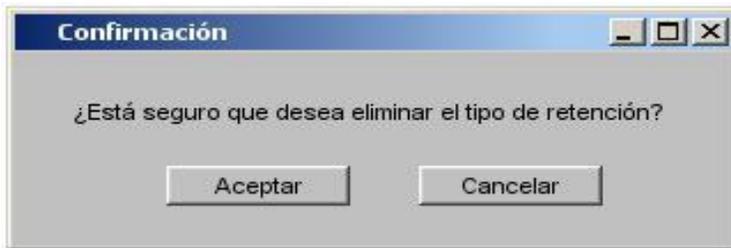
Condición de parada

Cuenta

Cancelar Aceptar

Este campo se mostrará deshabilitado

Anexo 3: Eliminar tipo de retención



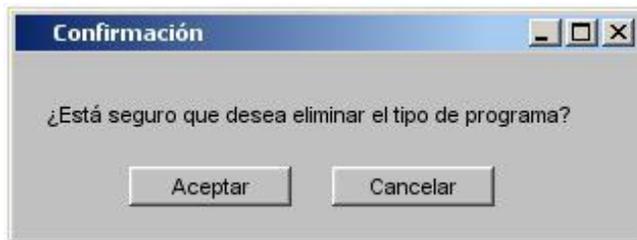
Anexo 4: Adicionar tipo de programa



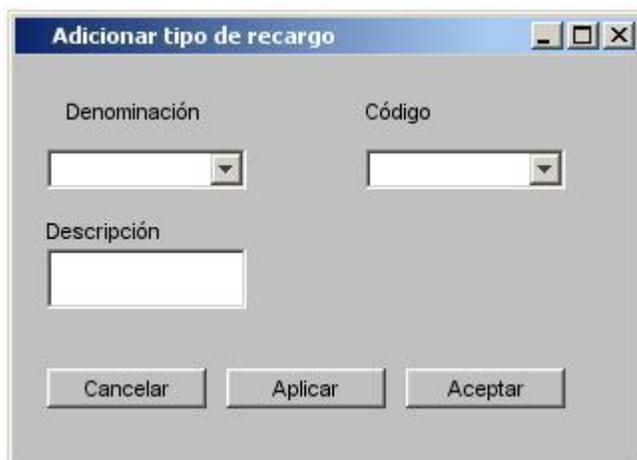
Anexo 5: Modificar tipo de programa



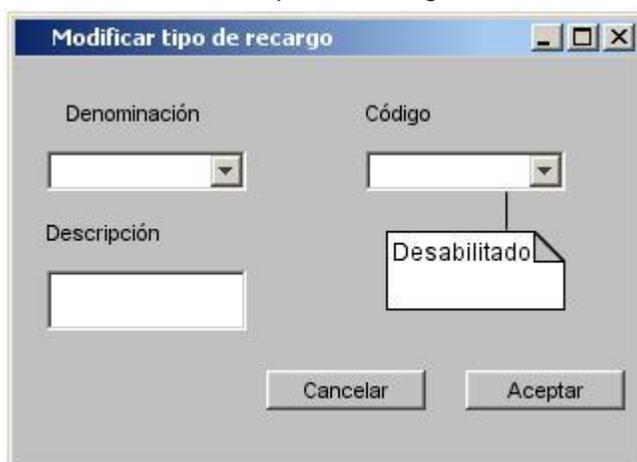
Anexo 6: Eliminar tipo de programa



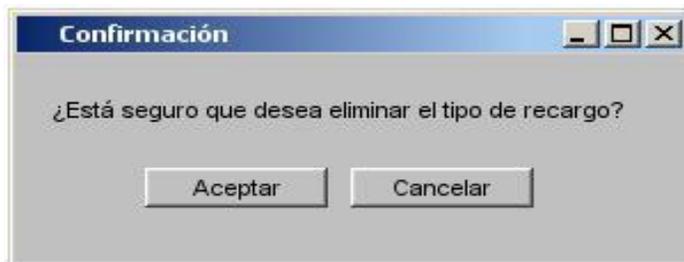
Anexo 7: Adicionar tipo de recargo



Anexo 8: Modificar tipo de recargo



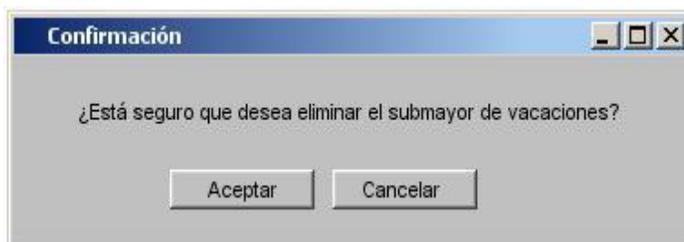
Anexo 9: Eliminar tipo de recargo



Anexo 10: Apertura del submayor de vacaciones

A form titled "Adicionar vacaciones" with a blue header bar. It contains several input fields and dropdown menus arranged in two columns. The fields are: "No carnet" (text input), "Nombre" (text input), "Primer apellido" (text input), "Segundo apellido" (text input), "Acumulado días" (text input), "Acumulado importe" (text input), "Operación" (dropdown menu), and "Cuenta" (dropdown menu). At the bottom, there are two buttons: "Cancelar" and "Aplicar".

Anexo 11: Cierre del submayor de vacaciones



Anexo 12: Detalles del submayor de vacaciones

Detalles del submayor de vacaciones

Fecha	Nómina	Acumulado período	Acumulado importe	Acumulado total días	Acumulado total importe	Operación

Aceptar

Anexo 13: Configuración del submayor de vacaciones

Configuración

% Vacaciones

Cuenta

Cancelar Aceptar

Anexo 14: Ajuste del submayor de vacaciones

Ajuste de vacaciones

No carnet

Nombre

Primer apellido

Segundo apellido

Ajuste días

Ajuste importe

Operación

Solo se modificarán los campos señalados.

Cancelar Aceptar

Anexo 15: Apertura del submayor de retenciones

Adicionar submayor de retenciones

Datos del trabajador

No carnet Nombres Primer apellido Segundo apellido

Tipo de retención

Saldo de la deuda Primera quincena

Importe por plazo Cantidad de plazos Segunda quincena

Fecha de retención Sucursal bancaria

Estos campos se muestran cargados con los datos del trabajador que resultó de la búsqueda

Cancelar Aplicar Aceptar

Anexo 16: Modificar submayor de retenciones

Modificar submayor de retenciones

Datos del trabajador

No carnet: Nombres: Primer apellido: Segundo apellido:

Tipo de retención: Saldo de la deuda: Primera quincena:

Importe por plazo: Cantidad de plazos: Segunda quincena:

Fecha de retención: Sucursal bancaria:

Desabilitado

Desabilitado

Estos campos se muestran cargados con los datos del trabajador que resultó de la búsqueda

Cancelar Aceptar

Anexo 17: Cierre del submayor de retenciones

Confirmación

¿Está seguro que desea eliminar el submayor de retenciones?

Aceptar Cancelar

Anexo 20: Pago directo al banco del submayor de retenciones

The screenshot shows a window titled "Pago directo al banco" with a blue header bar. Below the header, the text "Datos del trabajador" is displayed. There are four input fields: "No. carnet", "Nombres", "Primer apellido", and "Segundo apellido". Below these are two more input fields: "Tipo de retención" and "Importe". A tooltip box points to the "Importe" field with the text "Este es el único campo que se llena". At the bottom of the window are two buttons: "Cancelar" and "Aceptar".

Anexo 21: Alerta de cumplido con el pago de la retención de los trabajadores

The screenshot shows a window titled "Alerta de cumplido con el pago de la retención" with a blue header bar. Below the header is a table with five columns: "Expediente", "Nombre", "Primer A", "Segundo A", and "Tipo retención". The table contains several empty rows.

Expediente	Nombre	Primer A	Segundo A	Tipo retención

Anexo 22: Adicionar recargo a la retención

The dialog box titled "Adicionar tipo de recargo a la retención" contains the following fields and controls:

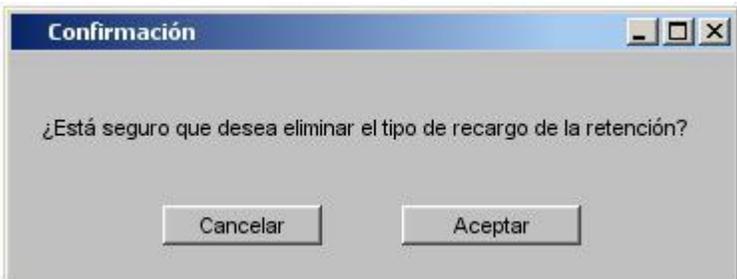
- Fecha:** A dropdown menu.
- Cantidad de plazo:** A text input field.
- Id retención:** A text input field.
- Id recargo:** A text input field.
- Importe:** A text input field.
- Buttons:** "Cancelar", "Aplicar", and "Aceptar" at the bottom.

Anexo 23: Modificar recargo a la retención

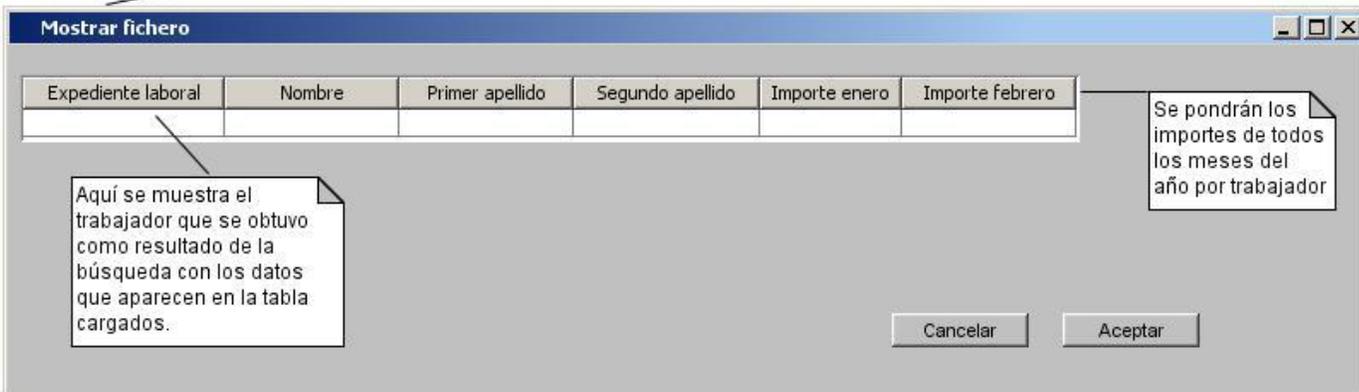
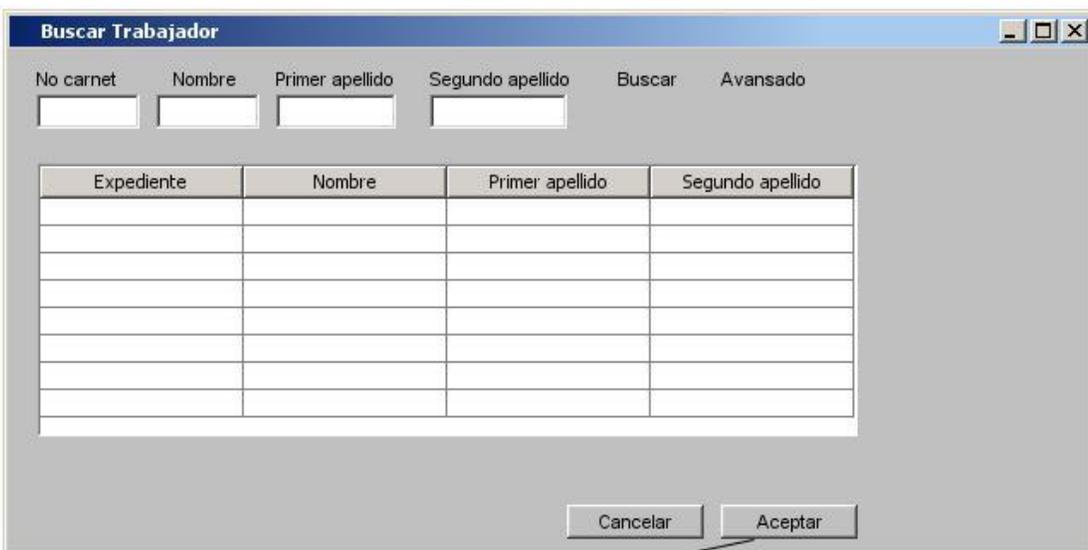
The dialog box titled "Modificar tipo de recargo a la retención" contains the same fields and controls as Anexo 22, but with additional annotations:

- Fecha:** A dropdown menu with a callout box pointing to it that says "Aparece deshabilitado".
- Id retención:** A text input field with a callout box pointing to it that says "Aparece deshabilitado".
- Id recargo:** A text input field with a callout box pointing to it that says "Aparece deshabilitado".
- Buttons:** "Cancelar", "Aplicar", and "Aceptar" at the bottom.

Anexo 24: Eliminar recargo a la retención



Anexo 25: Mostrar fichero



Anexo 26: Modificar fichero

Buscar Trabajador

No carnet Nombre Primer apellido Segundo apellido Buscar Avansado

Expediente	Nombre	Primer apellido	Segundo apellido

Cancelar Aceptar

Modificar fichero

Expediente laboral Nombre Primer apellido Segundo apellido

Importe enero Importe febrero

Aquí van los demás meses del año con los importes cargados, estos serán los únicos datos que se podrán modificar en este formulario.

Cancelar Aceptar

Anexo 27: Resumen devengado del trabajador en meses anteriores

The image shows two windows from a software application. The top window, titled "Filtrar", contains two date selection fields labeled "Fecha inicio" and "Fecha fin", each with a dropdown arrow. Below these fields are two buttons: "Cancelar" and "Aceptar". A line connects the "Aceptar" button to the bottom window.

The bottom window, titled "Resumen devengado del trabajador en el tiempo seleccionado", displays a table with the following columns: "No carnet", "Nombre", "Primer apellido", "Segundo apellido", and "Importe total período". Below the table is a text box containing the message: "Se muestra al trabajador con el total del salario de las fechas seleccionadas." At the bottom right of this window are two buttons: "Cancelar" and "Aceptar".

No carnet	Nombre	Primer apellido	Segundo apellido	Importe total período

Casos de Prueba

Anexo 28: Adicionar tipo de retención

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 Nomencladores.
- Se debe seleccionar el menú Tipo retención.

Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Adicionar tipo retención.	Se adiciona un tipo retención.	EP 1.1: Adicionar un tipo retención introduciendo los datos correctamente.	<ul style="list-style-type: none"> - Se deben introducir los datos del tipo retención correctamente en el formulario. - Se debe presionar el botón Aplicar o Aceptar.
		EP 1.2: Adicionar un tipo retención dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se deben introducir los datos del tipo retención en el formulario dejando al menos un campo requerido en blanco. - Se debe presionar el botón Aplicar o Aceptar.
		EP 1.3: Adicionar un tipo retención introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se deben introducir los datos del tipo retención 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 deben introducir o no los datos a adicionar. - Se debe presionar el botón Cancelar.
--	--	-----------------------------	--

Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Código.	Campo de texto	NO	Combinación de números.
2	Denominación.	Campo de texto.	NO	Combinación de letras.
3	Fecha inicio.	Elemento de fecha.	NO	Selección.
4	Prioridad.	Lista desplegable	NO	Selección.
5	Condición de parada.	Lista desplegable	NO	Selección.
6	Cuenta.	Lista desplegable	NO	Selección.

Juegos de datos a probar

Id del escenario	Escenario	Código	Denominación	Fecha inicio	Prioridad	Condición de parada	Cuenta	Respuesta del sistema	Resultado de la prueba
------------------	-----------	--------	--------------	--------------	-----------	---------------------	--------	-----------------------	------------------------

EP 1.1	Adicionar un tipo retención introduciendo los datos correctamente.	V(12)	V(Crédito a la población)	V(09/12/2008)	V(1)	V(Tiempo límite)	V(Caja)	Se muestra un mensaje: "El tipo de retención fue insertado satisfactoriamente."	
EP 1.2	Adicionar un tipo retención dejando campos requeridos en blanco.	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	V(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(Cuentas de ahorro)	I(Vacío)	V(1)	V(Tiempo límite)	V(Combustible)		
		V(7)	I(Vacío)	I(Vacío)	V(2)	V(Tiempo límite)	I(Vacío)		
		V(1)	V(Préstamo)	V(02/12/2008)	I(Vacío)	V(Vacío)	V(Energía)		
		I(Vacío)	I(Vacío)	I(Vacío)	V(3)	V(Vacío)	V(Caja)		
V(12358796)	I(Vacío)	V(01/12/2005)	I(Vacío)	V(Tiempo indefinido)	I(Vacío)				
EP 1.3	Adicionar un tipo retención introduciendo	I(fdgdg)	I(345)	I(rrrrrr)	I(df)	I(43543)	I(34534)	Se muestra un mensaje de error: "Por favor verifique	

ndo errores en los datos.	I(·\$\$%%)	V(Préstam o estudiantil)	I(12-12- 08)	V(1)	V(Tiem po límite)	V(Combustible)	nuevamente que hay campo(s) con valor(es) incorrecto(s).”
	V(78)	I(·\$%&)	I(3/5/08)	V(2)	V(Tiem po límite)	I(\$%&/)	
	V(1)	V(Cuentas de ahorro)	V(02/12/2 008)	I(gtrf)	I(34)	V(Energía)	
	I(34)	V(Crédito a la población)	I(ferer)	V(3)	V(Vacío)	V(Caja)	
	V(123)	I(V3o98)	V(01/12/2 005)	I(w)	V(Tiem po indefini do)	I(·DE·)	
	V(12)	I(3343##)	V(09/12/2 008)	V(1)	V(Tiem po límite)	V(Caja)	
	V(12)	V(Crédito a la población)	I(12-02- 08)	V(1)	V(Tiem po límite)	V(Caja)	
	V(12)	V(Crédito a la población)	V(09/12/2 008)	I(sw)	V(Tiem po límite)	V(Caja)	
	V(12)	V(Crédito a la población)	V(09/12/2 008)	V(1)	I(23\$\$)	V(Caja)	

		V(12)	V(Crédito a la población)	V(09/12/2008)	V(1)	V(Tiempo límite)	I(##\$DD333)		
EP 1.4	Cancelar operación.	NA	NA	NA	NA	NA	NA	Se cancela la operación y se cierra la ventana.	

Anexo 29: Modificar tipo de retención

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 Nomencladores.
- Se debe seleccionar el menú Tipo retención.

Requisitos a probar

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

1. Modificar tipo retención.	Se modifica un tipo retención.	EP 1.1: Modificar tipo retención introduciendo los datos correctamente.	<ul style="list-style-type: none"> - Se debe seleccionar el tipo de retención a modificar. - Se debe presionar el botón Modificar. - Se deben introducir los nuevos datos del tipo retención en el formulario. - Se debe presionar el botón Aceptar.
		EP 1.2: Modificar tipo retención dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se debe seleccionar el tipo de retención a modificar. - Se debe presionar el botón Modificar. - Se deben introducir los datos del tipo retención en el formulario dejando al menos un campo requerido en blanco. - Se debe presionar el botón Aceptar.
		EP 1.3: Modificar tipo retención introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se debe seleccionar el tipo de retención a modificar.

			<ul style="list-style-type: none"> - Se debe presionar el botón Modificar. - Se deben introducir los datos del tipo retención 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 seleccionar el tipo de retención a modificar. - Se debe presionar el botón Modificar. - Se deben introducir o no los datos del tipo retención en el formulario. - Se debe presionar el botón Cancelar.

Descripción de variable

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

3	Fecha inicio.	Elemento de fecha.	NO	Selección.
4	Prioridad.	Lista desplegable	NO	Selección.
5	Condición de parada.	Lista desplegable	SI	Selección.
6	Cuenta.	Lista desplegable	NO	Selección.

Juegos de datos a probar

Id del escenario	Escenario	Código	Denominación	Fecha inicio	Prioridad	Condición de parada	Cuenta	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar tipo retención introduciendo los datos correctamente.	NA	V(Denominación desde)	V(09/12/2008)	V(1)	V(Tiempo límite)	V(Caja)	Se muestra un mensaje: “El tipo de retención fue actualizado satisfactoriamente.”.	
EP 1.2	Modificar tipo retención dejando campos requeridos en blanco.	NA	I(Vacío)	I(Vacío)	I(Vacío)	V(Vacío)	I(Vacío)	Se muestra un mensaje de error: “ Por favor verifique nuevamente que hay campo(s) con valor(es)	
		NA	V(Cuentas de ahorro)	I(Vacío)	V(1)	V(Tiempo límite)	V(Combustible)		
		NA	I(Vacío)	I(Vacío)	V(2)	V(Tiempo límite)	I(Vacío)		

		NA	V(Préstamo estudiantil)	V(02/12/2008)	I(Vacío)	V(Vacío)	V(Energía)	incorrecto(s)."	
		NA	I(Vacío)	I(Vacío)	V(3)	V(Vacío)	V(Caja)		
		NA	I(Vacío)	V(01/12/2005)	I(Vacío)	V(Tiempo indefinido)	I(Vacío)		
EP 1.3	Modificar tipo retención introduciendo errores en los datos.	NA	I(435)	I(rrrrrr)	I(df)	I(43543)	I(34534)	Se muestra un mensaje de error: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)."	
		NA	V(Préstamo estudiantil)	I(12-12-08)	V(1)	V(Tiempo límite)	V(Combustible)		
		NA	I(.\$%&)	I(3/5/08)	V(2)	V(Tiempo límite)	I(.\$%&/)		
		NA	V(Cuentas de ahorro)	V(02/12/2008)	I(gtrf)	I(34)	V(Energía)		
		NA	V(Crédito a la población)	I(sfsdsd)	V(3)	V(Vacío)	V(Caja)		
		NA	I(V3o98)	V(01/12/2005)	I(w)	V(Tiempo indefinido)	I(·DE·)		

		NA	V(Crédito a la población)	V(09/12/2008)	V(1)	V(Tiempo límite)	V(Caja)		
		NA	I(3343##)	V(09/12/2008)	V(1)	V(Tiempo límite)	V(Caja)		
		NA	V(Crédito a la población)	I(12-02-08)	V(1)	V(Tiempo límite)	V(Caja)		
		NA	V(Crédito a la población)	V(09/12/2008)	I(sw)	V(Tiempo límite)	V(Caja)		
EP 1.4	Cancelar operación.	NA	NA	NA	NA	NA	NA	Se cancela la operación y se cierra la ventana.	

Anexo 30: Eliminar tipo de retención

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 Nomencladores.
- Se debe seleccionar el menú Tipo retención.

Requisitos a probar

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

Descripción de variable

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

Juegos de datos a probar

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

EP 1.1	Eliminar tipo retención correctamente.	Se muestra un mensaje: “El tipo de retención fue eliminado satisfactoriamente.”.	
EP 1.2	Cancelar operación.	Se cancela la operación y se cierra la ventana.	

Anexo 31: Adicionar tipo de programa

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 Nomencladores.
- Se debe seleccionar la opción Programas de retenciones.
- Se debe seleccionar el botón **Adicionar**.

Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Adicionar nomenclador de programas.	Se adiciona un nuevo programa.	EP 1.1: Adicionar un nomenclador de programa introduciendo los datos correctamente.	<p>Se deben introducir los datos del nomenclador correctamente en el formulario.</p> <p>Se debe presionar el botón Aplicar o Aceptar.</p>

		EP 1.2: Adicionar un nomenclador de programa dejando campos requeridos en blanco.	<p>Se deben introducir los datos del nomenclador en el formulario dejando al menos un campo requerido en blanco.</p> <p>Se debe presionar el botón Aplicar o Aceptar.</p>
		EP 1.3: Adicionar un nomenclador de programa introduciendo errores en los datos.	<p>Se deben introducir los datos del nomenclador introduciendo errores en los datos.</p> <p>Se debe presionar el botón Aplicar o Aceptar.</p>
		EP 1.4: Cancelar operación.	<p>Se deben introducir o no los datos a adicionar.</p> <p>Se debe presionar el botón Cancelar.</p>

Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
----	-----------------	---------------	----------------	-------------

1	Denominación.	Campo de texto.	NO	Combinación de letras y números. Debe empezar con una letra.
2	Fecha inicio.	Elemento de fecha.	NO	Selección.

Juegos de datos a probar

Id del escenario	Escenario	Denominación	Fecha inicio	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar un nomenclador de programa introduciendo los datos correctamente.	V(Primera Quincena)	V(09/12/2008)	Se muestra un mensaje: "El programa fue insertado satisfactoriamente."	
EP 1.2	Adicionar un nomenclador de programa dejando campos requeridos en blanco.	I(Vacío)	V(09/12/2008)	Se muestra un mensaje de error: " Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s). "	
		V(Segunda Quincena)	I(Vacío)		
		I(Vacío)	I(Vacío)		
EP 1.3	Adicionar un nomenclador de programa introduciendo errores en los datos.	I(1 Quincena)	I(09-12-2008)	Se muestra un mensaje de error: " Por favor verifique nuevamente que hay campo(s) con valor(es)	
		I(*/-)	I(9/12/2008)		
		I(Segundo*)	I(ahfa)		
		I(prog1*)	I(a09/12/2008)		
		I(*24)	I(09/12/2008*#)		

		I(34 trimestre)	V(12/05/2008)	incorrecto(s). "	
		V(Semanal)	I(12-12-08)		
EP 1.4	Cancelar operación.	NA	NA	Se cancela la operación y se cierra la ventana.	

Anexo 32: Modificar tipo de programa

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 Nomencladores.
- Se debe seleccionar la opción Programas de retenciones.

Requisitos a probar

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

1. Modificar nomenclador de programas.	Se modifica un nomenclador de programas.	EP 1.1: Modificar nomenclador de programas introduciendo los datos correctamente.	<p>Se debe seleccionar el nomenclador de programas a modificar.</p> <p>Se debe presionar el botón Modificar.</p> <p>Se deben introducir los nuevos datos del nomenclador en el formulario.</p> <p>Se debe presionar el botón Aceptar.</p>
		EP 1.2: Modificar nomenclador de programas dejando campos requeridos en blanco.	<p>Se debe seleccionar el nomenclador de programas a modificar.</p> <p>Se debe presionar el botón Modificar.</p> <p>Se deben introducir los datos del nomenclador en el formulario dejando al menos un campo requerido en blanco.</p> <p>Se debe presionar el botón Aceptar.</p>
		EP 1.3: Modificar nomenclador de	<p>Se debe seleccionar el nomenclador de</p>

		programas introduciendo errores en los datos.	<p>programas a modificar.</p> <ul style="list-style-type: none"> - Se debe presionar el botón Modificar. - Se deben introducir los datos del nomenclador en el formulario introduciendo algunos datos inválidos. - Se debe presionar el botón Aceptar.
		EP 1.4: Cancelar operación.	<p>Se debe seleccionar el nomenclador de programas a modificar.</p> <ul style="list-style-type: none"> - Se debe presionar el botón Modificar. - Se deben introducir o no los datos del nomenclador en el formulario. - Se debe presionar el botón Cancelar.

Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
----	-----------------	---------------	----------------	-------------

1	Denominación.	Campo de texto.	NO	Combinación de letras y números. El primer carácter debe ser una letra.
2	Fecha inicio.	Elemento de fecha.	NO	Selección.

Juegos de datos a probar

Id del escenario	Escenario	Denominación	Fecha inicio	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar nomenclador de programas introduciendo los datos correctamente.	V(Segunda Quincena)	V(09/12/2008)	Se muestra un mensaje: "El programa fue actualizado satisfactoriamente."	
EP 1.2	Modificar nomenclador de programas dejando campos requeridos en blanco.	I(Vacío)	V(09/12/2008)	Se muestra un mensaje de error: " Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s). "	
		V(Tercera Quincena)	I(Vacío)		
		I(Vacío)	I(Vacío)		
EP 1.3	Modificar nomenclador de programas introduciendo errores en los datos.	I(1 Quincena)	I(09-12-2008)	Se muestra un mensaje de error: " Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s). "	
		I(*/-)	I(9/12/2008)		
		I(Segundo*)	I(ahfa)		
		I(prog1*)	I(a09/12/2008)		
		I(*24)	I(09/12/2008*#)		
		I(34 trimestre)	V(12/05/2008)		

		V(Semanal)	I(12-12-08)		
EP 1.4	Cancelar operación.	NA	NA	Se cancela la operación y se cierra la ventana.	

Anexo 33: Eliminar tipo de programa

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 Nomencladores.
- Se debe seleccionar la opción Programas de retenciones.

Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Eliminar nomenclador de programas.	Se elimina un nomenclador de programas.	EP 1.1: Eliminar nomenclador de programas correctamente.	<ul style="list-style-type: none"> - Se debe seleccionar el nomenclador 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 nomenclador a eliminar. — Se debe presionar el botón Eliminar. — Se debe presionar el botón Cancelar.
--	--	-----------------------------	--

Descripción de variable

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

Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar nomenclador de programas correctamente.	Se muestra un mensaje: "El programa fue eliminado satisfactoriamente."	
EP 1.2	Cancelar operación.	Se cancela la operación y se cierra la ventana.	

Anexo 34: Adicionar tipo recargo

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 Nomencladores.
- Se debe seleccionar Nomencladores.
- Se debe seleccionar el menú Gestionar tipo de recargo.
- Se debe seleccionar la opción **Adicionar**.

Requisitos a probar

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

		EP 1.3: Adicionar un tipo de recargo introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se deben introducir los datos del tipo de recargo 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 deben introducir o no los datos a adicionar. - Se debe presionar el botón Cancelar.

Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Denominación.	Combo Box.	NO	Selección.
2	Descripción.	Campo de texto.	NO	Permite combinación de números y letras.

Juegos de datos a probar

Id del escenario	Escenario	Denominación	Fecha inicio	Respuesta del sistema	Resultado de la prueba
------------------	-----------	--------------	--------------	-----------------------	------------------------

EP 1.1	Adicionar un tipo de recargo introduciendo los datos correctamente.	V(Atraso por enfermedad)	V(EI trabajador se encuentra enfermo)	Se muestra un mensaje: "El tipo de recargo fue insertado satisfactoriamente."	
EP 1.2	Adicionar un tipo de recargo dejando campos requeridos en blanco.	I(Vacío)	V(EI trabajador se encuentra enfermo)	Se muestra un mensaje de error: " Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s). "	
		V(Cumpliendo misión)	I(Vacío)		
		I(Vacío)	I(Vacío)		
EP 1.3	Adicionar un tipo de recargo introduciendo errores en los datos.	I(3434)	I(33#@\$#)\$)	Se muestra un mensaje de error: " Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s). "	
		V(Cumpliendo misión)	I(@##\$#%)		
		I(@#\$\$@#Ser4454)	V(EI trabajador se encuentra enfermo)		
EP 1.4	Cancelar operación.	NA	NA	Se cancela la operación y se cierra la ventana.	

Anexo 35: Modificar tipo recargo

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 Nomencladores.
- Se debe seleccionar Nomencladores.
- Se debe seleccionar el menú Gestionar tipo de recargo.

Requisitos a probar

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

		<p>EP 1.2: Modificar tipo de recargo dejando campos requeridos en blanco.</p>	<ul style="list-style-type: none"> - Se debe seleccionar el tipo de recargo a modificar. - Se debe presionar el botón Modificar. - Se deben introducir los datos del tipo de recargo en el formulario dejando al menos un campo requerido en blanco. - Se debe presionar el botón Aceptar.
		<p>EP 1.3: Modificar tipo de recargo introduciendo errores en los datos.</p>	<ul style="list-style-type: none"> - Se debe seleccionar el tipo de recargo a modificar. - Se debe presionar el botón Modificar. - Se deben introducir los datos del tipo de recargo 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 seleccionar el tipo de recargo de programas a modificar. - Se debe presionar el botón Modificar. - Se deben introducir o no los datos del tipo de recargo en el formulario. - Se debe presionar el botón Cancelar.
--	--	-----------------------------	---

Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Denominación.	Combo Box.	NO	Selección.
2	Descripción.	Campo de texto.	NO	Permite combinación de números y letras.

Juegos de datos a probar

Id del escenario	Escenario	Denominación	Descripción	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar tipo de recargo introduciendo los datos correctamente.	V(Atraso por enfermedad)	V(El trabajador se encuentra enfermo)	Se muestra un mensaje: "El recargo fue actualizado satisfactoriamente."	

EP 1.2	Modificar tipo de recargo dejando campos requeridos en blanco.	I(Vacío)	V(EI trabajador se encuentra enfermo)	Se muestra un mensaje de error: " Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s). "	
		V(Cumpliendo misión)	I(Vacío)		
		I(Vacío)	I(Vacío)		
EP 1.3	Modificar tipo de recargo introduciendo errores en los datos.	I(3434)	I(33#\$\$@#\$\$)	Se muestra un mensaje de error: " Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s). "	
		V(Cumpliendo	I(@##\$#\$%)		
		I(@#\$\$@#Ser4454)	V(EI trabajador se encuentra enfermo)		
EP 1.4	Cancelar operación.	NA	NA	Se cancela la operación y se cierra la ventana.	

Anexo 36: Eliminar tipo recargo

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 Nomencladores.
- Se debe seleccionar Nomencladores.
- Se debe seleccionar el menú Gestionar tipo de recargo.

Requisitos a probar

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

Descripción de variable

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

Juegos de datos a probar

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

EP 1.1	Eliminar tipo de recargo correctamente.	Se muestra un mensaje: "El tipo de recargo fue eliminado satisfactoriamente."	
EP 1.2	Cancelar operación.	Se cancela la operación y se cierra la ventana.	

Anexo 37: Detalles del submayor de vacaciones

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 Submayores.
- Se debe seleccionar el menú Submayor de vacaciones.

Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Detalles del submayor de vacaciones.	Se muestran los detalles de la cuenta submayor.	EP 1.1: Mostrar detalles de la cuenta submayor.	<ul style="list-style-type: none"> - Se selecciona la cuenta de la cual se quieren ver los detalles. - Se presiona el botón Detalles.

Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción

1	NA	NA	NA	NA
---	----	----	----	----

Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Mostrar detalles de la cuenta submayor.	Se muestra en el grid los datos de la cuenta.	

Anexo 38: Configuración del submayor de vacaciones

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 Submayores.
- Se debe seleccionar el menú Configuración del submayor de vacaciones.
- Se debe seleccionar el botón **Configurar**.

Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Configurar el submayor de vacaciones.	Se actualiza la configuración del submayor de vacaciones.	EP 1.1: Configurar el submayor de vacaciones introduciendo los datos correctamente.	<ul style="list-style-type: none"> - Se deben introducir en el formulario los datos correctamente. - Se debe presionar el botón Aceptar.

		EP 1.2: Configurar el submayor de vacaciones dejando campos requeridos en blanco.	<ul style="list-style-type: none"> Se deben introducir en el formulario los datos dejando campos requeridos en blanco. Se debe presionar el botón Aceptar.
		EP 1.3: Configurar el submayor de vacaciones introduciendo errores en los datos.	<ul style="list-style-type: none"> Se deben introducir en el formulario los datos, introduciendo algunos con errores. Se debe presionar el botón Aceptar.
		EP 1.4: Cancelar operación.	<ul style="list-style-type: none"> Se deben introducir o no en el formulario los datos. Se debe presionar el botón Cancelar.

Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	% Vacaciones.	Campo de texto.	NO	Combinación números.
2	Cuenta.	Lista desplegable.	NO	Selección.

Juegos de datos a probar

Id del escenario	Escenario	% Vacaciones	Cuenta	Respuesta del sistema	Resultado de la prueba
EP 1.1	Configurar el submayor de vacaciones introduciendo los datos correctamente.	V(52.90)	V(Caja)	Se muestra un mensaje: "La configuración del submayor de vacaciones fue actualizada satisfactoriamente."	
EP 1.2	Configurar el submayor de vacaciones dejando campos requeridos en blanco.	I(Vacío)	V(100 Banco)	Se muestra un mensaje de error: "Este campo es requerido."	
		V(52.90)	I(vacío)		
EP 1.3	Configurar el submayor de vacaciones introduciendo errores en los datos.	I(as)	V(444 Materias primas)	Se muestra un mensaje de error.	
		V(52.90)	NA		
EP 1.4	Cancelar operación.	NA	NA	Se cancela la operación y se cierra la ventana.	

Anexo 39: Detalles del submayor de retenciones

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 Submayores.
- Se debe seleccionar el menú Gestionar submayor de retenciones.

Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Mostrar detalles del submayor de retenciones.	Mostrar detalles de un submayor de retenciones.	EP 1.1: Mostrar detalles de un submayor de retenciones correctamente.	<ul style="list-style-type: none"> - Se debe seleccionar el submayor de retenciones del que se quiere ver los detalles. - Se debe presionar el botón Detalles.

Descripción de variable

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

Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Mostrar detalles de un submayor de retenciones correctamente.	Se muestra en un grid los datos del submayor de retenciones en cuestión.	

Anexo 40: Filtrar del submayor de retenciones

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 Fichero histórico devengado.
- Se debe seleccionar el botón **Filtrar**.

Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Resumen devengado del trabajador en meses anteriores	Se muestran una cierta cantidad del salario devengado de un trabajador, dependiendo de la cantidad de meses que se señalen por el especialista.	EP 1.1: Filtrar el resumen devengado del trabajador en meses anteriores por fecha inicio.	Se debe marcar la opción Fecha inicio. Se debe presionar el botón Filtrar .
		EP 1.2: Filtrar el resumen devengado del trabajador en meses anteriores por fecha fin.	Se debe marcar la opción Fecha fin. Se debe presionar el botón Filtrar .
		EP 1.3: Cancelar operación.	Se debe marcar o no los campos que aparecen. Se debe presionar el botón Cancelar .

Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Fecha inicio.	Checkbox	NO	Selección.
2	Fecha fin.	Checkbox	NO	Selección.

Juegos de datos a probar

Id del escenario	Escenario	Fecha inicio	Fecha fin	Respuesta del sistema	Resultado de la prueba
EP 1.1	Filtrar el resumen devengado del trabajador en meses anteriores por fecha inicio.	V(Si)	V(No)	Se muestra en un grid el resultado del filtrado.	
EP 1.2	Filtrar el resumen devengado del trabajador en meses anteriores por fecha fin.	V(No)	V(Si)	Se muestra en un grid el resultado del filtrado.	

EP 1.3	Cancelar operación.	NA	NA	Se cancela la operación y se cierra la ventana.	
--------	---------------------	----	----	---	--

Anexo 41: Pago directo al banco del submayor de retenciones

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 Submayores.
- Se debe seleccionar el menú Gestionar submayor de retenciones.

Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Pagos submayor de retenciones.	Adicionar un importe a un submayor de retenciones.	EP 1.1: Adicionar un importe a un submayor de retenciones introduciendo los datos correctamente.	<ul style="list-style-type: none"> - Se debe seleccionar la opción Pagos. - Se deben introducir los datos que se solicitan. - Se debe presionar el botón Aceptar.

		<p>EP 1.2: Adicionar un importe a un submayor de retenciones dejando campos requeridos en blanco.</p>	<p>Se debe seleccionar la opción Adicionar.</p> <p>Se deben introducir los datos requeridos dejando al menos un campo requerido en blanco.</p> <p>Se debe presionar el botón Aceptar.</p>
		<p>EP 1.3: Adicionar un importe a un submayor de retenciones introduciendo errores en los datos.</p>	<p>Se debe seleccionar la opción Adicionar.</p> <p>Se deben introducir los datos requeridos introduciendo errores en los mismos.</p> <p>Se debe presionar el botón Aceptar.</p>
		<p>EP 1.4: Cancelar operación.</p>	<p>Se debe seleccionar la opción Adicionar.</p> <p>Se deben introducir o no los datos a adicionar.</p> <p>Se debe presionar el botón Cancelar.</p>

Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
----	-----------------	---------------	----------------	-------------

1	Importe.	Campo de texto.	NO	Combinación de números.
---	----------	-----------------	----	-------------------------

Juegos de datos a probar

Id del escenario	Escenario	Importe	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar un importe a un submayor de retenciones introduciendo los datos correctamente.	V(100)	Se muestra un mensaje: "Se ha realizado el pago directo satisfactoriamente al banco."	
EP 1.2	Adicionar un importe a un submayor de retenciones dejando campos requeridos en blanco.	I(Vacío)	Se muestra un mensaje de error.	
EP 1.3	Adicionar un importe a un submayor de retenciones introduciendo errores en los datos.	I(sdfs)	Se muestra un mensaje de error.	
		I(".\$)		
		I(\$%TR4)		

EP 1.4	Cancelar operación.	NA	Se cancela la operación y se cierra la ventana.	
--------	---------------------	----	---	--

Anexo 42: Alerta de cumplido con el pago de la retención de los trabajadores

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 Submayores.
- Se debe seleccionar el menú Gestionar submayor de retenciones.
- Se debe seleccionar el botón **Alerta**.
- Se debe buscar un trabajador.

Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Mostar alerta de cumplido con el pago de la retención de los trabajadores.	Se muestra alerta de cumplido con el pago de la retención de los trabajadores.	EP 1.1: Mostrar alerta de cumplido con el pago de la retención de los trabajadores.	Se debe presionar el botón Alerta .

Descripción de variable

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

Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Mostrar alerta de cumplido con el pago de la retención de los trabajadores.	Se muestra en un grid los datos de las retenciones que han sido pagadas.	

Anexo 43: Adicionar recargo a la retención

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 Submayor de retenciones.
- Se debe seleccionar el menú Gestionar retenciones
- Se debe seleccionar la opción **Adicionar**.

Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Adicionar recargo.	Se adiciona un recargo a una retención.	EP 1.1: Adicionar un recargo a una retención introduciendo los datos correctamente.	<ul style="list-style-type: none"> - Se deben introducir los datos del recargo correctamente en el formulario. - Se debe presionar el botón Aplicar o Aceptar.

		EP 1.2: Adicionar un recargo a una retención dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se deben introducir los datos del recargo en el formulario dejando al menos un campo requerido en blanco. - Se debe presionar el botón Aplicar o Aceptar.
		EP 1.3: Adicionar un recargo a una retención introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se deben introducir los datos del recargo a la retención 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 deben introducir o no los datos a adicionar. - Se debe presionar el botón Cancelar.

Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Fecha.	Campo de fecha.	NO	Selección.

2	Cantidad de plazos.	Campo de texto.	NO	Combinación de números.
3	Id retención.	Campo de texto.	NO	Combinación de números.
4	Id recargo.	Campo de texto.	NO	Combinación de números.
5	Importe.	Campo de texto.	NO	Combinación de números.

Juegos de datos a probar

Id del escenario	Escenario	Fecha	Cantidad de plazos	Id retención	Id recargo	Importe	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar un recargo a una retención introduciendo los datos correctamente.	V(12/04/2009)	V(4)	V(1587)	V(256)	V(200,00)	Se muestra un mensaje: "El recargo fue insertado satisfactoriamente."	
EP 1.2	Adicionar un recargo a una retención 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(5)	I(Vacío)	V(23)	V(30,50)		
		V(13/02/2009)	I(Vacío)	I(Vacío)	V(2)	V(45)		
		V(16/02/2008)	V(8)	V(47)	I(Vacío)	I(Vacío)		
		I(Vacío)	I(Vacío)	I(Vacío)	V(7859)	I(Vacío)		
		V(15/07/2009)	I(Vacío)	V(4)	I(Vacío)	V(86,08)		

EP 1.3	Adicionar un recargo a una retención introduciendo errores en los datos.	I(12-03-2006)	I(dfgf)	I(rrrrrr)	I(df)	I(dfdds)	Se muestra un mensaje de error: “Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s).”.
		I(05/2/09)	V(12)	I(erwerw)	V(43)	V(45,50)	
		V(12/05/2009)	I(12365)	I(saasfa)	V(2)	V(58,00)	
		V(20/01/2004)	V(erew)	V(23)	I(gtrf)	I(sdfsd)	
		I(34)	V(213)	I(ferer)	V(3)	I(sdsds33)	
EP 1.4	Cancelar operación.	NA	NA	NA	NA	NA	Se cancela la operación y se cierra la ventana.

Anexo 44: Modificar recargo a la retención

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 Submayor de retenciones.
- Se debe seleccionar el menú Gestionar de retenciones.

Requisitos a probar

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

<p>1. Modificar recargo.</p>	<p>Se modifica un recargo a una retención.</p>	<p>EP 1.1: Modificar un recargo a una retención introduciendo los datos correctamente.</p>	<ul style="list-style-type: none"> - Se debe seleccionar el recargo a modificar. - Se debe presionar el botón Modificar. - Se deben introducir los nuevos datos del recargo en el formulario. - Se debe presionar el botón Aceptar.
		<p>EP 1.2: Modificar recargo a una retención dejando campos requeridos en blanco.</p>	<ul style="list-style-type: none"> - Se debe seleccionar el recargo a modificar. - Se debe presionar el botón Modificar. - Se deben introducir los datos del recargo en el formulario dejando al menos un campo requerido en blanco. - Se debe presionar el botón Aceptar.

		<p>EP 1.3: Modificar recargo a una retención introduciendo errores en los datos.</p>	<ul style="list-style-type: none"> - Se debe seleccionar el recargo a modificar. - Se debe presionar el botón Modificar. - Se deben introducir los datos del recargo 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 seleccionar el recargo a modificar. - Se debe presionar el botón Modificar. - Se deben introducir o no los datos del recargo en el formulario. - Se debe presionar el botón Cancelar.

Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Fecha.	Campo de fecha.	NO	Selección.
2	Cantidad de plazos.	Campo de texto.	NO	Combinación de números.

3	Id retención.	Campo de texto.	NO	Combinación de números.
4	Id recargo.	Campo de texto.	NO	Combinación de números.
5	Importe.	Campo de texto.	NO	Combinación de números.

Juegos de datos a probar

Id del escenario	Escenario	Fecha	Cantidad de plazos	Id retención	Id recargo	Importe	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar un recargo a una retención introduciendo los datos correctamente.	NA	V(4)	NA	NA	V(200,00)	Se muestra un mensaje: “El recargo fue actualizado satisfactoriamente.”.	
EP 1.2	Modificar recargo a una retención dejando campos requeridos en blanco.	NA	I(Vacío)	NA	NA	I(Vacío)	Se muestra un mensaje de error: “Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s).”.	
		NA	V(5)	NA	NA	V(30,50)		
		NA	I(Vacío)	NA	NA	V(45)		
		NA	V(8)	NA	NA	I(Vacío)		
		NA	I(Vacío)	NA	NA	I(Vacío)		
EP 1.3	Modificar recargo a una retención introduciendo errores en los	NA	I(dfdf)	NA	NA	I(dfdds)	Se muestra un mensaje de error: “Por favor verifique nuevamente que hay campo(s) con	
		NA	V(12)	NA	NA	V(45,50)		
		NA	I(12365)	NA	NA	V(58,00)		

	datos.	NA	V(erev)	NA	NA	I(sdfsd)	valor(es)	
		NA	V(213)	NA	NA	I(sdsds333)	incorrecto(s).”.	
EP 1.4	Cancelar operación.	NA	NA	NA	NA	NA	Se cancela la operación y se cierra la ventana.	

Anexo 45: Eliminar recargo a la retención

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 Submayor de retenciones.
- Se debe seleccionar el menú Gestionar de retenciones.

Requisitos a probar

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

Descripción de variable

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

Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar recargo a una retención.	Se muestra un mensaje: "El recargo fue eliminado satisfactoriamente."	
EP 1.2	Cancelar operación.	Se cancela la operación y se cierra la ventana.	

Anexo 46: Mostrar fichero

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 Fichero histórico devengado.

Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Mostrar fichero.	Mostrar fichero.	EP 1.1: Mostrar fichero correctamente.	<ul style="list-style-type: none"> - Se debe seleccionar la opción Mostrar Fichero. - Se busca al trabajador del cual se quieren ver los datos. - Se muestran los datos en la pantalla.

Descripción de variable

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

Juegos de datos a probar

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Mostrar fichero correctamente.	Se muestra en un grid los datos requeridos.	

Anexo 47: Modificar fichero

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 Fichero histórico devengado.

Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Modificar fichero.	Se modifica un tipo retención.	EP 1.1: Modificar fichero introduciendo los datos correctamente.	<ul style="list-style-type: none"> - Se debe buscar al trabajador - Se debe seleccionar el fichero a modificar. - Se debe presionar el botón Modificar. - Se deben introducir los nuevos datos del fichero en el formulario. - Se debe presionar el botón Aceptar.

		<p>EP 1.2: Modificar fichero dejando campos requeridos en blanco.</p>	<ul style="list-style-type: none"> - Se debe buscar al trabajador. - Se debe seleccionar el fichero a modificar. - Se debe presionar el botón Modificar. - Se deben introducir los datos del fichero en el formulario dejando al menos un campo requerido en blanco. - Se debe presionar el botón Aceptar.
		<p>EP 1.3: Modificar fichero introduciendo errores en los datos.</p>	<ul style="list-style-type: none"> - Se debe buscar al trabajador. - Se debe seleccionar el fichero a modificar. - Se debe presionar el botón Modificar. - Se deben introducir los datos del fichero 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 buscar al trabajador. - Se debe seleccionar el fichero a modificar. - Se debe presionar el botón Modificar. - Se deben introducir o no los datos del fichero en el formulario. - Se debe presionar el botón Cancelar.
--	--	-----------------------------	---

Descripción de variable

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Expediente laboral.	Campo de texto	NO	Combinación de números.
2	Nombre.	Campo de texto.	NO	Combinación de letras.
3	Primer apellido.	Campo de texto.	NO	Combinación de letras.
4	Segundo apellido.	Campo de texto.	NO	Combinación de letras.
5	Importe por mes.	Campo de texto	NO	Combinación de números.

Juegos de datos a probar

Id del escenario	Escenario	Expediente laboral	Nombre	Primer apellido	Segundo apellido	Importe por mes	Respuesta del sistema	Resultado de la prueba
------------------	-----------	--------------------	--------	-----------------	------------------	-----------------	-----------------------	------------------------

EP 1.1	Modificar fichero introduciendo los datos correctamente.	NA	NA	NA	NA	V(Tiempo límite)	Se muestra un mensaje: "El fichero fue actualizado satisfactoriamente."	
EP 1.2	Modificar tipo retención dejando campos requeridos en blanco.	NA	NA	NA	NA	I(Vacío)	Se muestra un mensaje de error: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)."	
EP 1.3	Modificar tipo retención introduciendo errores en los datos.	NA	NA	NA	NA	I(adsds)	Se muestra un mensaje de error: "Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)."	
		NA	NA	NA	NA	I(\$%^*)		
		NA	NA	NA	NA	V(2000,0 0) límite)		
EP 1.4	Cancelar operación.	NA	NA	NA	NA	NA	Se cancela la operación y se cierra la ventana.	

Anexo 48: Tabla de seguimiento de los requisitos

Agrupación	Requisitos	Primera Fase	Segunda Fase	Observaciones de Negocio	Observaciones de Formato	Observaciones de Consistencia
Gestionar nomenclador tipos de retenciones.	Adicionar tipos de retenciones.	X				
	Modificar tipos de retenciones.	X				
	Eliminar tipos de retenciones.	X				
Submayor de vacaciones.	Apertura del submayor.	X				
	Cierre del submayor.	X				
	Actualizar el submayor	X				
	Detalles del submayor.	X				X
	Conciliación con contabilidad del submayor.	X				X
	Dar alta automática a un trabajador.		X			
	Generar comprobante de operaciones dado una modificación.		X			
	Ajuste del submayor.	X		X		
Submayor de retenciones.	Adicionar retención del trabajador.	X				
	Modificar retención del trabajador.	X				
	Eliminar retención del trabajador.	X				
	Detalles del submayor.	X		X		
	Filtrar del submayor.	X				X
	Actualizar submayor.	X				
	Pago directo al banco del submayor.	X		X		X
	Alerta de cumplido con el pago de la retención de los trabajadores.		X			
	Descuento adelantado de las retenciones.		X			
	Conciliación con	X				X

	contabilidad del submayor.					
Gestionar nomenclador de programas.	Adicionar programa.	X				X
	Modificar programa.	X				
	Eliminar programa.	X				
Gestionar nomenclador de recargo.	Adicionar recargo.		X			
	Modificar recargo.		X			
	Eliminar recargo.		X			
Configuración del submayor de vacaciones.	Configuración del submayor	X				
Gestionar recargo a la retención.	Adicionar recargo a la retención.		X			
	Modificar recargo a la retención.		X			
	Eliminar recargo a la retención.		X			
Fichero histórico devengado.	Mostrar fichero.		X			
	Modificar fichero.		X			
	Alta automática del fichero.		X			
	Resumen devengado del trabajador en meses anteriores.		X			
Total		21	14	3	0	6

Descripción de las clases del diseño

Anexo 49: Gestionar submayor de vacaciones

Nombre: Gestionar submayor de vacaciones Controller	
Tipo de clase: Controller	
Para cada responsabilidad	
Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	gestionarsubmayorvacacionesAction()
Descripción:	Constructor de la clase.
Nombre:	cargarbusAction()
Descripción:	Se carga la ventana del buscar Trabajador.
Nombre:	cargareliminarsubmayorvacacionesAction()
Descripción:	Carga la interfaz correspondiente a esta operación.
Nombre:	cargarinsertarsubmayorvacacionesAction()
Descripción:	Carga la interfaz correspondiente a esta operación.
Nombre:	getsaldocuenta(idcuenta)
Descripción:	Dado el identificador de una cuenta de un submayor, devuelve el saldo de la misma.
Nombre:	getsaldocuentacontabilidad(idcuenta)
Descripción:	Dado el identificador de una cuenta en contabilidad, devuelve el saldo de la misma.
Nombre:	ajustesubvacacionesAction()
Descripción:	Se modifica el submayor de vacaciones en cuestión.

Nombre: Dat submayor de vacaciones Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	DatSubmayorvacModel()
Descripción:	Constructor de la clase.
Nombre:	Insertar(DatSubmayorvac : DatSubmayorvac)
Descripción:	Se adiciona un nuevo submayor de vacaciones, pasándole un objeto de este tipo.
Nombre:	Actualizar(DatSubmayorvac : DatSubmayorvac)
Descripción:	Se actualizan los datos del submayor que se pasa como parámetro.
Nombre:	Eliminar(DatSubmayorvac)
Descripción:	Se elimina el submayor de vacaciones pasado como parámetro.
Nombre:	BuscarSubmayorVacacionesById(idsubmayorvacaciones)
Descripción:	Devuelve un submayor de vacaciones dado su identificador.
Nombre:	Verificarexistenciatrabajador()
Descripción:	Verifica si un trabajador se encuentra o no en la base de datos.

Nombre:	Generarcomprobantedeoperaciones()
Descripción:	Cuando se realiza una modificación al submayor de vacaciones y se actualizan los datos del mismo, este método se encarga de enviar a nómina los siguientes datos: cuenta objeto, cuenta contrapartida, importe.
Nombre:	Gettrabvacaciones()
Descripción:	Este método proporciona a nomina la cantidad de vacaciones acumuladas de un trabajador.

Nombre: Dat submayor de vacaciones	
Tipo de clase: Model(Domain)	
Para cada responsabilidad	
Nombre:	setUp()
Descripción:	En este método establecen las relaciones entre las diferentes tablas en la base de datos con la cual se relaciona esta clase.
Nombre:	GetLLave()
Descripción:	Genera identificadores. Busca en la base de datos el último identificador y retorna este número más 1.
Nombre:	getCantSubmayores_Vacaciones(idestructuracomun)
Descripción:	Devuelve la cantidad de Submayores de vacaciones en la base de datos.
Nombre:	Buscar(idsubmayorvac)
Descripción:	Dado el identificador del submayor de vacaciones, lo busca y lo devuelve.
Nombre:	GetTodos()
Descripción:	Devuelve todos los Submayores de vacaciones que existen en la base de datos.
Nombre:	GetPorLimite(limite = 10, inicio = 0, idestructuracomun)
Descripción:	Dado el identificador de una estructura, devuelve de 10 en 10 los Submayores de vacaciones.
Nombre:	VerificarTrabajador(idtrabajador)
Descripción:	Dado el identificador de un trabajador, verifica si existe o no.

Anexo 50: Gestionar submayor de retenciones

Nombre: Gestionar submayor de retenciones Controller	
Tipo de clase: Controller	
Para cada responsabilidad	
Nombre:	init()
Descripción:	Se inicializan los parámetros.

Nombre:	gestionarsubmayorretencionesAction()
Descripción:	Constructor de la clase.
Nombre:	buscarAction()
Descripción:	Se carga la interfaz correspondiente a esta operación.
Nombre:	cargarmodificarAction()
Descripción:	Se carga la interfaz y los datos correspondientes a esta operación.
Nombre:	cargareliminarAction()
Descripción:	Se carga la interfaz correspondiente a esta operación.
Nombre:	cargarinsertarAction()
Descripción:	Se carga la interfaz correspondiente a esta operación.
Nombre:	descuentoporadelantadoAction()
Descripción:	Se carga la interfaz correspondiente a esta operación.
Nombre:	detallesubretAction()
Descripción:	Muestra todos los datos del submayor de retenciones en cuestión.
Nombre:	filtrarsubretAction()
Descripción:	Se carga la interfaz correspondiente a esta operación.
Nombre:	cargarinsertartiporecAction()
Descripción:	Se carga la interfaz correspondiente a esta operación.
Nombre:	cargarmodificartiporecAction()
Descripción:	Se carga la interfaz y los datos correspondientes a esta operación.
Nombre:	cargareliminartiporecAction()
Descripción:	Se carga la interfaz correspondiente a esta operación.
Nombre:	pagodirectobanAction()
Descripción:	Se carga la interfaz correspondiente a esta operación.

Nombre: Dat submayor de retenciones Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	DatSubmayorretencionesModel()
Descripción:	Constructor de la clase.
Nombre:	Insertar(DatSubmayorretencione : DatSubmayorretencione)
Descripción:	Se adicionan los datos del submayor de retenciones.
Nombre:	Actualizar(DatSubmayorretencione : DatSubmayorretencione)

Descripción:	Son modificados los datos del submayor de retenciones que es pasado como parámetro.
Nombre:	Eliminar(DatSubmayorretencione : DatSubmayorretencione)
Descripción:	Se elimina el submayor de retenciones que es pasado como parámetro.
Nombre:	BuscarsubmayorretencionesByld(idsubmayor)
Descripción:	Devuelve el submayor de retenciones dado su identificador.
Nombre:	Filtrarsubret(selección)
Descripción:	Hace una selección de los datos en dependencia del atributo que se le pase como parámetro.

Nombre: Dat subamajor retenciones	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	setUp()
Descripción:	En este método establecen las relaciones entre las diferentes tablas en la base de datos con la cual se relaciona esta clase.
Nombre:	GetLLave()
Descripción:	Genera identificadores. Busca en la base de datos el último identificador y retorna este número más 1.
Nombre:	Buscar(idsubmayorretenciones)
Descripción:	Dado el identificador del submayor de retenciones, lo busca y lo devuelve.
Nombre:	GetTodos()
Descripción:	Devuelve todos los Submayores de retenciones que existen en la base de datos.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve de 10 en 10, los Submayores de retenciones con sus respectivos parámetros.

Nombre: Dat Registro de retenciones Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	DatRegistroretencioneModel()
Descripción:	Constructor de la clase.
Nombre:	Insertar(DatRegistroretencione : DatRegistroretencione)
Descripción:	Se adiciona un nuevo registro de retenciones, pasándole un objeto de este tipo.
Nombre:	Actualizar(DatRegistroretencione : DatRegistroretencione)
Descripción:	Se actualizan los datos del registro de retenciones que se pasa como parámetro.
Nombre:	Eliminar(DatRegistroretencione)

Descripción:	Se elimina el registro de retenciones pasado como parámetro.
Nombre:	BuscarregistroretencionesById(idregistroretenciones)
Descripción:	Devuelve el registro de retenciones dado su identificador.

Nombre: Dat Registro de retenciones	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	setUp()
Descripción:	En este método establecen las relaciones entre las diferentes tablas en la base de datos con la cual se relaciona esta clase.
Nombre:	GetLLave()
Descripción:	Genera identificadores. Busca en la base de datos el último identificador y retorna este número más 1.
Nombre:	Buscar(idregistroretenciones)
Descripción:	Dado el identificador del registro de retenciones, lo busca y lo devuelve.
Nombre:	GetTodos()
Descripción:	Devuelve todos los registros de retenciones que existen en la base de datos.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve de 10 en 10, los registros de retenciones con sus respectivos parámetros que tengan el atributo hasta, vacío.
Nombre:	GetPorLim(limite = 10, inicio = 0, idestructuracomun)
Descripción:	Devuelve de 10 en 10, los registros de retenciones donde el identificador de estructura sea el mismo.
Nombre:	GetUltimold()
Descripción:	Devuelve el último identificador que este registrado en la base de datos.

Anexo 51: Gestionar concepto de retención

Nombre: Gestionar concepto retenciones Controller	
Tipo de clase: Controller	
Para cada responsabilidad	
Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	gestionarconceptoretencionesAction()
Descripción:	Constructor de la clase.
Nombre:	cargarmodificarconceptoretencionesAction()
Descripción:	Carga la interfaz y los datos correspondientes a esta operación.
Nombre:	cargarinsertarconceptoretencionAction()
Descripción:	Carga la interfaz correspondiente a esta operación.

Nombre:	cargareliminarconceptoAction()
Descripción:	Carga la interfaz correspondiente a esta operación.

Nombre: Dat Concepto retenciones Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	DatConceptoretencioneModel()
Descripción:	Constructor de la clase.
Nombre:	Insertar(DatConceptoretencione : DatConceptoretencione)
Descripción:	Se adiciona un nuevo concepto de retención, pasándole un objeto de este tipo.
Nombre:	Actualizar(DatConceptoretencione : DatConceptoretencione)
Descripción:	Se actualizan los datos del concepto de retención que se pasa como parámetro.
Nombre:	Eliminar(DatConceptoretencione)
Descripción:	Se elimina el concepto de retención pasado como parámetro.
Nombre:	BuscarConceptoRetencione(idconceptoretención)
Descripción:	Devuelve el concepto de retención dado su identificador.
Nombre:	Existeprograma(idprograma)
Descripción:	Verifica si un programa existe o no en la base de datos.

Nombre: Dat Concepto retenciones	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	setUp()
Descripción:	En este método establecen las relaciones entre las diferentes tablas en la base de datos con la cual se relaciona esta clase.
Nombre:	GetLLave()
Descripción:	Genera identificadores. Busca en la base de datos el último identificador y retorna este número más 1.
Nombre:	Buscar(idconceptoretenciones)
Descripción:	Dado el identificador del concepto de retención, lo busca y lo devuelve.
Nombre:	GetTodos()
Descripción:	Devuelve todos los conceptos de retenciones que existen en la base de datos.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve de 10 en 10, los conceptos de retenciones con sus atributos, que tengan el parámetro hasta vacío.

Anexo 52: Gestionar operaciones

Nombre: Gestionar operación Controller	
Tipo de clase: Controller	
Para cada responsabilidad	
Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	gestionaroperacionAction()
Descripción:	Constructor de la clase.
Nombre:	cargarinsertaroperacionAction()
Descripción:	Se carga la interfaz correspondiente a esta operación.
Nombre:	cargarModificaroperacionAction()
Descripción:	Se carga la interfaz correspondiente y lo datos necesarios para realizar esta operación.
Nombre:	cargareliminaroperacionAction()
Descripción:	Se carga la interfaz correspondiente a esta operación.

Nombre: Nom Operación sub Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	NomOperacionsubModel()
Descripción:	Constructor de la clase.
Nombre:	Insertar(NomOperacionsub : NomOperacionsub)
Descripción:	Se adiciona una nueva operación, pasándole un objeto de este tipo.
Nombre:	Actualizar(NomOperacionsub : NomOperacionsub)
Descripción:	Se actualizan los datos de la operación que se pasa como parámetro.
Nombre:	Eliminar(NomOperacionsub)
Descripción:	Se elimina la operación que se pasa como parámetro.
Nombre:	BuscaroperacionByld(idoperacion)
Descripción:	Devuelve la operación dado su identificador.

Nombre: Nom Operaciones sub	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	

Nombre:	setUp()
Descripción:	En este método establecen las relaciones entre las diferentes tablas en la base de datos con la cual se relaciona esta clase.
Nombre:	GetLLave()
Descripción:	Genera identificadores. Busca en la base de datos el último identificador y retorna este número más 1.
Nombre:	Buscar(idoperacion)
Descripción:	Dado el identificador de la operación, lo busca y lo devuelve.
Nombre:	GetTodos()
Descripción:	Devuelve todas las operaciones que existen en la base de datos.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve de 10 en 10, las operaciones.

Anexo 53: Gestionar tipo de retenciones

Nombre: Gestionar tipo retención Controller	
Tipo de clase: Controller	
Para cada responsabilidad	
Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	gestionartiporetencionAction()
Descripción:	Constructor de la clase.
Nombre:	cargarmodificartiporetencionAction()
Descripción:	Carga la interfaz y los datos correspondientes a esta operación.
Nombre:	cargarinsertartiporetencionAction()
Descripción:	Carga la interfaz correspondiente a esta operación.
Nombre:	cargareliminartiporetencionAction()
Descripción:	Carga la interfaz correspondiente a esta operación.

Nombre: Nom Tipo retención Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	NomTiporetencionModel()
Descripción:	Constructor de la clase.
Nombre:	Insertar(NomTiporetencion : NomTiporetencion)
Descripción:	Se adiciona un nuevo tipo de retención, pasándole un objeto de este tipo.
Nombre:	Actualizar(NomTiporetencion : NomTiporetencion)
Descripción:	Se actualizan los datos del tipo de retención que se pasa como parámetro.

Nombre:	Eliminar(NomTiporetencion)
Descripción:	Se elimina el tipo de retención pasado como parámetro.
Nombre:	BuscarotiporetencionById(idtiporetencion)
Descripción:	Devuelve el tipo de retención dado su identificador.

Nombre: Nom Tipo retención	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	setUp()
Descripción:	En este método establecen las relaciones entre las diferentes tablas en la base de datos con la cual se relaciona esta clase.
Nombre:	GetLLave()
Descripción:	Genera identificadores. Busca en la base de datos el último identificador y retorna este número más 1.
Nombre:	Buscar(idtiporetencion)
Descripción:	Dado el identificador del tipo de retención, lo busca y lo devuelve.
Nombre:	GetTodos()
Descripción:	Devuelve todas los tipos de retenciones que existen en la base de datos.
Nombre:	GetTodosPorTipoRetencion(tiporetencion)
Descripción:	Devuelve todas los tipos de retenciones que existen en la base de datos que correspondan con el tipo de retención que le pasan por parámetro.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve de 10 en 10, los tipos de retención con sus respectivos parámetros que tengan el atributo hasta, vacío.

Anexo 54: Gestionar nomenclador de programas

Nombre: Gestionar programa Controller	
Tipo de clase: Controller	
Para cada responsabilidad	
Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	gestionaprogramaAction()
Descripción:	Constructor de la clase.
Nombre:	dameidestructuraAction()

Descripción:	Devuelve el identificador de estructura correspondiente.
Nombre:	cargarmodificarprogramaAction()
Descripción:	Carga la interfaz y los datos correspondientes a esta operación.
Nombre:	cargarinsertarprogramaAction()
Descripción:	Carga la interfaz correspondiente a esta operación.
Nombre:	cargareliminarprogramaAction()
Descripción:	Carga la interfaz correspondiente a esta operación.

Nombre: Nom Programa Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	NomProgramaModel()
Descripción:	Constructor de la clase.
Nombre:	Insertar(NomPrograma : NomPrograma)
Descripción:	Se adiciona un nuevo programa, pasándole un objeto de este tipo.
Nombre:	Actualizar(NomPrograma : NomPrograma)
Descripción:	Se actualizan los datos del programa que se pasa como parámetro.
Nombre:	Eliminar(NomPrograma:NomPrograma)
Descripción:	Se elimina el programa pasado como parámetro.
Nombre:	BuscarProgramaById(idprograma)
Descripción:	Devuelve el programa dado su identificador.

Nombre: Nom Programa	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	setUp()
Descripción:	En este método establecen las relaciones entre las diferentes tablas en la base de datos con la cual se relaciona esta clase.
Nombre:	GetLLave()
Descripción:	Genera identificadores. Busca en la base de datos el último identificador y retorna este número más 1.
Nombre:	Buscar(idprograma)
Descripción:	Dado el identificador del programa, lo busca y lo devuelve.
Nombre:	GetTodos(idestructuracomun)
Descripción:	Devuelve todas los tipos de retenciones que existen en la base de datos.

Nombre:	GetPorLimite(limite, inicio, idestructuracomun)
Descripción:	Devuelve de 10 en 10, los tipos de retención con sus respectivos parámetros que tengan el atributo hasta, vacío.

Anexo 55: Configuración del submayor de vacaciones

Nombre: Gestionar configuración submayor de vacaciones Controller	
Tipo de clase: Controller	
Para cada responsabilidad	
Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	gestionarconfiguracionsubmayorvacAction()
Descripción:	Constructor de la clase.
Nombre:	cargargridConfiguracionsubmayorvacAction()
Descripción:	Se carga la ventana de la operación Configuración del submayor de vacaciones.
Nombre:	cargarModificarConfiguracionsubmayorvacAction()
Descripción:	Carga la interfaz y los datos correspondientes a esta operación.

Nombre: Nom Configuración submayor vacaciones Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	NomConfiguracionsubmayorvacModel()
Descripción:	Constructor de la clase.
Nombre:	Insertar(NomConfiguracionsubmayorvac : NomConfiguracionsubmayorvac)
Descripción:	Se adiciona una nuevo programa, pasándole un objeto de este tipo.
Nombre:	Actualizar(NomConfiguracionsubmayorvac : NomConfiguracionsubmayorvac)
Descripción:	Se actualizan los datos del programa que se pasa como parámetro.
Nombre:	Eliminar(NomPrograma:NomPrograma)
Descripción:	Se elimina el programa pasado como parámetro.
Nombre:	BuscarProgramaByld(idprograma)
Descripción:	Devuelve el programa dado su identificador.

Nombre: Nom Configuración submayor vacaciones	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	setUp()

Descripción:	En este método establecen las relaciones entre las diferentes tablas en la base de datos con la cual se relaciona esta clase.
Nombre:	GetLLave()
Descripción:	Genera identificadores. Busca en la base de datos el último identificador y retorna este número más 1.
Nombre:	Buscar(idconfiguracionsubmayorvac)
Descripción:	Dado el identificador de la configuración del submayor de vacaciones, lo busca y lo devuelve.
Nombre:	GetTodos()
Descripción:	Devuelve todas las configuraciones que existen en la base de datos.
Nombre:	GetPorLimite(limite = 5, inicio = 0, idestructuracomun)
Descripción:	Devuelve de 5 en 5.

Anexo 56: Fichero histórico devengado

Nombre: Gestionar fichero histórico devengado Controller	
Tipo de clase: Controller	
Para cada responsabilidad	
Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	gestionarficherohistoricodevengadoAction()
Descripción:	Constructor de la clase.
Nombre:	cargar mostrar fichero historicodevengadoAction()
Descripción:	Se muestran los datos del fichero histórico devengado del trabajador seleccionado.
Nombre:	cargar modificar fichero historicodevengadoAction()
Descripción:	Se carga la interfaz correspondiente y los datos necesarios para realizar esta operación.
Nombre:	cargar alta automática fichero historicodevengadoAction()
Descripción:	Muestra en los datos del fichero histórico devengado del trabajador que ha sido incorporado a la nómina de la empresa.
Nombre:	cargar resumen devengadoAction()
Descripción:	Muestra los datos del resumen devengado del trabajador seleccionado.

Nombre: Dat Fichero histórico devengado Model	
Tipo de clase: Model (Business)	
Para cada responsabilidad	
Nombre:	DatFicherohistoricodevengadoModel()
Descripción:	Constructor de la clase.

Nombre:	Modificarficherohistoricodevengado()
Descripción:	Se modifica los datos del fichero histórico devengado.
Nombre:	Filtrarresumendeavengado(seleccion)
Descripción:	Hace una selección de los datos en dependencia del atributo que se le pase como parámetro.
Nombre:	BuscarotiporecargobyId(idfichero)
Descripción:	Devuelve el fichero histórico devengado dado su identificador.

Nombre: Dat Fichero histórico devengado	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	setUp()
Descripción:	En este método establecen las relaciones entre las diferentes tablas en la base de datos con la cual se relaciona esta clase.
Nombre:	GetLLave()
Descripción:	Genera identificadores. Busca en la base de datos el último identificador y retorna este número más 1.
Nombre:	Buscar(idficherohistoricodevengado)
Descripción:	Dado el identificador del fichero histórico devengado, lo busca y lo devuelve.
Nombre:	GetTodos()
Descripción:	Devuelve todos los ficheros históricos devengados que existen en la base de datos.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve de 10 en 10, los ficheros históricos devengados con sus respectivos parámetros.
Nombre:	GetFicherohis(idfichero)
Descripción:	Dado el identificador del fichero histórico devengado devuelve todos los datos del mismo.

Anexo 57: Gestionar tipo de recargo

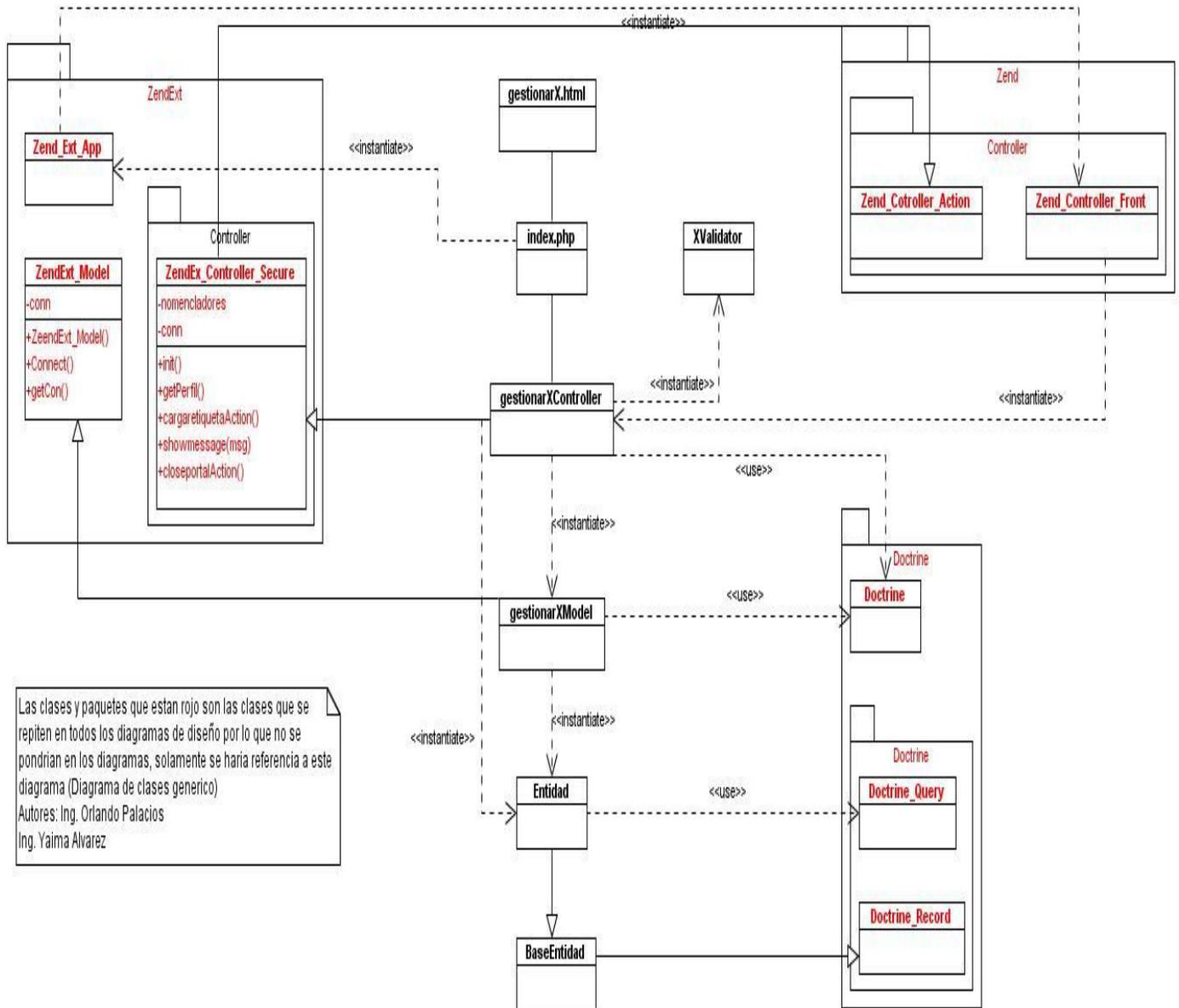
Nombre: Gestionar tipo recargo Controller	
Tipo de clase: Controller	
Para cada responsabilidad	
Nombre:	init()
Descripción:	Se inicializan los parámetros.
Nombre:	gestionartiporecargotipoAction()
Descripción:	Constructor de la clase.
Nombre:	cargarmodificartiporecargotipoAction()

Descripción:	Carga la interfaz y los datos correspondientes a esta operación.
Nombre:	cargarinsertartiporecargaAction()
Descripción:	Carga la interfaz correspondiente a esta operación.
Nombre:	cargareliminartiporecargaAction()
Descripción:	Carga la interfaz correspondiente a esta operación.

Nombre: Nom Tipo recargo Model	
Tipo de clase: Model (Bussines)	
Para cada responsabilidad	
Nombre:	NomTiporecargaModel()
Descripción:	Constructor de la clase.
Nombre:	Insertar(NomTiporecarga : NomTiporecarga)
Descripción:	Se adiciona un nuevo tipo de recargo, pasándole un objeto de este tipo.
Nombre:	Actualizar(NomTiporecarga : NomTiporecarga)
Descripción:	Se actualizan los datos del tipo de recargo que se pasa como parámetro.
Nombre:	Eliminar(NomTiporecarga)
Descripción:	Se elimina el tipo de recargo pasado como parámetro.
Nombre:	BuscarotiporecargaByld(idtiporecarga)
Descripción:	Devuelve el tipo de recargo dado su identificador.

Nombre: Nom Tipo recargo	
Tipo de clase: Model (Domain)	
Para cada responsabilidad	
Nombre:	setUp()
Descripción:	En este método establecen las relaciones entre las diferentes tablas en la base de datos con la cual se relaciona esta clase.
Nombre:	GetLLave()
Descripción:	Genera identificadores. Busca en la base de datos el último identificador y retorna este número más 1.
Nombre:	Buscar(idtiporecarga)
Descripción:	Dado el identificador del tipo de recargo, lo busca y lo devuelve.
Nombre:	GetTodos()
Descripción:	Devuelve todas los tipos de recargos que existen en la base de datos.
Nombre:	GetPorLimite(limite = 10, inicio = 0)
Descripción:	Devuelve de 10 en 10, los tipos de recargo con sus respectivos parámetros.

Diagrama de clases genérico



Las clases y paquetes que estan rojo son las clases que se repiten en todos los diagramas de diseño por lo que no se pondrian en los diagramas, solamente se haria referencia a este diagrama (Diagrama de clases generico)
 Autores: Ing. Orlando Palacios
 Ing. Yaima Alvarez

GLOSARIO

Proceso: Conjunto de actividades enlazadas entre sí 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.

No. carnet: Número de carnet de identidad.