



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS.
FACULTAD 7.**

Trabajo de Diploma para Optar por el Título de
Ingeniero en Ciencias Informáticas.

Título: Módulo para el control del pago a los colaboradores
de la salud v2.0.

Autores:

Arlenys Mariño Valdés
Alejandro Alfonso Hernández

Tutores:

Ing. Yisel Reyes Cardoso
Ing. Rolando Pompa González

Ciudad de La Habana, junio de 2010
“Año 52 de la Revolución”

“Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consientes de recibir el premio en la satisfacción del deber cumplido...”

Ernesto Che Guevara.

Declaración de Autoría.

Declaramos que somos los únicos autores del presente trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 25 días del mes de junio del año 2010.

Arlenys Mariño Valdés
Autora

Alejandro Alfonso Hernández
Autor

Ing. Yisel Reyes Cardoso
Tutora

Ing. Rolando Pompa González
Tutor

Datos de Contacto.

Síntesis del tutor: Ing. Yisel Reyes Cardoso: Ingeniera en Ciencias Informáticas de la UCI. Posee Categoría Docente de Instructor recién graduado. Ha participado en proyectos de desarrollo de sistemas informáticos para la salud desde el año 2006. Imparte la asignatura Matemática Discreta y Algebra Lineal. Actualmente pertenece al Departamento de Sistemas de Apoyo a la Salud del Centro de Informática Médica (CESIM).

Correo: ycardoso@uci.cu

Síntesis del tutor: Ing. Rolando Pompa González: Obtuvo el Título de Ingeniero en Ciencias Informáticas en la tercera graduación, curso 2008-2009, de la Universidad de las Ciencias Informáticas (UCI). Es miembro del Centro de Informática Médica (CESIM) en el Departamento Sistema de Apoyo a la Salud como desarrollador y responsable de tres módulos en desarrollo.

Correo: rpompa@uci.cu

Resumen.

El presente trabajo tiene como objetivo mejorar el Módulo de Pago del Sistema de Gestión para la Colaboración Médica de la versión 1.0 en cuanto a la trazabilidad y la auditoría. Este sistema brinda la posibilidad de conocer la información referente al pago a los colaboradores de la salud y sus designados. Facilita al Departamento de Economía de la Unidad Central de Cooperación Médica la información para acreditar el saldo correcto en sus cuentas a cada colaborador.

Para el desarrollo de esta aplicación fueron utilizadas varias herramientas y tecnologías en software libre, como son: el sistema gestor de base de datos PostgreSQL v8.3, la librería ExtJS v 3.0.1, el servidor Web Apache v2.2, el framework Symfony v1.4.3, el lenguaje de programación PHP v5.2.6 y las herramientas Enterprise Architect v7.1 y NetBeans v6.8.

Con la instalación y puesta en práctica del Módulo de Pago del Sistema para la Colaboración Médica en su versión 2.0, se facilitará la informatización de proceso del control del pago de los colaboradores de la salud, mejorando la trazabilidad y la auditoría con respecto a la versión 1.0. Los datos que brinda la nueva versión del Módulo de Pago son de gran importancia en la toma de decisiones y el manejo de recursos por parte de la UCCM.

El Módulo de Pago del Sistema para la Colaboración Médica en su versión 2.0, fue presentado en la Jornada Científica Estudiantil UCI y en el Fórum de Ciencia y Técnica UCI alcanzando premios de destacado y relevante respectivamente.

Tabla de Contenidos.

Declaración de Autoría.	3
Datos de Contacto.	IV
Resumen.....	V
Tabla de Contenidos.....	VI
Introducción.	1
Capítulo 1. Fundamentación Teórica.....	6
1.1 Conceptos fundamentales.....	6
1.1.1 Ministerio de Salud Pública (MINSAP).....	6
1.1.2 Sistema Nacional de Salud (SNS).	6
1.1.3 Informática para la Salud (INFOMED).....	6
1.1.4 Ministerio de Inversión y Colaboración Extranjera (MINVEC).....	7
1.1.5 Cooperación.	7
1.1.6 Unidad Central de Cooperación Médica (UCCM).	7
1.2 Navegadores.....	7
1.3 Estado del arte.	8
1.3.1 Nivel nacional.	9
1.3.2 Universidad de las Ciencias Informáticas (UCI).	10
1.4 Antecedentes del sistema.	10
1.5 Gestión de proyecto.	11
1.5.1 Modelo de Capacidad y Madurez (CMMI, por sus siglas en inglés).	11
1.5.2 Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés).....	12
1.5.3 Lenguaje Unificado de Modelado (UML, por sus siglas en inglés).....	13
1.6 Tecnologías y lenguajes utilizados.....	13
1.6.1 Servicios Web.	14
1.6.2 Servidor Web Apache.	14
1.6.3 Sistemas Gestores de Base de Datos (SGBD) PostgreSQL.	14
1.6.4 Ajax.	14
1.6.5 Framework Symfony.	14
1.6.6 ExtJS.....	15
1.6.7 Lenguaje de programación Web Hypertext Pre-processor (PHP).....	15

TABLA DE CONTENIDOS.

1.6.8	JavaScript.....	15
1.6.9	Herramienta JMeter.....	15
1.7	Arquitectura del sistema.....	16
1.7.1	Modelo Vista Controlador (MVC).....	16
1.7.2	Arquitectura Cliente-Servidor.....	16
1.8	Herramientas.....	16
1.8.1	Enterprise Architect 7.1.....	16
1.8.2	NetBeans.....	17
Capítulo 2. Características del Sistema.....		18
2.1	Reglas del negocio a considerar.....	18
2.2	Propuesta del sistema.....	19
2.3	Modelo del negocio.....	19
2.4	Modelo de casos de uso del negocio.....	20
2.5	Modelo de objetos del negocio.....	24
2.5.1	Diagrama de actividades del Caso de Uso del Negocio (DCUN).....	24
2.6	Modelo de sistema.....	24
2.6.1	Captura de requisitos.....	24
2.6.2	Concepción del sistema.....	28
2.6.3	Actores del sistema.....	28
2.6.4	Diagrama de Casos de Uso del Sistema (DCUS).....	29
2.6.5	Descripción extendida de los CUS.....	30
Capítulo 3. Análisis y Diseño del Sistema.....		35
3.1	Descripción de la arquitectura. Fundamentación.....	35
3.2	Análisis de posibles implementaciones, componentes o módulos ya existentes que puedan ser rehusados. Estrategias de integración.....	36
3.2.1	Componentes reutilizados.....	36
3.2.2	Estrategia de integración.....	36
3.3	Análisis.....	37
3.3.1	Modelo de Análisis.....	37
3.3.2	Clases de análisis.....	37
3.3.3	Diagrama de clase del análisis.....	37

TABLA DE CONTENIDOS.

3.4	Diseño.....	40
3.4.1	Modelo de diseño.....	40
3.4.2	Diagrama de clases de diseño del sistema.....	41
3.4.3	Diagrama de interacción.....	45
3.4.4	Diagrama de secuencia.....	45
3.4.5	Descripción de las clases del diseño.....	49
Capítulo 4.	Implementación y Prueba.....	54
4.1	Modelo de datos.....	54
4.2	Implementación.....	54
4.3	Tratamiento de errores.....	56
4.4	Seguridad.....	56
4.5	Estrategias de codificación. Estándares y estilos a utilizar.....	56
4.6	Tiempo de respuesta de la solución implementada.....	56
Conclusiones.	58
Recomendaciones.	59
Referencias Bibliográficas.	60
Bibliografía.	64
Glosario de Términos.....	68
Anexos.....	69
Anexo 1. Diagrama de actividades.....	69
Diagrama de actividades del caso de uso del negocio “Realizar misión”.....	69
Diagrama de actividades del caso de uso del negocio “Cobrar salario”.....	70
Diagrama de actividades del caso de uso del negocio “Cobrar salario de ayuda familiar”.....	71

Introducción.

La informática, se ha convertido en un eslabón fundamental del desarrollo económico, político y social. Se puede afirmar que es aquella ciencia que abarca el estudio y aplicación de la gestión automática de la información, utilizando dispositivos electrónicos y sistemas computacionales. Además, está definida como el procesamiento automático de la información. Para que un sistema informático funcione correctamente debe cumplir con tres requisitos fundamentales: [1]

- En su inicio, la captación de la información tiene que ser digital.
- Durante el proceso, cumplir con el tratamiento de la información.
- En la salida, transmitir los resultados binarios.

Desde hace varios años, la informática se ha fundamentado como una de las herramientas de apoyo al proceso sanitario; evolucionando como ciencia en la informática médica. El objetivo principal de esta ciencia es mejorar el manejo de la gestión de la información de la salud, para garantizar el aumento en la calidad de la atención sanitaria. Para lograr dicho objetivo, no sólo utilizan métodos novedosos, sencillos y eficaces de gestión administrativa en instituciones médicas, sino que también, dispone de recursos informáticos de valor que reducen la posibilidad de errores en el diagnóstico de las enfermedades. Estos además, facilitan el trabajo de las estadísticas médicas, aumentan la confiabilidad de las mismas y a su vez se ahorra tiempo en cuanto a la realización del análisis para comprobar teorías o suposiciones. [2]

En esta rama, las Tecnologías de la Información y las Comunicaciones (TIC) han brindado apoyo destacándose en el fortalecimiento de dichos sistemas de salud, debido a que presentan un gran potencial de impacto en casi todos los aspectos de este sector. [3]

En Cuba, antes del triunfo de la Revolución no se había logrado la posibilidad de brindar una colaboración médica, pero ya en 1960 ofrece su primera ayuda médica internacional enviando una brigada de colaboradores a Chile, después del terremoto que azotó a ese país. [4]

A fines de 1962, en el acto inaugural de la Facultad de Ciencias Básicas y Pre-clínicas "Victoria de Girón", el Comandante en Jefe Fidel Castro anunció al pueblo de Cuba la decisión del Gobierno revolucionario de brindar ayuda médica internacional en el campo de la salud. Esta expresión se pone de manifiesto al año siguiente cuando comienza la colaboración médica internacional cubana, con el envío de la primera brigada médica cubana a Argelia. [5]

Entre las funciones llevadas a cabo por la colaboración médica cubana se destaca la asistencia médica clínico-quirúrgica a la población de los países en los lugares más apartados, además del desarrollo conjunto de las campañas de educación sanitaria y de vacunación masiva de la población, el control

higiénico-epidemiológico y la prestación de servicios en brigadas médicas a damnificados por huracanes, sismos y otras contingencias en diferentes países. La mayor expresión de solidaridad e internacionalismo de la colaboración médica de Cuba se evidenció a finales de 1998 tras el azote del huracán Mitch a varios países de Centroamérica, principalmente a las naciones de Honduras y Guatemala. [6]

Durante los 45 años de colaboración médica más de 37 800 trabajadores de la salud han cumplido misiones en 83 países. De ese impresionante número de trabajadores los médicos sobrepasan los 25 700, quienes en su mayoría han brindado sus conocimientos en zonas intrincadas y de difíciles condiciones de vida, siendo portadores de la calidad científica y el ejemplo de los profesionales formados por la Revolución. [7]

Debido al considerable número de cooperantes cubanos existentes decididos a brindar ayuda solidaria a países hermanos, surge la necesidad de crear la Unidad Central de Cooperación Médica (UCCM), con el fin de llevar un control seguro de la información sobre los colaboradores cubanos. Actualmente es la institución rectora donde se gestiona la información de las misiones médicas cubanas en el exterior. La UCCM está constituida por varios departamentos, entre ellos se encuentra el Departamento de Economía, el mismo es el encargado de financiar el salario de los colaboradores en misiones médicas.

En sus inicios, todo el proceso que se desempeñaba en este departamento se realizaba de manera manual, provocando así, la pérdida y deterioro de valiosa información de los colaboradores.

El uso de las TIC permite evitar la ocurrencia de estos inconvenientes, y facilita la organización y gestión de los datos de los colaboradores de la salud, de modo que se puede realizar un pago fácil, confiable y eficaz.

Por tal motivo, la UCCM en conjunto con la Universidad de las Ciencias Informáticas (UCI) y en colaboración con la empresa de Soluciones Informáticas (SOFTEL), se propuso desarrollar un sistema elaborado con tecnologías Web, lenguaje de programación PHP 5.1, y gestor de base datos MySQL. Esta aplicación presenta una arquitectura que no se ajusta a los lineamientos definidos por el Centro de Informática Médica (CESIM), al no utilizar el sistema gestor de bases de datos, lenguaje de programación, framework de desarrollo y patrón de arquitectura.

Las causas por las cuales el sistema no se encuentra funcionando correctamente son:

- No proporciona la trazabilidad y la auditoría, que son elementos fundamentales para la seguridad de los recursos monetarios que se manejan en él.
- La aplicación no cuenta con las mejores prácticas y patrones de diseño propuestos para la Web.

- El Módulo de Pago se encuentra separado del resto de los módulos del Sistema de Gestión de la Colaboración Médica v1.0, pero comparten un mismo modelo de datos, lo que implica duplicación de código fuente y que los cambios aplicados en uno repercuten fuertemente en los demás.
- No se utiliza un patrón de arquitectura que sea capaz de separar la lógica del negocio, la lógica del servidor y la presentación de la aplicación.
- El tiempo de respuesta del sistema es lento debido a la sobrecarga de almacenamiento temporal de datos (caché) con el uso excesivo de la variable Sesión.

Debido a los problemas planteados anteriormente, se ha determinado como **problema científico**: ¿Cómo mejorar la trazabilidad y la auditoría, siguiendo los lineamientos de arquitectura definidos por el Centro de Informática Médica en el Módulo de Pago de los colaboradores en misiones médicas, en la versión 1.0 del Sistema de Gestión para la Colaboración Médica en la UCCM?

El **objeto de estudio** del presente trabajo se enfoca en la gestión de la información en el Departamento de Economía de la UCCM. El **campo de acción** se enmarca en la información del pago a los colaboradores de la salud en el Departamento de Economía de la UCCM.

Para solucionar las problemáticas mencionadas, se define como **objetivo general**: Desarrollar el Módulo de Pago en la versión 2.0 del Sistema para la Colaboración Médica, con el fin de mejorar la trazabilidad y la auditoría siguiendo los lineamientos de arquitectura definidos por el Centro de Informática Médica.

Las **tareas de la investigación** que se llevarán a cabo para darle cumplimiento al objetivo trazado son:

- Investigar la arquitectura y trazabilidad del manejo de información en los sistemas existentes relacionados con el pago de nóminas, para obtener un mejor punto de partida en la realización del nuevo sistema.
- Analizar la forma de pago de los colaboradores en misiones médicas que se efectúa en el Departamento de Economía de la UCCM, para obtener una visión general del proceso.
- Asimilar las tecnologías y arquitectura definidas por el: Departamento Sistemas de Apoyo a la Salud (SAS) en el desarrollo de la nueva solución, para mantener una homogeneidad en el uso de las herramientas utilizadas en el mismo.
- Desarrollar los artefactos que propone el Proceso Unificado de Desarrollo en sus flujos de trabajo enfocado al trabajo con el framework Symfony.
- Rediseñar la Interfaz Gráfica de Usuario usando ExtJS 3.0.1, para proveer una interfaz más usable, amigable e interactiva al usuario.

- Implementar las funcionalidades del Módulo de Pago, consumiendo los servicios Web Registro de Ciudadanos (RC), Registro de Personal de la Salud (RPS), Registro de Localidades (RL), Sistema de Administración (SAAA), para obtener la información necesaria durante el desarrollo del sistema.

Luego de desarrollar el Módulo de Pago del Sistema para la Colaboración Médica v 2.0, se garantizará la centralización y seguridad de los usuarios que se autentican en el sistema. Además, se proporcionará la trazabilidad y la auditoría garantizando la rapidez y efectividad al gestionar la información de los colaboradores de la salud, permitiendo la obtención de reportes mediante la recuperación de datos históricos almacenados en el sistema.

El presente trabajo de diploma contiene cuatro capítulos, donde se desarrolla el Módulo de Pago de la versión 2.0 del Sistema para la Colaboración Médica de la Unidad Central de Cooperación Médica:

El primer capítulo: **Fundamentación Teórica**, estará dedicado a diferentes aspectos teóricos relacionados con el desarrollo de este trabajo. Se abordarán conceptos fundamentales relacionados directa o indirectamente con el módulo de pago a los colaboradores para un mejor entendimiento del mismo, el estado del arte, los antecedentes del sistema, un análisis de las tecnologías en las cuales se sustenta el desarrollo del sistema, así como las herramientas que se utilizaron para el desarrollo del mismo.

El segundo capítulo: **Características del sistema**, posibilita obtener una visión más clara de los procesos del pago al colaborador en el modelado del negocio y del sistema. Se describen los actores, trabajadores, el modelo del negocio y el modelo de objetos, así como los diagramas de actividades y los requisitos tanto funcionales como no funcionales, especificando cada uno de ellos en aras de que se cumpla con las necesidades del cliente.

El tercer capítulo: **Análisis y Diseño**, profundiza el análisis y el diseño del sistema. Se define el diagrama de clases del análisis, que forma parte de los casos de uso y las relaciones existentes entre ellas. Se realizan los diagramas de interacción por cada realización de caso de uso que apoya luego la fase de implementación. Se obtienen las clases del diseño que presentan a la aplicación, además de una explicación sobre los principios de diseño que se tuvieron en cuenta para la realización del sistema. También se realiza la descripción de la interfaz, la descripción de la seguridad y la concepción de la ayuda para un mejor manejo de la aplicación.

El cuarto capítulo: **Implementación**, lleva el resultado del diseño a componentes que son ficheros de código fuente y scripts; estos son los archivos que finalmente integran la aplicación de forma física. En el capítulo se indica cómo los elementos del diseño se implementarán y se organizarán de acuerdo al

diagrama de despliegue, que permite la configuración de nodos de procesamiento y los componentes que residen en ellos. Además, se realizará el diagrama de componente general que organiza las dependencias entre un conjunto de componentes; todo dando lugar al buen control de la aplicación.

Capítulo 1. Fundamentación Teórica.

El presente capítulo tiene como objetivo fundamental abordar distintos temas que sirven de soporte teórico al sistema a implementar. Contiene la realización de un estudio detallado de las diferentes tecnologías existentes en la actualidad, plasmándose los aspectos fundamentales que influyeron en su elección.

Se aborda detalladamente los conceptos fundamentales, el estado del arte del sistema y se hace alusión a los antecedentes de los sistemas que han tenido relación con la gestión que se realiza para lograr el pago de los médicos colaboradores de Cuba.

Además, se realiza un análisis de la metodología a utilizar, el servidor de bases de datos, el servidor Web, y los lenguajes utilizados en la aplicación. Finalmente, se hace referencia a las herramientas que son necesarias en la ingeniería y desarrollo de la aplicación.

1.1 Conceptos fundamentales.

1.1.1 Ministerio de Salud Pública (MINSAP).

El MINSAP es el organismo rector del Sistema Nacional de Salud, encargado de dirigir, ejecutar y controlar la aplicación de la política del Estado y del Gobierno en cuanto a la salud pública, el desarrollo de las Ciencias Médicas y la Industria Médico Farmacéutica. Además, es el órgano responsable de todas las misiones médicas en el exterior.

1.1.2 Sistema Nacional de Salud (SNS).

El Sistema Nacional de Salud cubano depende del Ministerio de Salud Pública de Cuba. Este tiene el objetivo de distribuir de manera equitativa y justa los recursos nacionales en el sector de la Salud. A través de este sistema se establecen conexiones tanto intersectoriales como intersectoriales concertadas. El SNS incorpora a todas las instituciones que fueron creadas con la finalidad específica de participar en las actividades de promoción, recuperación y rehabilitación de la salud y prevención de la enfermedad de individuos y comunidades, dentro del territorio de la República. Además, tiene como finalidad primordial prestar servicios a todas las personas de manera equitativa, oportuna y eficiente, sin discriminación de ninguna clase, mediante acciones de recuperación y rehabilitación integral del enfermo. [8]

1.1.3 Informática para la Salud (INFOMED).

INFOMED trabaja para construir colectivamente un ecosistema de personas, servicios y fuentes de información para la salud. El mismo debe permitir el acceso oportuno y eficiente a información de calidad, y desplegar las capacidades creativas de los miembros de la red como productores de información y

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

conocimientos, para lograr las metas de salud de nuestro país y de otros pueblos del mundo. Se ha caracterizado por el uso de las tecnologías de la información y la comunicación con una visión social y a partir del desarrollo de las capacidades locales. Fue la primera red nacional que utilizó el sistema operativo GNU/Linux en todos sus servidores. [9]

1.1.4 Ministerio de Inversión y Colaboración Extranjera (MINVEC).

El MINVEC es el organismo de la administración central del estado a cargo de la promoción de las inversiones foráneas en el país. También dirige y controla el proceso de negociaciones para constituir las diferentes formas de asociaciones económicas con participación extranjera. La inversión extranjera en Cuba se concibe como un complemento a los esfuerzos nacionales por impulsar el desarrollo y a través de este proceso se buscan tres elementos básicos: tecnología, mercados de exportación y capital. [10]

1.1.5 Cooperación.

La Cooperación es el aporte de origen externo que se solicita a países u organismos internacionales, con el fin de apoyar el desarrollo nacional mediante acciones, proyectos y programas específicos. Además, se refiere abstractamente a todo proceso en donde se involucre el trabajo de varias personas en conjunto. [11]

1.1.6 Unidad Central de Cooperación Médica (UCCM).

La UCCM es la institución encargada de gestionar y almacenar la información referente a los colaboradores cubanos de la salud.

1.2 Navegadores.

1.2.1 Mozilla Firefox e Internet Explorer (IE).

Ambos exploradores presentan una forma rápida y eficiente de navegar por la Web, que permiten abrir varias páginas en una misma ventana dentro de pestañas separadas. Abren enlaces en segundo plano mientras se está leyendo otra página, continuando luego con los enlaces cuando se haya terminado, estando las páginas disponibles cuando se les necesite. [12]

Este sistema puede ser visualizado tanto en Mozilla Firefox como en IE debido al framework utilizado. Es recomendable utilizar Mozilla Firefox ya que se encuentra disponible para un buen número de plataformas distintas, utilizando las mismas características en cada una de ellas, al contrario de IE, que sólo puede ejecutarse en Windows. [13]

1.3 Estado del arte.

Debido a las características que presenta el método de realización del pago a los colaboradores en misiones médicas en el entorno cubano, es importante destacar que en el mundo existen diferentes sistemas de software enfocados al pago de nóminas que son utilizados por varias empresas en dependencia de sus necesidades. Entre los sistemas existentes se encuentran:

- El Sistema de Nómina y Recursos Humanos: Es una solución centralizada corporativa de Software de Nómina que permite realizar pagos semanales, quincenales o mensuales a trabajadores, ya sea por nómina individual, grupos de nóminas, regiones y por países, cumpliendo con todas las leyes del Trabajo y Hacendarías impuestas por la ley de México. Los tiempos de implementación del sistema dentro de la organización que la utilice son cortos en comparación con otras soluciones de Software de Nómina de compañías extranjeras. El sistema se encuentra realizado con el lenguaje de programación Progress Software, y puede usar la Base de Datos de Progress o conectarse de manera nativa a Bases de Datos: Oracle, SQL Server, Sybase y con driver de conexión a otras. [14]
- Software de Nómina (SPD NOMINA): Es un software de nómina que ha sido desarrollado de acuerdo a las necesidades de administración y control para el cálculo de conceptos y pago de nómina que presentan las empresas. Este software fue desarrollado en México, cuenta con procesos y cálculos apegados a la legislación de dicha nación. Integra todo lo relacionado a un programa de nómina de manera que no es necesario llevar otros controles adicionales en papel u hojas de cálculo. Es un sistema robusto y configurable, maneja un alto volumen de transacciones y es totalmente portable a plataformas Windows, Unix, y Linux. [15]
- Sistema de Nómina (ENIAC): El sistema permite registrar y controlar los datos de los obreros y empleados de la empresa, además de captar las excepciones en cuanto a percepciones y deducciones en cada período. Emite las declaraciones mensuales o bimestrales y la póliza de afectación contable, con el detalle que cada empresa considere necesario. En este sistema se pueden manejar diferentes tipos de nóminas: semanal, catorcenal, quincenal y mensual. [16]

Luego de analizar los sistemas antes mencionados, se concluyó que su uso no está restringido a un sector en específico, por lo que pueden ser desplegados en varias empresas para el manejo de sus nóminas. De modo que las diferencias existentes entre una y otra pueden ser salvadas por las opciones de configuración que brindan los sistemas en cuanto al entorno donde serán aplicados. A pesar de presentar esas características, estos sistemas no son útiles en entidades de salud debido a que la

realización de las nóminas se gestiona de forma diferente, por lo que no es posible su utilización en la UCCM. Además, se debe destacar que estos sistemas no presentan información visible al usuario sobre la trazabilidad y la auditoría en caso de que estas existan.

1.3.1 Nivel nacional.

En el país diversas instituciones han desarrollado aplicaciones informáticas con el objetivo de automatizar el proceso de pago a la población, y así, evitar equivocaciones realizando dicha gestión manualmente. Entre los sistemas existentes en los últimos años se encuentran:

- Sistema de Nómina/Pre-nómina (Cóndor): Incluye el paquete Nómina, que controla los pagos y las retenciones de salarios de los trabajadores, y Pre-nómina que es el receptor de información de recursos humanos. Entre ambos existe el enlace que permite su funcionamiento armónico. Este sistema actualmente se encuentra en su versión 6.0. [17]
- Sistema de Nóminas (RODAS XXI): Es capaz de guardar por trabajador los pagos y las 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. Permite 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. [18]
- Sistema de Nóminas (Jet Nóminas): 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 y demás). Tiene incorporado el cálculo de nóminas en divisas y permite descontar automáticamente de la nómina el contravalor en moneda nacional. De la misma forma permite efectuar el pago de empresas bajo el régimen de perfeccionamiento empresarial, incluyendo el uso del coeficiente de participación laboral y la aplicación del aporte y descuento en caso que proceda. [19]

Al finalizar el estudio de estos sistemas, se determinó que pueden ser utilizados en diversos tipos de empresas puesto que son compatibles con las regulaciones que establecen los organismos de la Administración Central del Estado para sus entidades. Además, el entorno de ejecución de estos sistemas no es único debido a la flexibilidad y adaptabilidad que presentan. Estos no están dirigidos a instituciones

de salud debido a que el cálculo de las nóminas se realiza de manera diferente. Universidad de las Ciencias Informáticas (UCI).

1.3.2 Universidad de las Ciencias Informáticas (UCI).

Actualmente en la UCI está en uso un sistema de Planificación de Recursos de la Empresa (ERP) denominado Sistema de Gestión Integral (ASSETS) en su versión 2.1, el cual cuenta con un módulo de recursos humanos. Este sistema permite generar automáticamente los comprobantes de operaciones de contabilidad por separado, es decir, genera los comprobantes de salario, los de vacaciones, los de subsidios, los de nóminas y los de reintegros. El sistema puede calcular y contabilizar nóminas de salario para varios tipos de pago, incluyendo el pago de horas extras y condiciones laborales anormales. De igual forma, calcula y contabiliza las nóminas de vacaciones y subsidios. [20]

Después de analizar este sistema, se comprobó que no es posible su utilización debido a que no se adapta a las necesidades de la UCCM en cuanto a la forma de pago de los colaboradores. Además, el proveedor en Cuba del sistema no permite que se la realicen modificaciones al código fuente.

1.4 Antecedentes del sistema.

En sus inicios, la UCCM contaba con un sistema que no permitía mantener un registro actualizado de los colaboradores de la salud, ni enviaba la información detallada y consolidada a los niveles superiores, por lo que no está preparado para desempeñar funciones de gestor de altas prestaciones. Este sistema fue elaborado en Delphi como lenguaje de programación y desarrollado en una base de datos en Microsoft Access 97. Fue creado en Santiago de Cuba, en el 2004. [21]

Por estas razones fue necesaria la implementación de un nuevo sistema con tecnologías Web, lenguaje de programación PHP 5 y una base de datos que utiliza MySQL como gestor. Este software a diferencia del anterior gestiona el pago al colaborador a través de la Resolución Ministerial No. 321/08 del Ministerio de Salud Pública; además, permite visualizar la información de acuerdo con niveles jerárquicos establecidos.

Aún así este sistema posee un bajo rendimiento y no permite realizar la trazabilidad y auditoría de las acciones efectuadas por el usuario.

1.5 Gestión de proyecto.

1.5.1 Modelo de Capacidad y Madurez (CMMI, por sus siglas en inglés).

El CMMI es un marco de referencia que las organizaciones pueden emplear para mejorar sus procesos de desarrollo, adquisición, y mantenimiento de productos y servicios. El mismo es una nueva generación de modelos de madurez, que basado en los principios de calidad total, propone un conjunto de buenas prácticas que enseñan qué hacer para implantar procesos productivos más efectivos.

Este modelo plantea la concepción de 5 posibles niveles de madurez que puede alcanzar una organización dependiendo del grado de sofisticación de sus procesos. A su vez, cada nivel, con excepción del inicial, queda caracterizado por un conjunto de áreas de proceso que agrupan prácticas que, al ser ejecutadas colectivamente, permiten cumplir con algún objetivo que es considerado importante para el modelo. Estos pueden ser visualizados de forma continua o de forma escalonada. [22]

Los niveles de CMMI son:

- Inicial.
- Administración básica de proyectos.
- Proceso estandarizado.
- Proceso administrado cuantitativamente.
- Proceso en mejora continua.

El nivel 2 de CMMI tiene como objetivo conseguir que en los proyectos de la organización haya una buena gestión de los requisitos y que los procesos estén planeados, ejecutados, medidos y controlados. De esta manera se ayuda a que los proyectos se ejecuten y gestionen de acuerdo con los planes de la organización, a que se conozca el estado de los elementos de trabajo de este y a que estos elementos satisfagan las especificaciones, estándares y objetivos. Dentro de este nivel se incluyen 7 áreas de trabajo las cuales se relacionan mutuamente: Administración de requerimientos (REQM), Planificación de proyectos (PP), Monitoreo y control de proyecto (PMC), Medición y análisis (MA), Aseguramiento de la calidad de productos y procesos (PPQA), Administración de la Configuración (CM) y Administración de acuerdos con proveedores (SAM). [23]

La Universidad de las Ciencias Informáticas, en busca de mejorar la eficiencia y calidad de los procesos de desarrollo de software y la gestión de requerimientos, ha adoptado como estrategia la aplicación en la organización del nivel anteriormente especificado. Trabajando dentro del mismo, en el área de procesos de REQM.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

Esta área tiene como propósito mantener bajo control los requerimientos que el producto a desarrollar deberá satisfacer. Las prácticas incluidas aquí apuntan a que los requerimientos no solo estén claramente identificados, sino también que todos los involucrados en el proyecto estén de acuerdo en su significado. Un tema fundamental planteado en esta área de proceso es que cualquier cambio realizado a los requerimientos se efectúe de manera controlada y que el resto de los artefactos del proyecto se mantengan consistentes.

Un aspecto importante en esta área es el de trazabilidad bidireccional; pues cuando los requerimientos son correctamente administrados se debería estar en condiciones de relacionar cuál ha sido el origen de los requerimientos, cuál es la relación entre los requerimientos de bajo nivel y los de alto nivel, cuáles son los artefactos relacionados con los requerimientos, y cuáles componentes del producto implementan cada requerimiento. Esta práctica es sumamente importante para poder realizar un buen análisis de impacto ante posibles cambios, y fundamental para poder determinar si el alcance del proyecto ha sido cubierto o no.

El modelo CMMI es una alternativa para definir qué hacer durante el desarrollo de un proyecto, sin embargo este no propone cómo realizarlo. Es de este modo que CMMI se apoya en metodologías que suministren esta información, teniendo en cuenta que la UCI hasta el momento ha interactuado con el Proceso Unificado de Desarrollo, se ha mantenido esta metodología de desarrollo de software. De esta manera, el trabajo en conjunto de CMMI como plan de mejora, y el Proceso Unificado de Desarrollo como metodología de apoyo de este, guían a la organización para la realización del producto. El Proceso Unificado de Desarrollo por su parte utiliza Leguaje Unificado de Modelado como lenguaje para modelar todos los artefactos de los flujos de trabajo.

1.5.2 Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés).

RUP, no es más que un proceso de desarrollo de software, es decir, un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. RUP, además de ser un proceso de desarrollo es también un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y tamaños de proyecto. [24]

Este proceso está compuesto por cuatro fases: Inicio, Elaboración, Construcción y Transición. Además, contiene nueve flujos de trabajo: Modelo de negocio, Requerimiento, Análisis y diseño, Implementación, Prueba, Instalación, Ambiente, Administración de proyecto, Administración de configuración y cambios.

También utiliza un Leguaje Unificado de Modelado para preparar todos los esquemas de un sistema software. [25]

Las características que definen a RUP son: [26]

- **Dirigido por casos:** Se utiliza respectivamente para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de casos de uso y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba, etc. Cuando los casos de usos se unen constituyen el modelo de casos de usos el cual describe la funcionalidad total del sistema.
- **Centrado en la arquitectura:** Incluye los aspectos estáticos y dinámicos más significativos del sistema. En ella influyen varios factores, como la plataforma en la que tiene que funcionar el software, los bloque de construcción reutilizables de que se dispone, las consideraciones de implementación, sistemas heredados y requisitos no funcionales. La arquitectura es una vista del diseño completo con las características más importantes resaltadas, lo cual ayuda al arquitecto a centrarse en los objetivos adecuados, ya sea la comprensibilidad, la capacidad de adaptación al cambio y la reutilización.
- **Iterativo e incremental:** Las iteraciones ofrecen como resultado un incremento del producto desarrollado que mejora las funcionalidades del sistema en desarrollo, es decir, las iteraciones hacen referencias a pasos en el flujo de trabajo y los incrementos, al incremento del producto.

1.5.3 Leguaje Unificado de Modelado (UML, por sus siglas en inglés).

UML 2.0 define reglas sintácticas que especifican cómo combinar elementos del lenguaje, y puede ser usado en todas las fases de desarrollo del software. Este lenguaje está diseñado para la visualización, especificación, construcción y documentación de los artefactos del sistema en los que el software deja un papel importante. [27]

UML es un lenguaje más expresivo, claro y uniforme que los anteriores definidos para el diseño Orientado a Objetos. No garantiza el éxito de los proyectos pero sí mejora sustancialmente el desarrollo de los mismos al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios. [28]

1.6 Tecnologías y lenguajes utilizados.

1.6.1 Servicios Web.

Un servicio Web es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios Web para intercambiar datos en redes de computadoras. [29]

1.6.2 Servidor Web Apache.

El servidor Web Apache 2.2 fue creado para proveer un alto grado de calidad y fortaleza para las implementaciones que utilizan el protocolo HTTP. Es un software libre y multiplataforma que permite a clientes o instituciones construir sistemas confiables con fines experimentales o para resolver un problema específico de la organización. [30]

1.6.3 Sistemas Gestores de Base de Datos (SGBD) PostgreSQL.

PostgreSQL 8.3 es un sistema de gestión de bases de datos objeto-relacional, presenta varias características, entre ellas: [31]

- Posee una gran escalabilidad. Es capaz de ajustarse al número de computadoras y a la cantidad de memoria que posee el sistema de forma óptima, pudiendo soportar una mayor cantidad de peticiones simultáneas de manera correcta (algunas comparaciones sugieren que soporta el triple de carga de lo que soporta MySQL).
- Implementa el uso de sub-consultas y transacciones, haciendo su funcionamiento mucho más eficaz, y ofreciendo soluciones en campos en los que MySQL no podría.
- Tiene la capacidad de comprobar la integridad referencial, así como también, la de almacenar procedimientos en la propia base de datos.

1.6.4 Ajax.

Ajax es un conjunto de tecnologías para el desarrollo de aplicaciones Web interactivas o aplicaciones fuertes en Internet. Es una tecnología asíncrona, en el sentido de que los datos adicionales requieren ser traídos del servidor y se cargan en segundo plano sin interferir con la visualización o el comportamiento de la página. [32]

Se ha demostrado también que Ajax es válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores debido a que está basado en estándares abiertos como JavaScript. [33]

1.6.5 Framework Symfony.

Symfony 1.4 es un framework diseñado para optimizar el desarrollo de las aplicaciones Web por las características que presenta. Este framework separa la lógica de negocio, la lógica de servidor y la

presentación de la aplicación Web, proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. [34]

Symfony está desarrollado completamente con PHP 5, es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataforma Windows. [35]

1.6.6 ExtJS.

ExtJS 3.01 es una librería JavaScript ligera de alto rendimiento que se utiliza para crear interfaces de usuarios, más agradables e interactivas. Se caracteriza por sus componentes de interfaz de usuario personalizables, con buen diseño y documentación. [36]

1.6.7 Lenguaje de programación Web Hypertext Pre-processor (PHP).

PHP 5.2.6 es un lenguaje de programación independiente de la plataforma que se use. Es rápido, con una gran librería de funciones, una amplia documentación y sobre todo es un lenguaje del lado del servidor gratuito; es decir, se ejecuta en el servidor Web justo antes de que se envíe la página a través de Internet al cliente. Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL, Oracle y Microsoft SQL Server; lo cual permite la creación de aplicaciones Web muy robustas. [37]

1.6.8 JavaScript.

JavaScript es un lenguaje que puede ser utilizado por profesionales y por quienes se inician en el desarrollo y diseño de sitios Web. No requiere de compilación ya que funciona del lado del cliente. Es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros. Además, pone a disposición del programador todos los elementos que forman la página Web, para que éste pueda acceder a ellos y modificarlos dinámicamente. [38]

En este lenguaje no es necesario declarar los tipos de variables que van a utilizarse, ya que realiza una conversión automática de tipos, además, las referencias a objetos se comprueban en tiempo de ejecución debido a que no es un lenguaje compilado. [39]

1.6.9 Herramienta JMeter.

Es una herramienta Java que permite realizar Pruebas de Rendimiento y Pruebas Funcionales sobre Aplicaciones Web que se destaca por su versatilidad, estabilidad y por ser gratuita. Además permite la ejecución de pruebas distribuidas entre distintos ordenadores, para realizar pruebas de rendimiento. [40]

1.7 Arquitectura del sistema.

1.7.1 Modelo Vista Controlador (MVC).

El MVC es un patrón de arquitectura de software muy utilizado en la Web, describe la forma de organizar el código de una aplicación separando los datos, la interfaz de usuario, y la lógica de control en tres componentes distintos: [41]

- Modelo: Representa la lógica de negocio, es decir encapsula los datos y las funcionalidades, es independiente de cualquier representación de salida y comportamiento de entrada.
- Vista: Es el objeto que muestra la información al usuario. Puede ser una página Web o una parte de ésta. Cada vista tiene asociado un componente controlador. Transforma el modelo en una página Web que permite al usuario interactuar con ella e interpretar el código que provee de datos dinámicos a la página.
- Controlador: Se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

1.7.2 Arquitectura Cliente-Servidor.

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Como ventajas principales de esta arquitectura se tiene el aumento de la productividad, la disminución de los costes de operación y la mejora en el rendimiento de la red. [42]

Entre las principales características de la arquitectura cliente/servidor, se pueden destacar que: [43]

- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

1.8 Herramientas.

1.8.1 Enterprise Architect 7.1.

Es una herramienta flexible, completa y potente de modelado en UML bajo plataforma Windows o Linux. Provee lo más nuevo en desarrollo de sistemas, administración de proyectos y análisis de negocio. Abarca integralmente el ciclo de vida de un proyecto software cubriendo el desarrollo desde el levantamiento de los requerimientos a través de las etapas de análisis, modelos de diseño, pruebas y finalmente el mantenimiento y re-uso. [44]

Fue construida en base al excepcional éxito de las versiones previas con un completo soporte para el estándar UML 2.0. Puede generar código fuente C++, Java, C#, VB.Net, Visual Basic, Delphi, PHP, Python y ActionScript. Permite la ingeniería inversa para muchos sistemas que contengan bases de datos Oracle, SQL Server, MySQL, Access, PostgreSQL y otras. [45]

1.8.2 NetBeans.

NetBeans 6.8 es un proyecto de código abierto con una gran base de usuarios y una comunidad en constante crecimiento. El soporte para PHP ha sido extendido para incluir el framework Symfony y PHP 5.3. [46]

NetBeans 6.8 es un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés) gratuito para desarrolladores de software. Ofrece todas las herramientas necesarias para crear aplicaciones Web y de escritorio con el lenguaje Java, C/C ++ y lenguajes dinámicos como PHP y JavaScript. Es fácil de instalar y se puede ejecutar tanto en Windows como en Linux; también tiene detección de errores de sintaxis en tiempo real. [47]

En este capítulo, luego de una investigación realizada se pudo constatar que ninguna de las aplicaciones existentes pueden darle solución a la situación problemática que presenta la UCCM. Además, se propuso la utilización del patrón de arquitectura MVC, el cual provee como beneficio que si se realiza un cambio en algunas de las capas del sistema, no se afectan las que no están implicadas en la modificación. Las tecnologías, arquitectura y lenguajes seleccionados para el desarrollo de esta aplicación Web se ajustan a los lineamientos de arquitectura definidos por el CESIM.

Capítulo 2. Características del Sistema.

En este capítulo se explican los procesos del pago al colaborador en el modelado del negocio y del sistema. Se destacarán las reglas establecidas en el negocio, se identificarán los actores y trabajadores del mismo, así como el diagrama de casos de uso y su formato expandido. Además, se dan a conocer los requerimientos funcionales y no funcionales con que debe contar el sistema.

2.1 Reglas del negocio a considerar.

2.1.1 Ayuda familiar.

- La UCCM es la responsable de realizar la acreditación de las cuentas a los colaboradores mensualmente.
- No se realiza el pago a los colaboradores que se encuentran de vacaciones o de tránsito en Cuba.
- Un designado sólo puede efectuar el cobro a un colaborador.

2.1.2 Cuenta especial.

- La UCCM es la encargada de acreditar la cuenta especial de los colaboradores mensualmente.

2.1.3 Personalización.

- Se realiza las personalizaciones a los colaboradores que se encuentren cumpliendo misión en su destino y que no se le haya realizado antes.
- Las Unidades Provinciales realizan la personalización del pago semanalmente, ya sea de ayuda familiar o cuenta especial.

2.1.4 Liberaciones.

- La UCCM es la encargada de realizar las liberaciones de los colaboradores por vacaciones o por fin de misión.
- Se realiza la liberación por vacaciones cuando el colaborador está de tránsito en Cuba por vacaciones.
- Se realiza liberación por fin de misión cuando el colaborador concluye su misión.

2.1.5 Cuenta Vitalicia.

- La UCCM es la encargada de realizar la creación de la cuenta vitalicia.
- La cuenta vitalicia se crea una vez que se realiza la liberación por fin de misión y la misión finalizada pertenezca al programa integral de salud.

2.1.6 Ajuste Cuenta Ayuda Familiar.

- La UCCM es la encargada de realizar los ajustes de la cuenta de ayuda familiar.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

- Se le realizan ajustes por crédito o débito a la cuenta de ayuda familiar cuando ocurre una demora o una equivocación en el pago de la nómina.

2.1.7 Ajuste Cuenta Especial.

- La UCCM es la encargada de realizar los ajustes de la cuenta especial.
- Se le realizan ajustes por crédito o débito a la cuenta especial cuando ocurre una demora o una equivocación en el pago de su saldo.

2.2 Propuesta del sistema.

Se desea informatizar toda la información del pago a los colaboradores de la salud que se gestiona en la UCCM mediante una aplicación Web, para mejorar la trazabilidad y la auditoría siguiendo los lineamientos de arquitectura definidos por el CESIM. Esta información será visualizada en la nación, los municipios de los cooperantes y sus provincias a través de la red telemática INFOMED. El sistema estará integrado a los servicios informatizados del RIS.

2.3 Modelo del negocio.

El modelo de negocio describe las funcionalidades que éste desea desarrollar y cómo los clientes se relacionan con él, es decir, ayuda a comprender cómo interactúan los actores del negocio con los casos de uso del mismo. [48]

2.3.1 Un actor del negocio.

Es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. [49]

En el presente sistema se pone de manifiesto de la siguiente manera:

Actores.	Justificación.
Colaborador:	Es el principal beneficiado con los resultados del negocio. El más interesado en recibir su salario eficientemente, con el cual se satisface.
Designado:	Es el encargado e interesado en recibir el salario de la ayuda familiar con el cual se beneficia.

Tabla #1. Descripción textual de los actores del negocio.

2.3.2 Un trabajador del negocio.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

Es quien define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Realizan las actividades y son propietarios de elementos. [50]

En el presente sistema se pone de manifiesto de la siguiente manera:

Trabajadores.	Justificación.
Especialista de la UCCM:	Encargado de pagarle al colaborador su salario según la evaluación adquirida al concluir su misión. También se encarga de pagarle al colaborador de misión especial una vez cumplido sus 12 meses en la misma.
Especialista de la UPCM:	Encargado en realizar las personalizaciones con todos los datos de los colaboradores, además de crear la cuenta especial y la cuenta de ayuda familiar.
Especialista del Banco:	Encargado de pagarle al beneficiario su salario de ayuda familiar mensualmente.

Tabla #2. Descripción textual de los trabajadores del negocio.

2.4 Modelo de casos de uso del negocio.

Describe los procesos de negocio de una empresa en términos de casos de uso y actores del negocio, que se corresponden con los procesos del negocio y los clientes, respectivamente. [51]

2.4.1 Diagrama casos de uso del negocio.

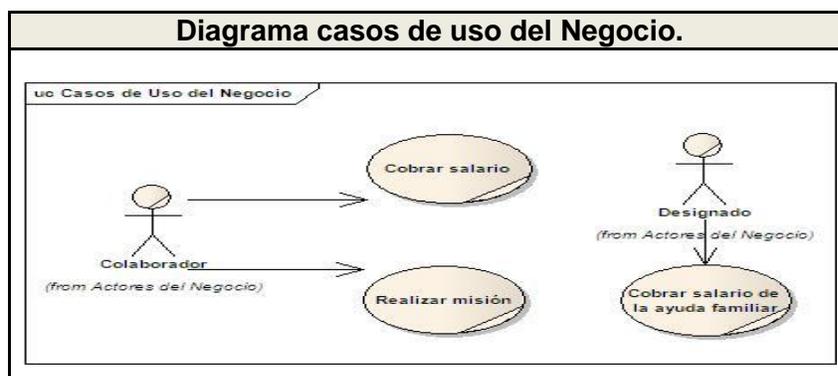


Figura #1: Diagrama de CUN.

2.4.2 Descripción de los Casos de uso del negocio.

CUN "Realizar Misión".

Descripción literal del Caso de Uso del Negocio:

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

Caso de Uso del Negocio:	Realizar Misión.
Actores:	Colaborador.
Resumen:	El caso de uso comienza cuando el colaborador se dirige a cumplir misión hacia otro país.
Casos de Uso asociados:	-
Acción del actor.	Respuesta del proceso de negocio.
1- El colaborador sale de misión.	2- El especialista de la UPCM realiza la personalización del colaborador, donde crea la cuenta especial y a la cuenta de ayuda familiar. 3- El especialista de la UPCM confecciona el reporte de colaborador.
Curso Alternativo de los eventos:	-
Mejoras propuestas.	En el diagrama de actividades “Realizar Misión”, se muestra de color gris todo lo que se desea informatizar para que el sistema funciones lo mejor posible. Para un mejor funcionamiento del sistema se informatizará: <ul style="list-style-type: none"> • La realización de la personalización del colaborador. • La confección del reporte del colaborador.

Tabla #3. Descripción textual del CU “Realizar misión”.

CUN “Cobrar Salario”.

Descripción literal del Caso de Uso del Negocio:

Caso de Uso del Negocio:	Cobrar Salario.
Actores:	Colaborador.
Resumen:	El caso de uso inicia cuando el colaborador decide pasar a obtener su salario y el especialista de la UCCM acude a realizarle el pago.
Casos de Uso asociados:	Buscar colaborador.
Acción del actor.	Respuesta del proceso de negocio.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

<p>1- El colaborador realiza el pedido de su salario.</p>	<p>2- El especialista de la UCCM solicita la documentación al colaborador.</p> <p>3- El especialista de la UCCM verifica los datos del colaborador.</p> <p>4- El especialista de la UCCM verifica la cuenta especial del colaborador.</p> <p>5- El especialista de la UCCM verifica si concluyó su misión.</p> <p>6- El especialista verifica la evaluación del colaborador.</p> <p>7- El especialista si el colaborador está de vacaciones.</p> <p>8- El especialista de la UCCM verifica si el tipo de misión del colaborador es misión especial.</p> <p>9- El especialista verifica su tiempo de misión.</p> <p>10- Verifica si pertenece al PIS.</p> <p>11- El especialista de la UCCM paga salario al colaborador.</p>
<p>Curso Alternativo de los eventos:</p>	<p>3.1. En caso de que los datos del colaborador no estén correctos, no se realiza el pago de su salario.</p> <p>6.1. En caso que el colaborador no concluyó su misión con evaluación satisfactoria, no se realiza el pago de su salario.</p> <p>8.1. En caso que el colaborador esté de vacaciones y el tipo de misión del no sea misión especial, no se realiza el pago de su salario.</p> <p>9.1. En caso que el colaborador no lleve más de 12 meses de misión, no se realiza el pago de su salario.</p> <p>10.1. En caso que el colaborador no pertenezca al PIS, no se le crea la cuenta de vitalicio.</p>
<p>Mejoras propuestas.</p>	<p>En el diagrama de actividades “Cobrar Salario”, se muestra de color verde todo lo que se desea automatizar para que el sistema funcione lo mejor posible.</p> <p>Para un mejor funcionamiento del sistema se automatizará:</p> <ul style="list-style-type: none">• La verificación de los datos del colaborador.• La verificación de la cuenta especial del colaborador.• La verificación de si terminó la misión el colaborador.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

	<ul style="list-style-type: none"> La verificación de la evaluación de la misión del colaborador. La verificación del tipo de misión del colaborador. La verificación del tiempo en misión del colaborador.
--	--

Tabla #4. Descripción textual del CU “Cobrar salario”.

CUN “Cobrar Salario de Ayuda Familiar”.

Descripción literal del Caso de Uso del Negocio:

Caso de Uso del Negocio:	Cobrar Salario de Ayuda Familiar.
Actores:	Beneficiario.
Resumen:	El caso de uso inicia cuando el beneficiario decide pasar a obtener su salario y el especialista del banco acude a realizarle el pago.
Casos de Uso asociados:	<p>Buscar colaborador.</p> <p>Buscar beneficiario.</p>
Acción del actor.	Respuesta del proceso de negocio.
1- El beneficiario realiza el pedido del salario de la ayuda familiar.	<p>2- El especialista del banco le pide documentación al beneficiario.</p> <p>3- El especialista del banco verifica los datos del beneficiario.</p> <p>4- El especialista del banco verifica la cuenta de ayuda familiar del colaborador a la cual el beneficiario le retirará su salario.</p> <p>5- El especialista del banco paga salario de ayuda familiar al beneficiario.</p>
Curso Alternativo de los eventos:	3.1. En caso de que los datos del beneficiario no estén correctos, no se realiza el pago del salario de la ayuda familiar.
Mejoras propuestas.	<p>En el diagrama de actividades “Cobrar Salario de Ayuda Familiar”, se muestra de color verde todo lo que se desea automatizar para que el sistema funciones lo mejor posible.</p> <p>Para un mejor funcionamiento del sistema se automatizará:</p> <ul style="list-style-type: none"> La verificación de los datos del beneficiario. La verificación de la cuenta de la ayuda familiar del colaborador al cual el beneficiario le retirará el salario.

Tabla #5. Descripción textual del CU "Cobrar salario de ayuda familiar".

2.5 Modelo de objetos del negocio.

El modelo de objetos describe cómo colaboran los trabajadores y las entidades dentro de flujo de trabajo del proceso de negocio. En el modelo de objetos del negocio se muestra la participación de los trabajadores, las entidades del negocio y la relación que existe entre ellos. [52]

Diagrama de Clases del Modelo de Objetos del Negocio.

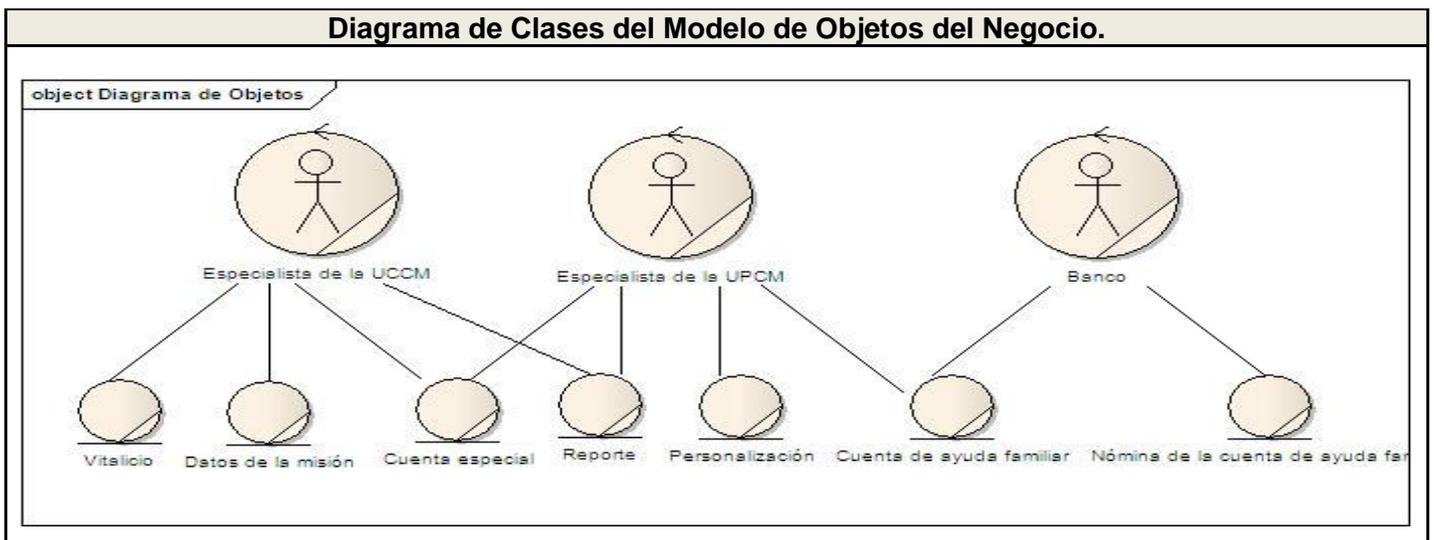


Figura #2: Diagrama de Clases del modelo de objetos del negocio.

2.5.1 Diagrama de actividades del Caso de Uso del Negocio (DCUN).

Un diagrama de actividad describe las actividades que deben ser realizadas en un proceso del negocio, así como las distintas rutas que pueden irse desencadenando. Este es dividido en canales, donde cada canal representa el actor que está llevando a cabo la actividad y muestra cómo se utilizan las entidades del negocio. [53]

Los diagramas de actividades se muestran en el [Anexo #1](#) del documento.

2.6 Modelo de sistema.

2.6.1 Captura de requisitos.

La captura de requisitos tiene como objetivo fundamental guiar el desarrollo hacia un sistema de software correcto, es decir, es la actividad que le permite al equipo de desarrollo de un sistema extraer las necesidades que debe cumplir el mismo para un mejor funcionamiento. [54]

El sistema a implementar deberá ser capaz de realizar las funcionalidades siguientes:

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

Requisitos Funcionales.	Descripción.
1. Autenticar.	Permite el acceso a la aplicación y a sus funcionalidades.
2. Buscar Colaborador.	Permite la búsqueda de un colaborador para realizarle las operaciones necesarias.
3. Listar Colaborador.	Muestra un listado con los colaboradores de la salud.
4. Buscar Ciudadano.	Permite la búsqueda de un ciudadano para realizarle las operaciones necesarias.
5. Cuenta de Ayuda Familiar:	
5.1. Mostrar salario del mes actual.	Muestra el salario de la ayuda familiar de un colaborador en el mes actual.
5.2. Mostrar salario histórico de nómina.	Muestra por mes y año el salario de un colaborador durante la misión actual.
5.3. Generar nómina.	Genera la nómina del colaborador por concepto de la cuenta de ayuda familiar.
5.4. Mostrar histórico.	Muestra un histórico de la cuenta de ayuda familiar con datos del colaborador a partir de una fecha dada.
6. Cuenta Especial:	
6.1. Mostrar salario histórico de nómina.	Muestra por mes y año el salario de un colaborador durante la misión actual.
6.2. Generar.	Genera la cuenta especial del colaborador.
6.3. Mostrar salario del mes actual.	Muestra el salario especial de un colaborador en el mes actual.
6.4. Mostrar histórico.	Muestra un histórico de la cuenta especial con datos del colaborador a partir de una fecha dada.
7. Ajustes de Cuentas:	
7.1. Ayuda familiar.	Permite ajustar los valores del crédito y el débito de la cuenta de ayuda familiar.
7.2. Especial.	Permite ajustar los valores del crédito y el débito de la cuenta especial.
7.3. Generar ajustes.	Permite generar los ajustes de crédito y débito que han sido

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

	introducidos.
8. Liberaciones:	
8.1. Vacaciones.	Realiza la liberación de las nóminas de un colaborador al estar de vacaciones.
8.2. Fin de Misión.	Realiza la liberación de las nóminas de un colaborador una vez terminada su misión.
9. Personalizaciones:	
9.1. Mostrar.	Muestra los colaboradores que estén personalizados y los datos respectivos a la cuenta especial y la cuenta de ayuda familiar.
9.2. Generar.	Genera la personalización a los colaboradores que aún no la tengan y se encuentren de misión.
9.3. Importar.	Permite importar los ficheros de las personalizaciones generadas por el sistema.

Tabla #6. Descripción textual de los requisitos funcionales.

Los requerimientos o requisitos no funcionales (RNF).

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto. [55]

Existen múltiples categorías para clasificar a los requisitos no funcionales, a continuación se muestran los más destacados en el este sistema: [56]

Número.	Categoría.	Enunciado.
RNF1	Software	Para la instalación y funcionamiento del sistema en el servidor es necesario el Sistema Operativo Windows o Linux, se requiere como SGBD a PostgreSQL 8.3, como servidor Web Apache 2.2 y como lenguaje de programación PHP 5.2.6. Además del flash player 10.
RNF2	Hardware	Se requiere una computadora con 128 Mb memoria de RAM o superior, un disco duro de 10Gb como mínimo, Procesador Pentium II 450 MHz o superior y una tarjeta de red sin importar la velocidad.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

RNF3	Apariencia o Interfaz externa	El software brindará una interfaz simple y de fácil uso para que el usuario no tenga dificultad al interactuar con el sistema. La interfaz del sistema se ajustará a los estándares establecidos para el desarrollo de un buen diseño, tendrá consistencia con el mundo real debido a que los conceptos manejados son conocidos y les resultan familiares a los usuarios. El sistema contará con dichas características teniendo en cuenta el propósito con el que se desarrolla la aplicación.
RNF4	Seguridad	Dispondrá de un mecanismo de seguridad basado en el modelo de autenticación y autorización para acceder al sistema. Se utilizará la encriptación de los datos que por cuestiones de seguridad no deben viajar al servidor en texto plano, como es el caso de las contraseñas. Se encriptará primero en el cliente y seguidamente se enviará al servidor.
RNF5	Confidencialidad	Según los permisos que posean los actores del sistema, se les permitirá realizar cambios en la información, mientras que a los usuarios solo se les permite visualizar la misma.
RNF6	Disponibilidad	Se garantizará a los usuarios del sistema el acceso a la información solicitada en todo momento, siempre y cuando tenga permiso para ello.
RNF7	Usabilidad	El sistema podrá ser usado por cualquier tipo de personas que posean conocimientos básicos en el manejo de la computadora y el ambiente Web en sentido general.
RNF8	Soporte	Se documentará la aplicación con un manual de ayuda con el objetivo de explicar el uso de la plataforma para garantizar el soporte de la herramienta. Se debe realizar el proyecto de forma que permita darle mantenimientos al sistema a fin de aumentar las funcionalidades y corregir los errores del mismo a través de versiones posteriores.
RNF9	Rendimiento	Teniendo en cuenta que la aplicación está concebida para un ambiente cliente/servidor, se garantizará que los tiempos de respuesta sean generalmente rápidos al igual que la velocidad de procesamiento de la información. Se seguirá una lógica de negocios en la comunicación entre el

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

		cliente y el servidor Web, ya que cada cual procesará lo que le corresponde.
RNF10	Portabilidad	El sistema podrá ser utilizado tanto en Windows como en Linux, esto se debe a que la aplicación está implementada sobre PHP, que es un lenguaje multiplataforma. El producto realizará su función sobre una plataforma Web, por lo que solo se necesita de un navegador en el cliente que interprete el HTML y las funciones básicas del JavaScript, independientemente de cuál sea su sistema operativo.

Tabla #7. Descripción textual de los requisitos no funcionales.

2.6.2 Concepción del sistema.

Este nuevo sistema tiene como misión llevar el control del Pago de todos los colaboradores de la salud. Para ello, la UCCM mantiene una relación estrecha con todos los municipios y provincias del país. Es por eso que se hace necesario conectarse a los diferentes registros ya disponibles en el Registro Informatizado de Salud, para obtener la información procedente de él.

Los registros que se utilizarán son el Registro de Personal de la Salud (RPS), el Registro de Ciudadano (RC), el Registro de Localidades (RL), el Registro de Ubicaciones (RU) y el Sistema de Administración (SAAA), los cuales, pueden utilizarse según el tipo de información que se desee obtener para poder disponer de la misma.

El sistema debe mantener actualizada la información referente al pago del cooperante, mantenerse con una Base de Datos centralizada, única y con niveles de seguridad. Debe permitir que se pueda emitir una nómina organizada por provincias, donde se puede llevar un mayor control de los colaboradores que deben cobrar cada mes. Al establecerse la comunicación a través de la aplicación Web, permite visualizar la información del pago desde diferentes computadoras sin necesidad de radicar o de estar en la misma intranet y en los reportes al organismo superior son más completos.

2.6.3 Actores del sistema.

Cada trabajador del negocio que tiene actividades a automatizar es un candidato a actor del sistema, o si algún actor del negocio va a interactuar con el sistema, entonces también será un actor del sistema. Los actores del sistema pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado; son parte del sistema; pueden intercambiar información con él y ser un recipiente pasivo de información. [57]

Trabajadores.	Justificación.
Editor de la UPCM.	Encargado de gestionar los reportes del saldo de la cuenta especial y las

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

	personalizaciones de los colaboradores, además de visualizar toda la información referente al colaborador.
Editor de la UCCM.	Encargado de gestionar el pago de ayuda familiar de cada colaborador; de realizarle ajustes tanto a la cuenta especial como a la cuenta de ayuda familiar del colaborador, así como de gestionar los registros históricos de cada cuenta; además de visualizar toda la información referente al colaborador.
Visualizador.	Los visualizadores son especialistas en cada uno de los niveles que tendrán el privilegio de obtener información del sistema correspondiente con el nivel al que pertenezca (nivel nacional, nivel municipal, nivel provincial).
Registro Personal de Salud (RPS).	Encargado de brindar información de todo el personal que pertenezca a la salud en Cuba.
Registro de Ciudadano (RC).	Encargado de brindar información de todos los ciudadanos de nuestro país.
Sistema de Administración (SAAA).	Encargado de mantener la seguridad, la integridad y el control del sistema.
Registro de Ubicaciones (RL).	Encargado de brindar información sobre las provincias, municipios, localidades, manzanas y calles de un colaborador.

Tabla #9. Descripción textual de los actores del sistema.

2.6.4 Diagrama de Casos de Uso del Sistema (DCUS).

Representa gráficamente a los procesos y su interacción con los actores.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

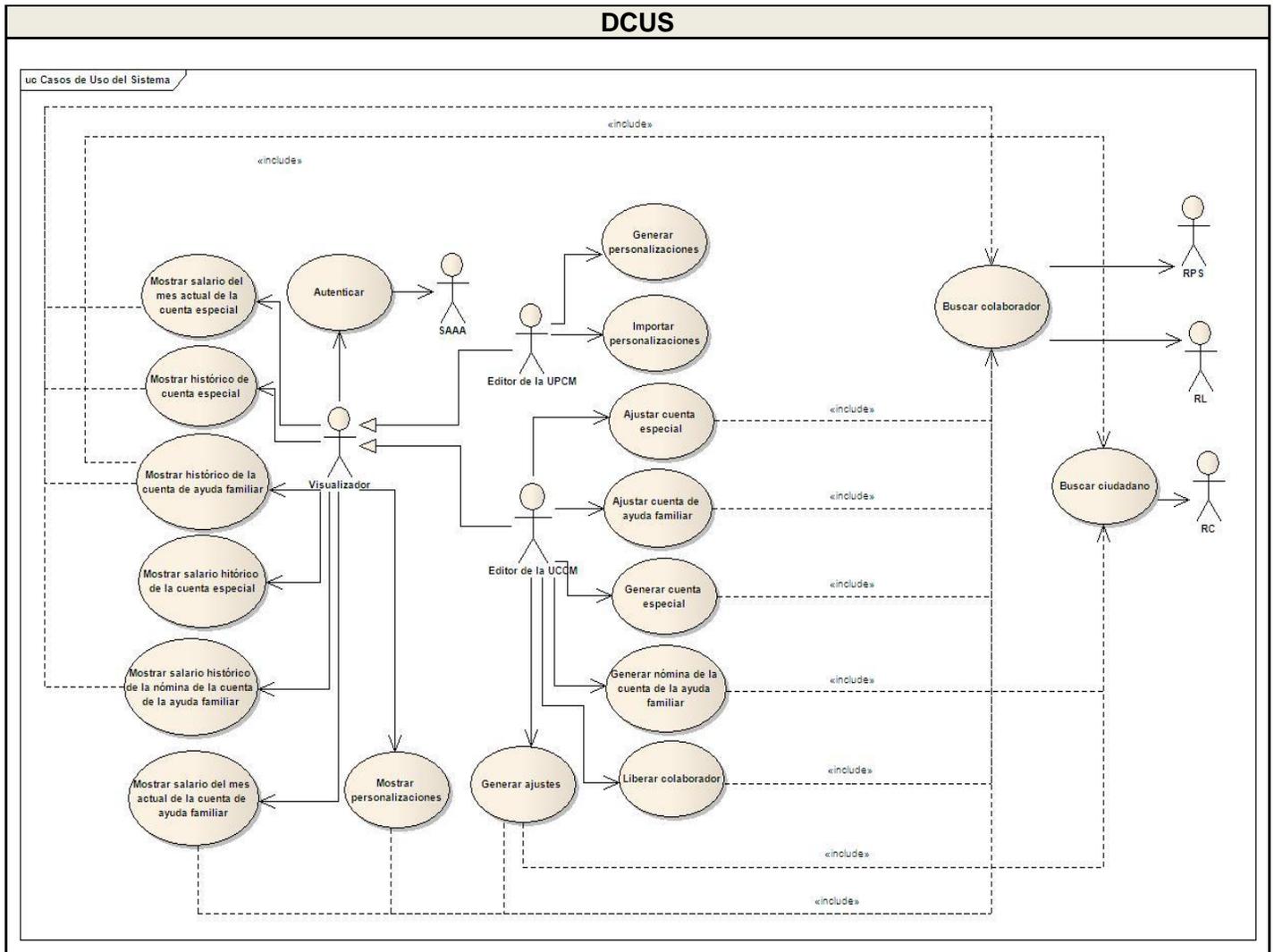


Figura #3: Diagrama de CUS.

2.6.5 Descripción extendida de los CUS.

CUS "Autenticar".

Objetivo:	Conocer los datos de la persona autenticada para validar sus permisos.
Actores:	Visualizador.
Resumen:	El caso de uso se inicia cuando el usuario requiere de acceso al sistema, permite que solamente puedan acceder a la aplicación todos aquellos que trabajaran directamente con la misma. Dándole acceso a operar en dicha aplicación.
Complejidad:	Baja.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

Prioridad:	Secundario.
Pre-condiciones:	El usuario debe tener los permisos necesarios para poder acceder a la aplicación, y la conectividad al SAAA debe ser fluida.
Post-condiciones:	Queda autenticado el usuario en el sistema.

Tabla #10. Descripción textual del CUS "Autenticar".

CUS "Buscar colaborador".

Objetivo:	Obtener los datos de un colaborador dado los parámetros de búsqueda para realizarle determinadas operaciones.
Actores:	Visualizador.
Resumen:	El caso de uso inicia cuando el actor accede a cualquiera de las opciones brindadas por el sistema, excepto Generar e Importar personalizaciones. El sistema brinda la posibilidad de introducir criterios de búsqueda para localizar al colaborador, el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra al colaborador que cumple con los criterios de búsqueda, el caso de uso termina.
Complejidad:	Alta.
Prioridad:	Crítico.
Pre-condiciones:	La información debe estar actualizada y disponible.
Post-condiciones:	Se buscó colaborador dado criterios.

Tabla #11. Descripción textual del CUS "Buscar colaborador".

CUS "Ajustar cuenta especial".

Objetivo:	Permite ajustar la cuenta especial de los colaboradores.
Actores:	Editor de la UCCM.
Resumen:	El caso de uso inicia cuando el editor de la UCCM accede a la opción Especial que brinda el campo "Ajustes de Cuentas", el sistema brinda la posibilidad de introducir los datos para ajustar la cuenta, el editor de la UCCM introduce los datos que considera necesarios para ajustar la cuenta, el sistema a partir de los datos introducidos ajusta la cuenta, y consultando la entidad Buscar colaborador ajusta la cuenta especial, el caso de uso termina.
Complejidad:	Alta.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

Prioridad:	Crítico.
Pre-condiciones:	Que la cuenta especial del colaborador esté disponible y actualizada.
Post-condiciones:	Se ajustó la cuenta especial del colaborador.

Tabla #12. Descripción textual del CUS "Ajustar cuenta especial".

CUS "Generar ajustes".

Objetivo:	Permite ajustar los archivos de crédito y débito de las cuentas de ayuda familiar y especial de los colaboradores.
Actores:	Editor de la UCCM.
Resumen:	El caso de uso inicia cuando el editor accede a la opción Generar Ajustes que brinda el campo "Ajuste de Cuentas", el sistema brinda la posibilidad de introducir los datos que desea ajustar, el editor introduce los datos que considera necesarios para ajustar el archivo, el sistema a partir de los datos introducidos ajusta el archivo, el caso de uso termina.
Complejidad:	Alta.
Prioridad:	Crítico.
Pre-condiciones:	Que los archivos a ajustar estén disponibles.
Post-condiciones:	Se ajustó el archivo de la opción deseada según los datos introducidos.

Tabla #13. Descripción textual del CUS "Generar ajustes".

CUS "Generar personalizaciones".

Objetivo:	Permite crear los archivos que serán llevados al banco para crear la cuenta donde se almacena el salario del colaborador.
Actores:	Editor de la UPCM.
Resumen:	El caso de uso inicia cuando el editor accede a la opción Generar que brinda el campo "Personalizaciones", el sistema brinda la posibilidad de introducir los criterios para generar personalizaciones, el editor introduce los datos que considera como criterios para generar la personalización, el sistema a partir de los criterios seleccionados y consultando las entidades Buscar colaborador y Mostrar personalización, genera la personalización, el caso de uso termina.
Complejidad:	Alta.
Prioridad:	Crítico.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

Pre-condiciones:	Que todos los datos estén correctamente actualizados y sin errores.
Post-condiciones:	Se generó la personalización dado criterios.

Tabla #14. Descripción textual del CUS “Generar personalizaciones”.

CUS “Generar cuenta especial”.

Objetivo:	Permite generar la cuenta especial con los datos del colaborador y su importe.
Actores:	Editor de la UCCM.
Resumen:	El caso de uso inicia cuando el editor de la UCCM accede a la opción Generar de la cuenta especial, el sistema brinda la posibilidad de introducir los criterios para generar la cuenta, el editor de la UCCM introduce los datos que considera como criterios para generar la cuenta, el sistema a partir de los criterios seleccionados y consultando la entidad de Buscar colaborador, genera la información de la cuenta, el caso de uso termina.
Complejidad:	Alta.
Prioridad:	Crítico.
Pre-condiciones:	Que todos los datos estén correctamente actualizados y sin errores.
Post-condiciones:	Se generó la cuenta especial dado los criterios.

Tabla #15. Descripción textual del CUS “Generar cuenta especial”.

CUS “Generar nómina de la ayuda familiar”.

Objetivo:	Permite generar la nómina con los datos del colaborador y su importe.
Actores:	Editor de la UCCM.
Resumen:	El caso de uso inicia cuando el editor de la UCCM accede a la opción Generar Nómina de la ayuda familiar, el sistema brinda la posibilidad de introducir los criterios para generar la nómina, el editor de la UCCM introduce los datos que considera como criterios para generar la nómina, el sistema a partir de los criterios seleccionados y consultando las entidades de Buscar colaborador y Buscar ciudadano, genera la información de la nómina, el caso de uso termina.
Complejidad:	Alta.
Prioridad:	Crítico.
Pre-condiciones:	Que todos los datos estén correctamente actualizados y sin errores.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

Post-condiciones:	Se generó la nómina de ayuda familiar dado los criterios.
--------------------------	---

Tabla #16. Descripción textual del CUS “Generar nómina de la ayuda familiar”.

CUS “Liberar colaborador”.

Objetivo:	Permite liberar de las cuentas de los colaboradores al realizar la liberación de los mismos.
Actores:	Editor de la UCCM.
Resumen:	El caso de uso inicia cuando el editor de la UCCM desea liberar a los colaboradores, este especialista accede a la opción Liberaciones, el cual permite liberar por fin de misión o por vacaciones, el sistema libera al(los) colaborador(es) que se encuentren en ese estado, el caso de uso termina.
Complejidad:	Alta.
Prioridad:	Crítico.
Pre-condiciones:	Que la información este correcta y actualizada.
Post-condiciones:	Se realizó la liberación del colaborador.

Tabla #17. Descripción textual del CUS “Liberar colaborador”.

En el presente capítulo, a partir de la descripción de la solución propuesta, se detalló la lógica del negocio y el funcionamiento de cada uno de sus procesos, de este análisis, se deriva que la implementación de un sistema informatizada puede mejorar considerablemente las deficiencias existentes en dichos procesos. A partir del análisis obtenido de los requerimientos funcionales y de la definición de las principales opciones del sistema, se determinó que la aplicación propuesta sería la forma más óptima de darle solución al problema a resolver planteado.

Capítulo 3. Análisis y Diseño del Sistema.

Este capítulo profundiza en el análisis y diseño del sistema; tiene como objetivo fundamental transformar los requerimientos, en un diseño que muestre cómo va a ser implementado el sistema. Describe a partir del análisis y diseño qué va a contener y gestionar el sistema y cómo este va a realizar la gestión a partir de las herramientas propuestas. En este capítulo además, se realizará la propuesta de la interfaz de usuario del software.

Durante el diseño, se toman decisiones estratégicas y tácticas para cumplir los requerimientos funcionales y de calidad de un sistema. Es la parte del proceso de desarrollo de software cuyo propósito primario es decidir cómo el sistema se llevará a cabo.

3.1 Descripción de la arquitectura. Fundamentación.

La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema. Una Arquitectura de Software, también denominada Arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información.

El sistema que se propone presenta una arquitectura basada en uno de los estilos más utilizados del patrón arquitectónico MVC, expuesto en el Capítulo 1.

El uso del framework Symfony permite una buena utilización del patrón MVC, brindándole a la aplicación la posibilidad de tener una separación clara entre cómo se muestra la información al usuario, cómo se manejan las acciones que el usuario desea hacer sobre el sistema y cómo se realizan estas acciones modificando y validando la información. Además, utiliza otros dos patrones de diseño de gran importancia, los cuales son:

- El Factory: Consiste en la definición de una clase que realiza una determinada tarea. Symfony utiliza las factorías en su funcionamiento interno, como por ejemplo, para los controladores y las sesiones. Cuando el framework necesita crear un nuevo objeto para una petición, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea. La principal ventaja de utilizar las definiciones de las factorías es que se hace muy sencillo modificar las características internas.
- El Decorator: Es la unión del archivo llamado layout.php con las plantillas, donde el primero almacena el código HTML que es común para todas las páginas de la aplicación, y el segundo contiene los formularios y datos específicos de cada interfaz. El resultado de la aplicación de este

patrón de diseño puede estar integrado también por parciales, slots, componentes y slots de componentes en caso de existir.

Otros patrones utilizados fueron los llamados Patrones Generales de Software para Asignar Responsabilidades (GRASP, por sus siglas en inglés), que tuvieron una importante utilidad en el diseño realizado. A cada clase le fueron asignadas las tareas que podían realizar según la información que poseían. Además se crearon las instancias de otras clases en correspondencia con la responsabilidad dada, poniéndose de manifiesto los patrones Experto y Creador. De esta manera los objetos logran valerse de la información que contienen para realizar las tareas que se les pide, además de utilizar el patrón Creador para definir quién será el responsable de crear una nueva instancia de una clase. Este diseño obtenido cumple con los patrones de Bajo acoplamiento y de Alta cohesión facilitando así la centralización de actividades de cada elemento, y logrando que estos realicen una única tarea dentro del sistema. [58]

3.2 Análisis de posibles implementaciones, componentes o módulos ya existentes que puedan ser rehusados. Estrategias de integración.

3.2.1 Componentes reutilizados.

El SISaludPlugin es un componente desarrollado por el Departamento Sistemas Especializados de la facultad siete de la UCI con el objetivo de añadir al framework Symfony la funcionalidad de autenticación mediante el SAAA, este plugin es muy específico para sistemas que necesiten de este tipo de autenticación, se encuentra en su fase de desarrollo inicial v0.1.

Se decide la utilización de este componente porque brinda todas las funcionalidades y la seguridad necesarias en el sistema en cuanto a la asignación de permisos a los futuros usuarios, además posee un alto grado de configuración y es fácil de utilizar.

3.2.2 Estrategia de integración.

El sistema en cuestión tiene la particularidad de consumir información gestionada por los diferentes servicios del Registro Informatizado para la Salud (RIS). Esta comunicación se realiza mediante el protocolo SOAP para la prestación de servicios Web, a continuación se especifican las funcionalidades de las cuales se obtiene información:

- SAAA: Sistema complementario utilizado para la autenticación del usuario, obtiene todos los permisos necesarios para acceder a las funcionalidades del sistema y a los restantes servicios Web.

- RC: Registro que ofrece la búsqueda total de ciudadanos, recibe un arreglo de parámetros con los datos del individuo que se desea buscar y retorna un listado con las personas que coinciden con la información suministrada.
- RL: Registro de localidades que permite obtener toda la información referente a las provincias, municipios, localidades, manzanas y calles en dependencia de la profundidad de la petición.
- RPS: Registro que ofrece la búsqueda de profesionales de la Salud, el mismo recibe un arreglo con los datos del personal de salud que se desea buscar y retorna un listado de las personas que coinciden con la información entrada por el usuario. Permite además listar nomencladores especiales, o sea que, retorna los datos con los codificadores de tipo de profesionales y sus respectivas especialidades.

3.3 Análisis.

3.3.1 Modelo de Análisis.

Proporciona la estructura a la vista interna del sistema, estructurado por clases y paquetes estereotipados. Define realizaciones de casos de uso, y cada una de ellas representa el análisis de un caso de uso del modelo de casos de uso. [59]

3.3.2 Clases de análisis.

Se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Tienen atributos y entre ellas se establecen relaciones de asociación, agregación/composición, generalización/especialización y tipos asociativos. Las clases del análisis están siempre identificadas con uno de los tres estereotipos existentes, los cuales son: [60]

- Interfaz: Se encarga de la modelación de toda la interacción que puede existir entre los actores y el sistema; constituye las fronteras del mismo.
- Control: Representa la coordinación, secuenciación, transacciones y a veces la lógica del negocio; se emplean a menudo para encapsular el control referido a un caso de uso.
- Entidad: Representa la información de larga duración y a menudo persistente que se maneja en el sistema.

3.3.3 Diagrama de clase del análisis.

Es un artefacto en el que se representan los conceptos en un dominio del problema. Representa el funcionamiento del mundo real, no de la implementación automatizada del mismo. [61]

CUS “Autenticar”.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA.

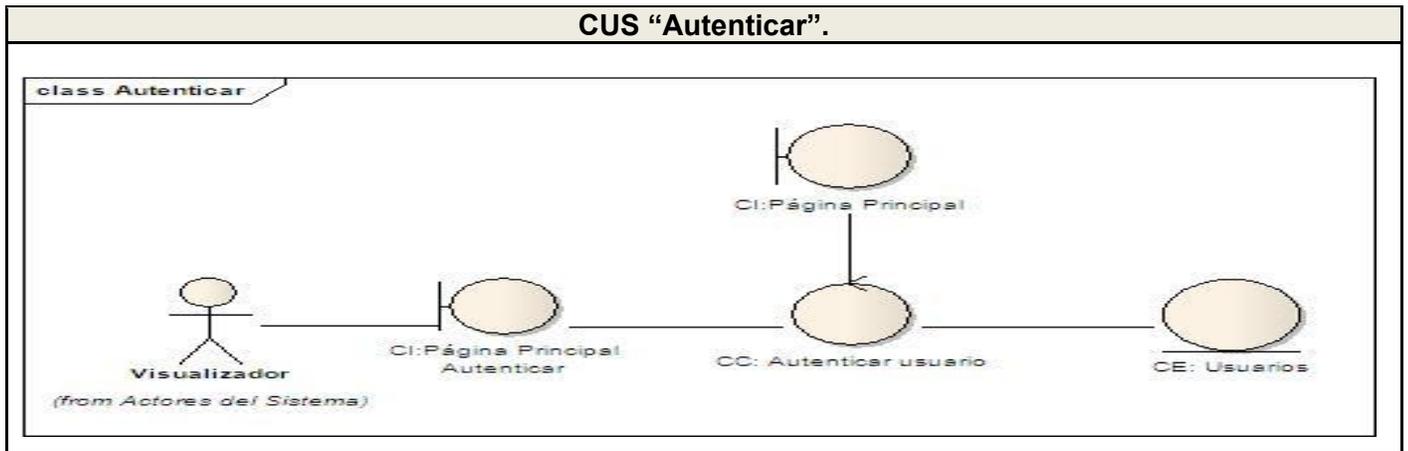


Figura #4: Diagrama de clases del análisis del CUS "Autenticar".

CUS "Buscar colaborador".

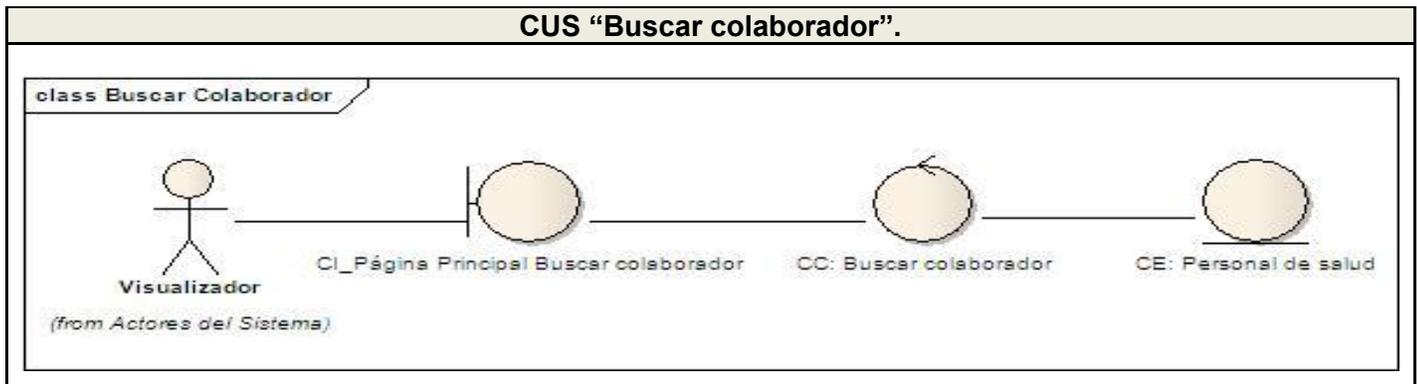


Figura #5: Diagrama de clases del análisis del CUS "Buscar colaborador".

CUS "Ajustar cuenta especial".

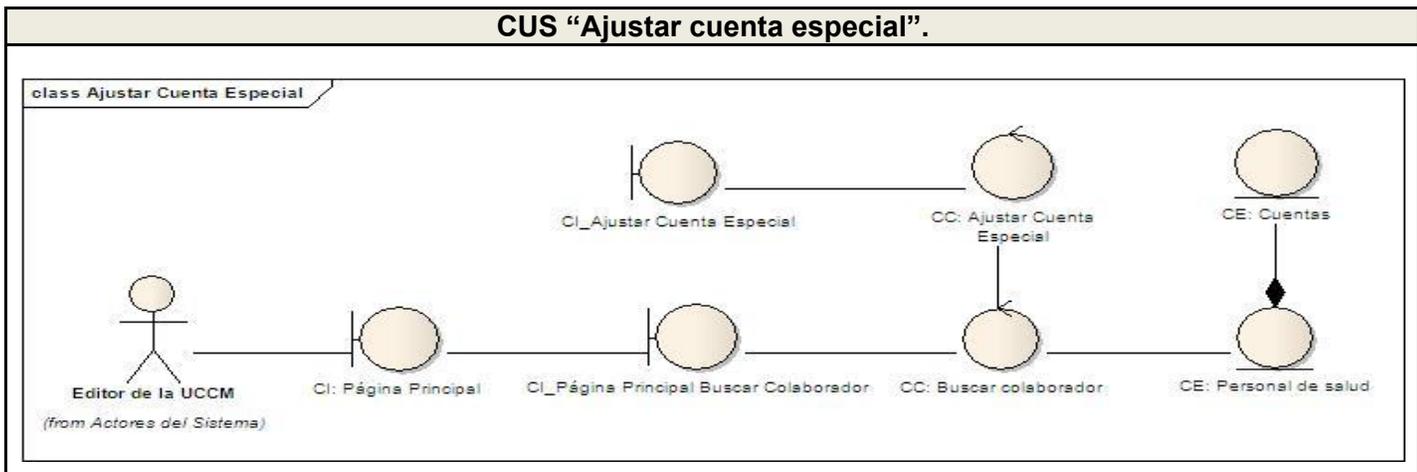


Figura #6: Diagrama de clases del análisis del CUS "Ajustar cuenta especial".

CUS “Generar Ajustes”.

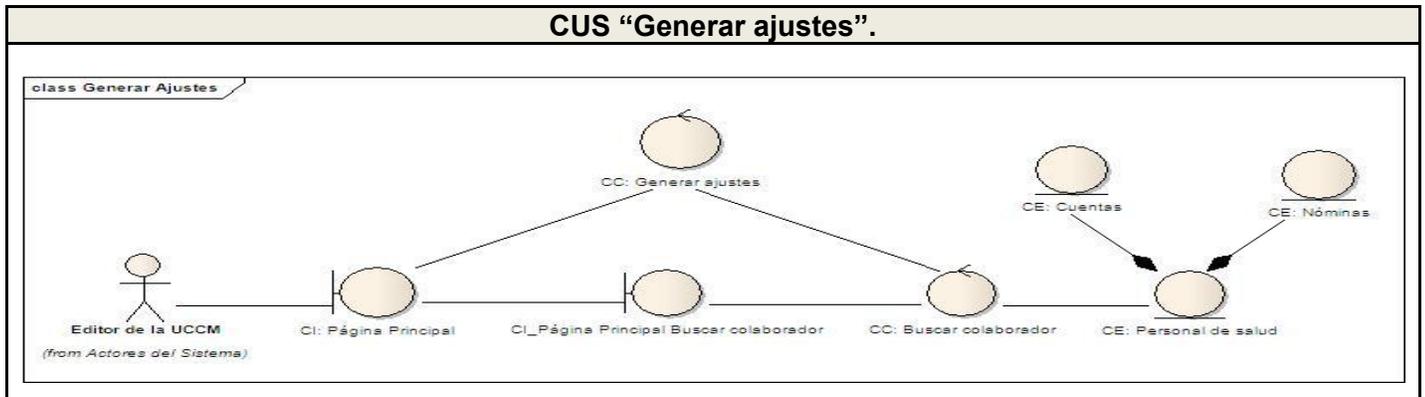


Figura #7: Diagrama de clases del análisis del CUS “Generar ajustes”.

CUS “Generar Personalizaciones”.

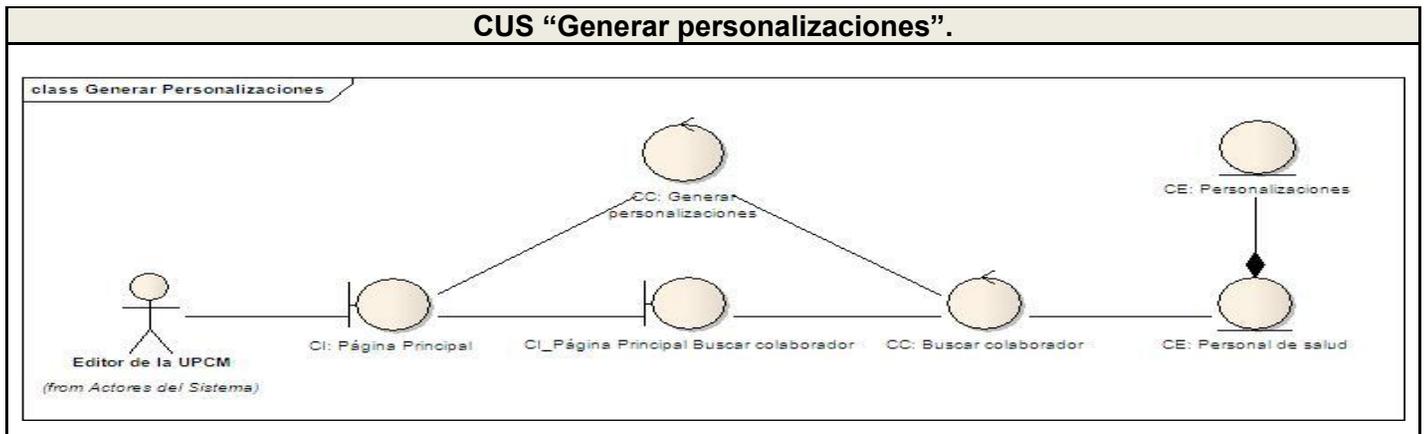


Figura #8: Diagrama de clases del análisis del CUS “Generar personalizaciones”.

CUS “Generar cuenta especial”.

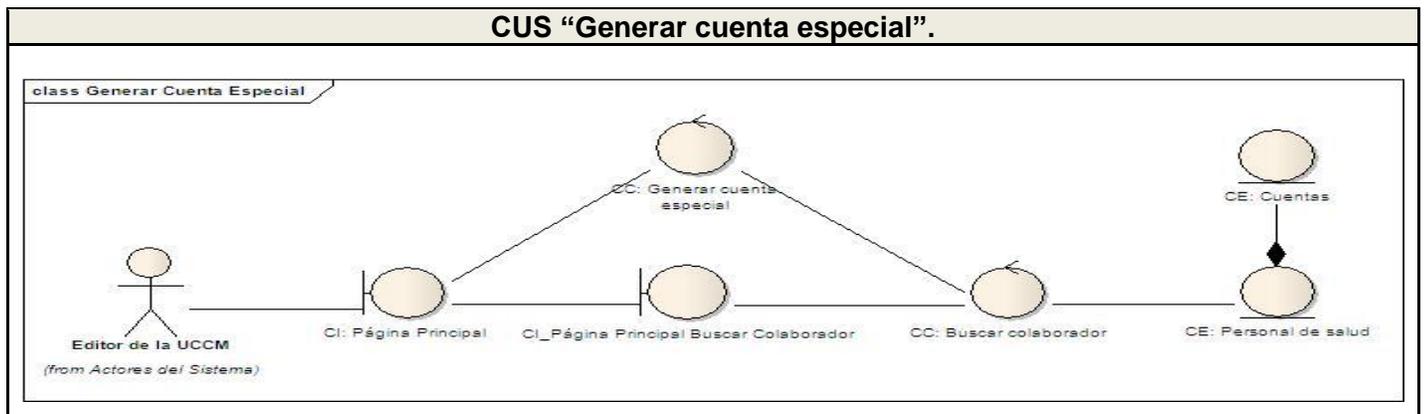


Figura #9: Diagrama de clases del análisis del CUS “Generar cuenta especial”.

CUS “Generar nómina de la cuenta ayuda familiar”.

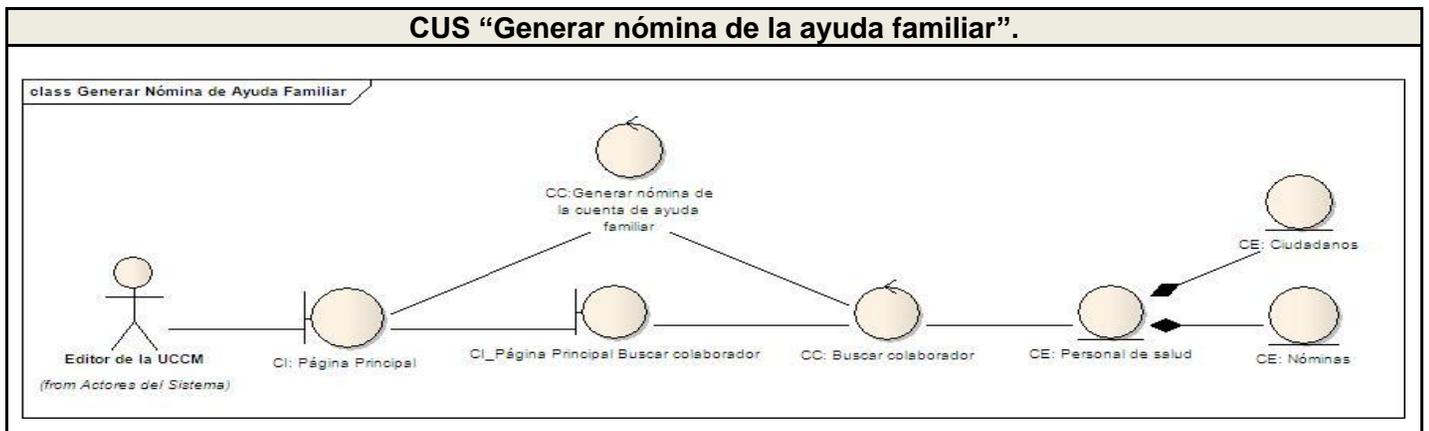


Figura #10: Diagrama de clases del análisis del CUS “Generar la nómina de la cuenta de ayuda familiar”.

CUS “Liberar colaborador”.

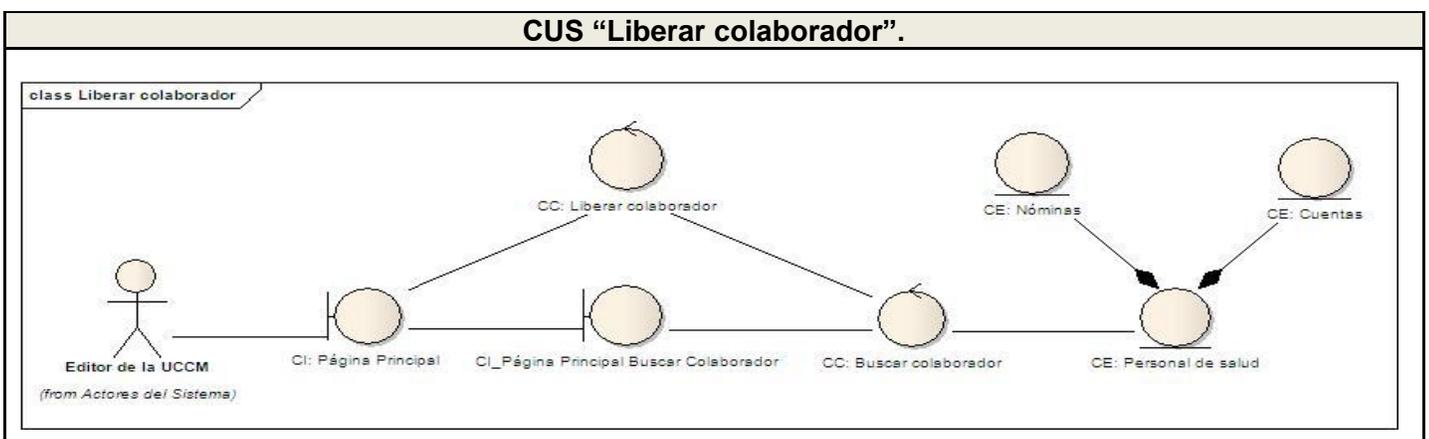


Figura #11: Diagrama de clases del análisis del CUS “Liberar colaborador”.

3.4 Diseño.

3.4.1 Modelo de diseño.

Mediante el modelo de diseño se hace un refinamiento del proceso de análisis anteriormente realizado. Para ello se tienen en cuenta los requisitos no funcionales del sistema ya que el principal propósito del modelado del diseño es crear un plano del modelo de implementación. También se define la arquitectura del sistema. Los casos de uso son realizados por las clases del diseño y sus objetos, a partir de los cuales se forma el diagrama de clases del diseño. [62]

3.4.2 Diagrama de clases de diseño del sistema.

Describen gráficamente las especificaciones de las clases del software, los elementos básicos que podemos encontrar son las clases y sus relaciones. [63]

CUS “Autenticar”.

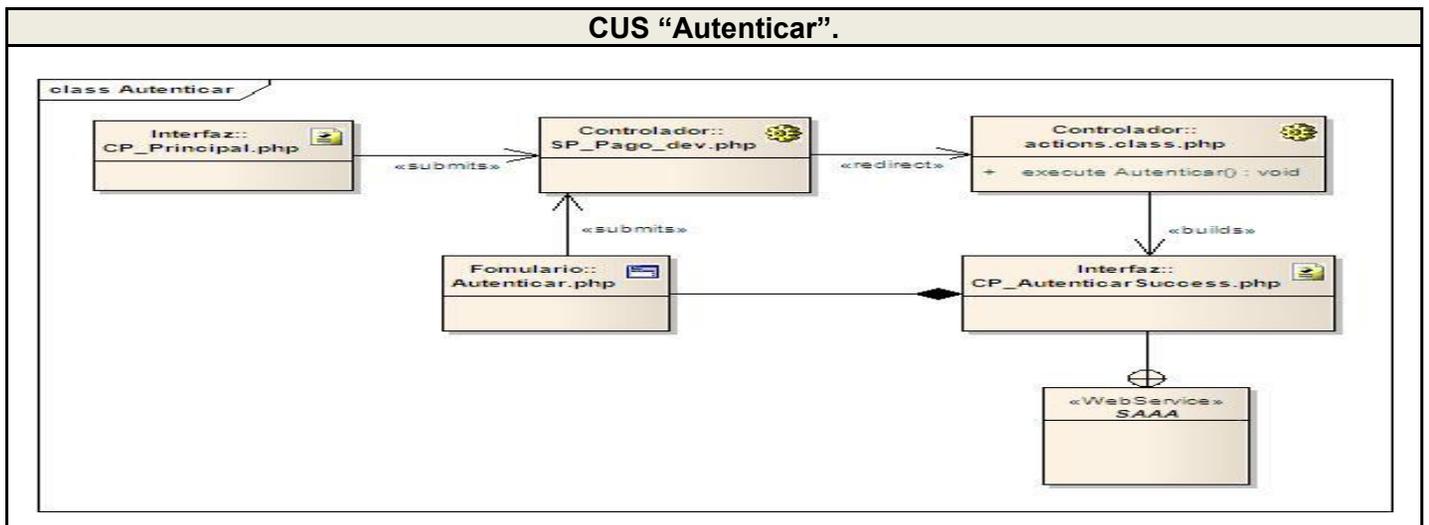


Figura #12: Diagrama de clases del diseño del CUS “Autenticar”.

CUS “Buscar colaborador”.

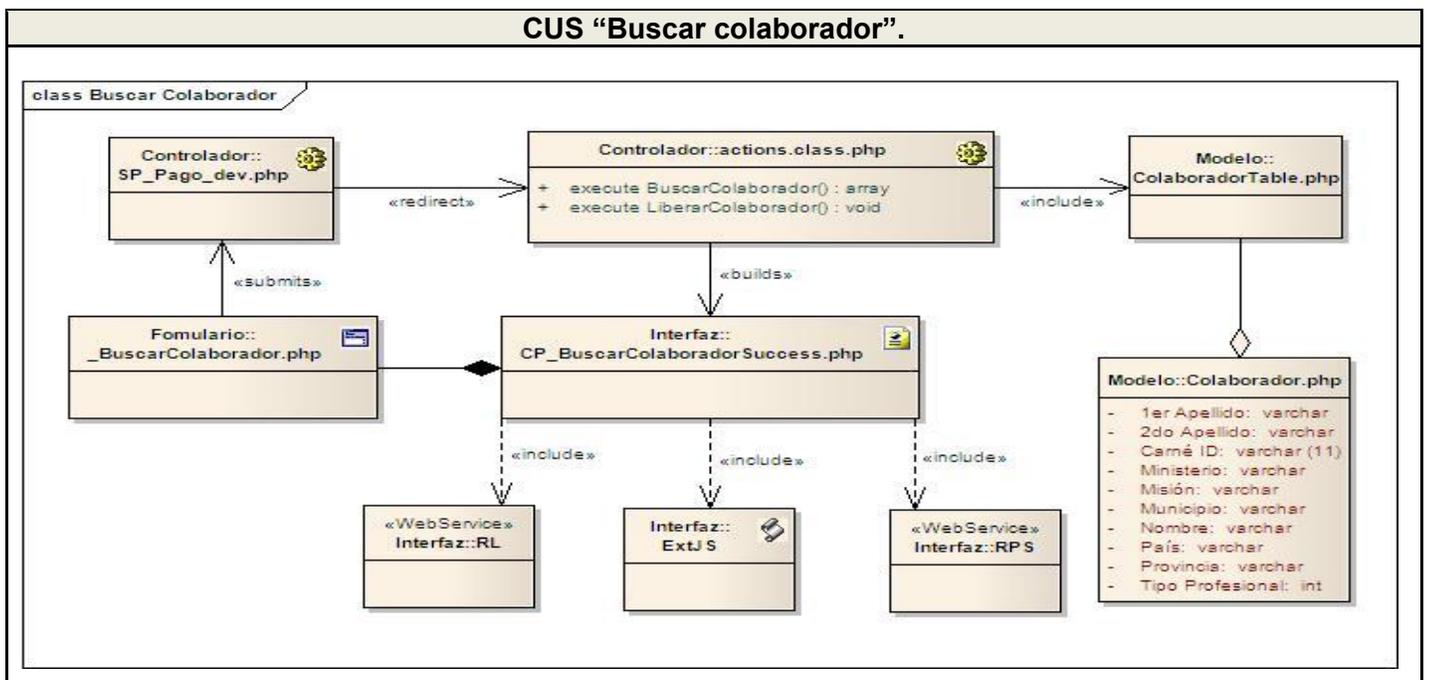


Figura #13: Diagrama de clases del diseño del CUS “Buscar colaborador”.

CUS “Ajustar cuenta especial”.

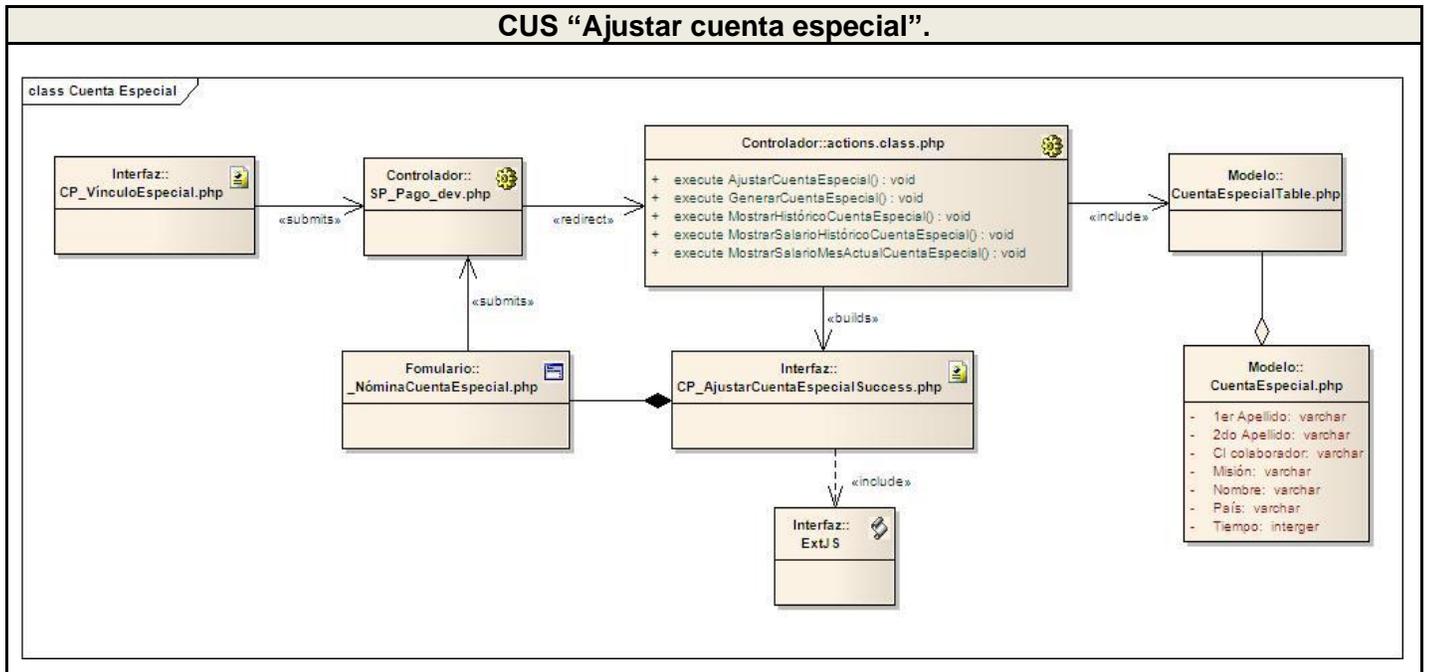


Figura #14: Diagrama de clases del diseño del CUS “Ajustar cuenta especial”.

CUS “Generar ajustes”.

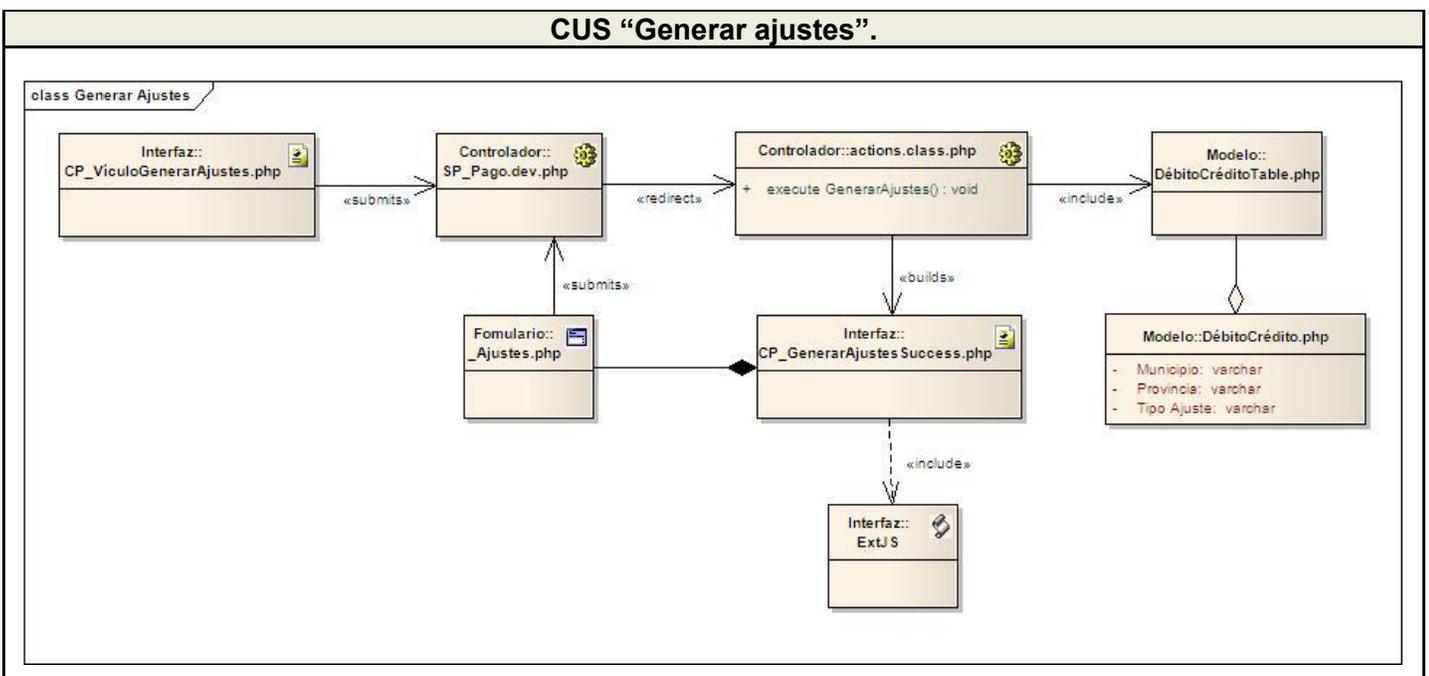


Figura #15: Diagrama de clases del diseño del CUS “Generar ajustes”.

CUS “Generar personalizaciones”.

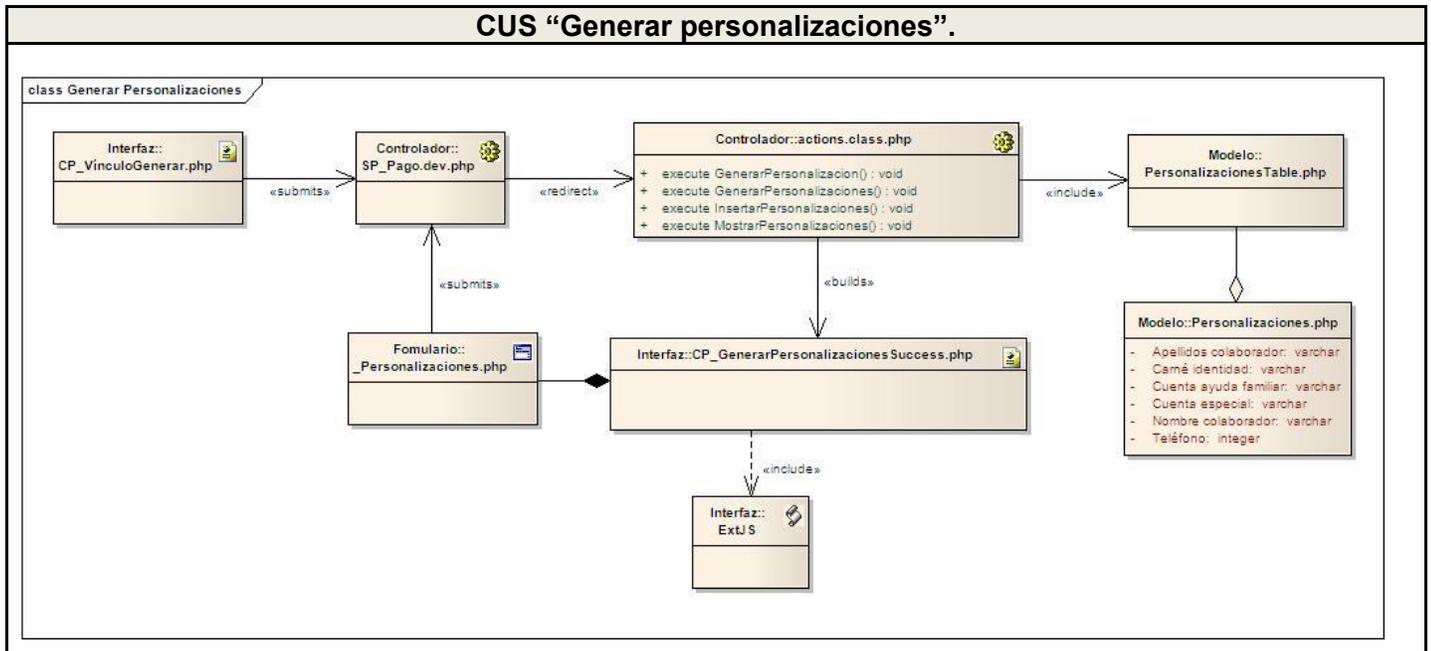


Figura #16: Diagrama de clases del diseño del CUS “Generar personalizaciones”.

CUS “Generar cuenta especial”.

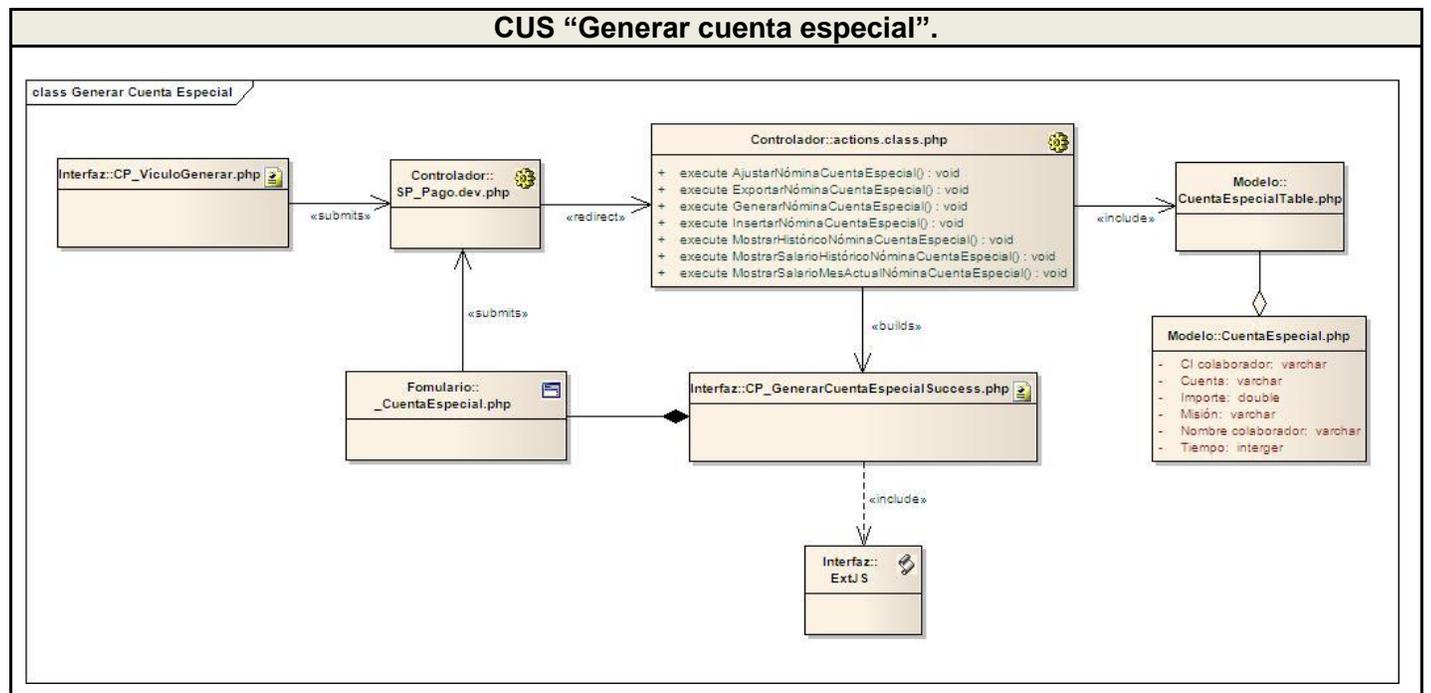


Figura #17: Diagrama de clases del diseño del CUS “Generar cuenta especial”.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA.

CUS “Generar nómina de la cuenta de ayuda familiar”.

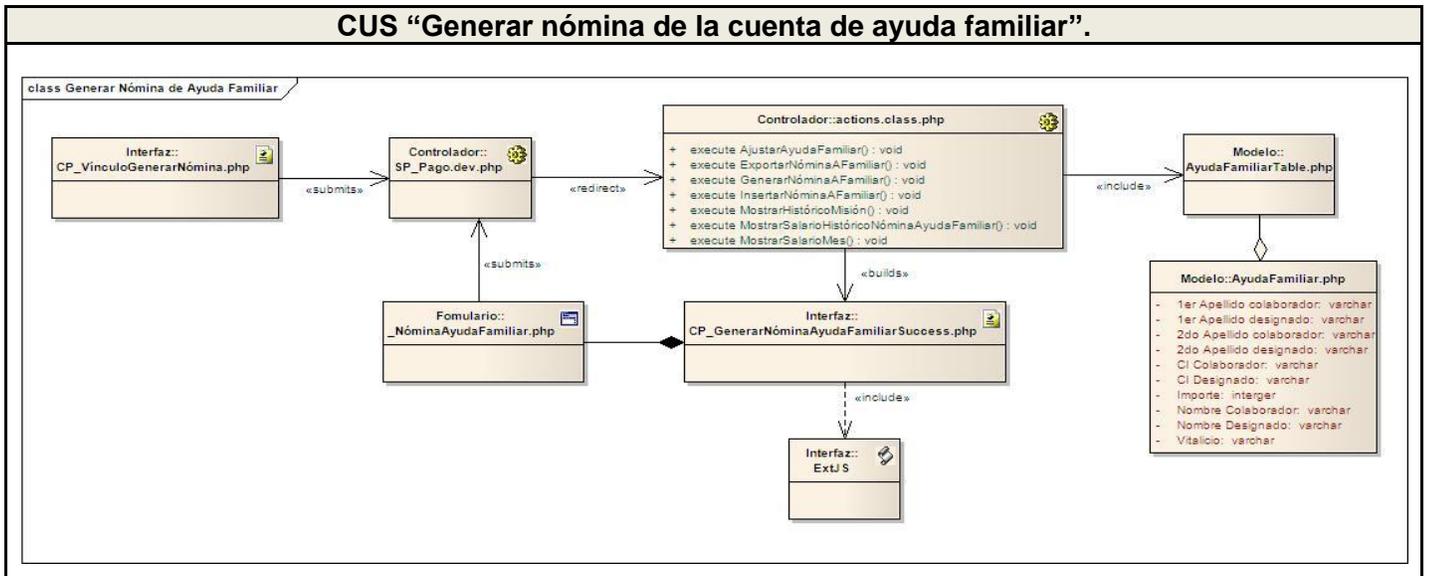


Figura #18: Diagrama de clases del diseño del CUS “Generar nómina de la cuenta de ayuda familiar”.

CUS “Liberar colaborador”.

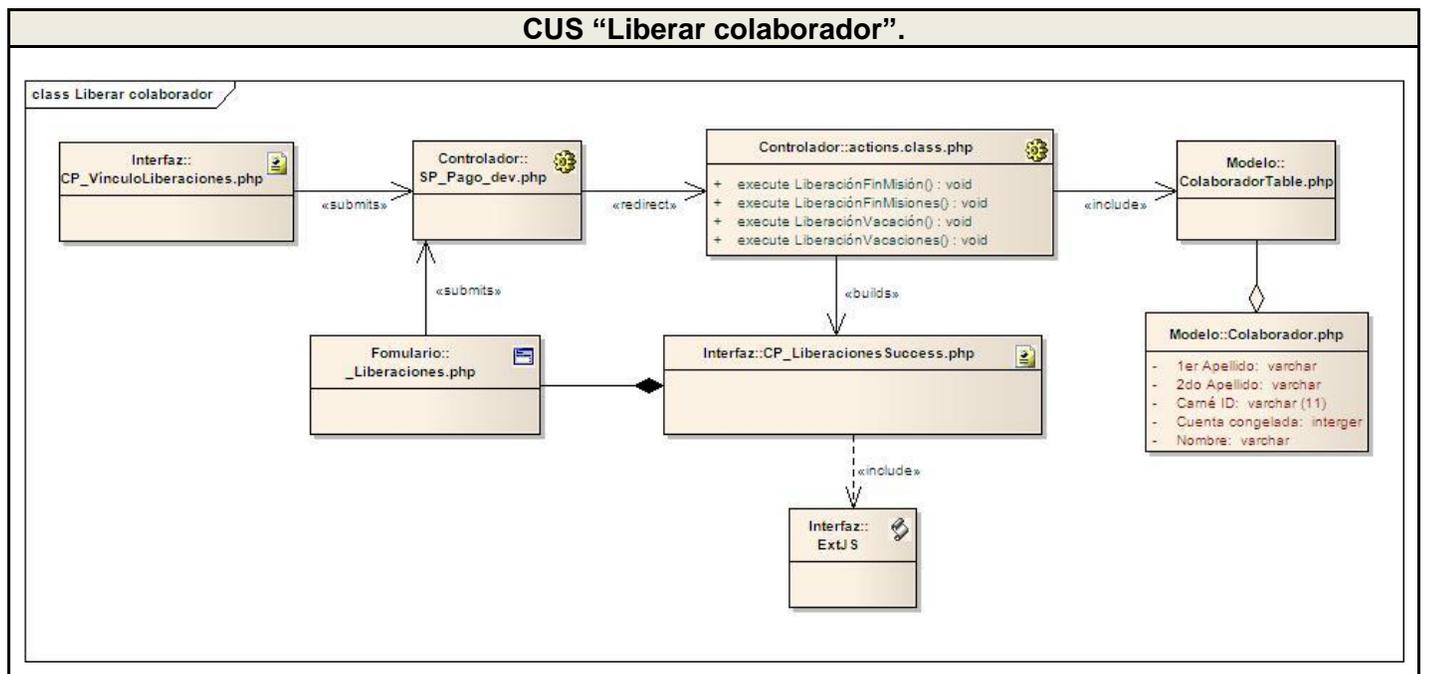


Figura #19: Diagrama de clases del diseño del CUS “Liberar colaborador”.

3.4.3 Diagrama de interacción.

Se trata de un término genérico que se aplica a varios tipos de diagramas que hacen hincapié en las interacciones entre objetos, en este caso se le denomina así a los diagramas de secuencia y a los diagramas de colaboración. Ambos diagramas son isomorfos, es decir, se pueden transformar de uno al otro sin pérdida de información. Generalmente se realizan los diagramas de colaboración en el análisis y los de secuencia en el diseño. [64]

3.4.4 Diagrama de secuencia.

Muestra una interacción que está organizada como una secuencia temporal de los mensajes; no muestra los enlaces existentes entre objetos. [65]

CUS "Autenticar".

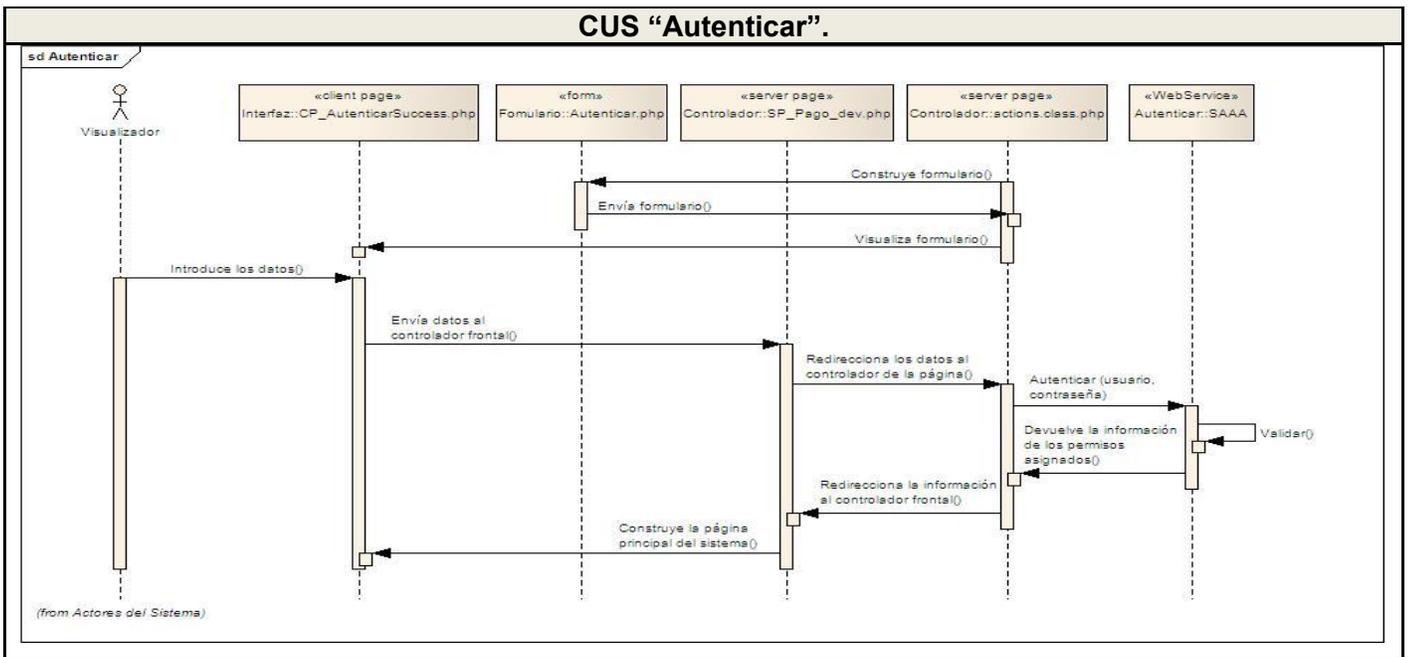


Figura #20: Diagrama de secuencia del CUS "Autenticar".

CUS “Buscar colaborador”.

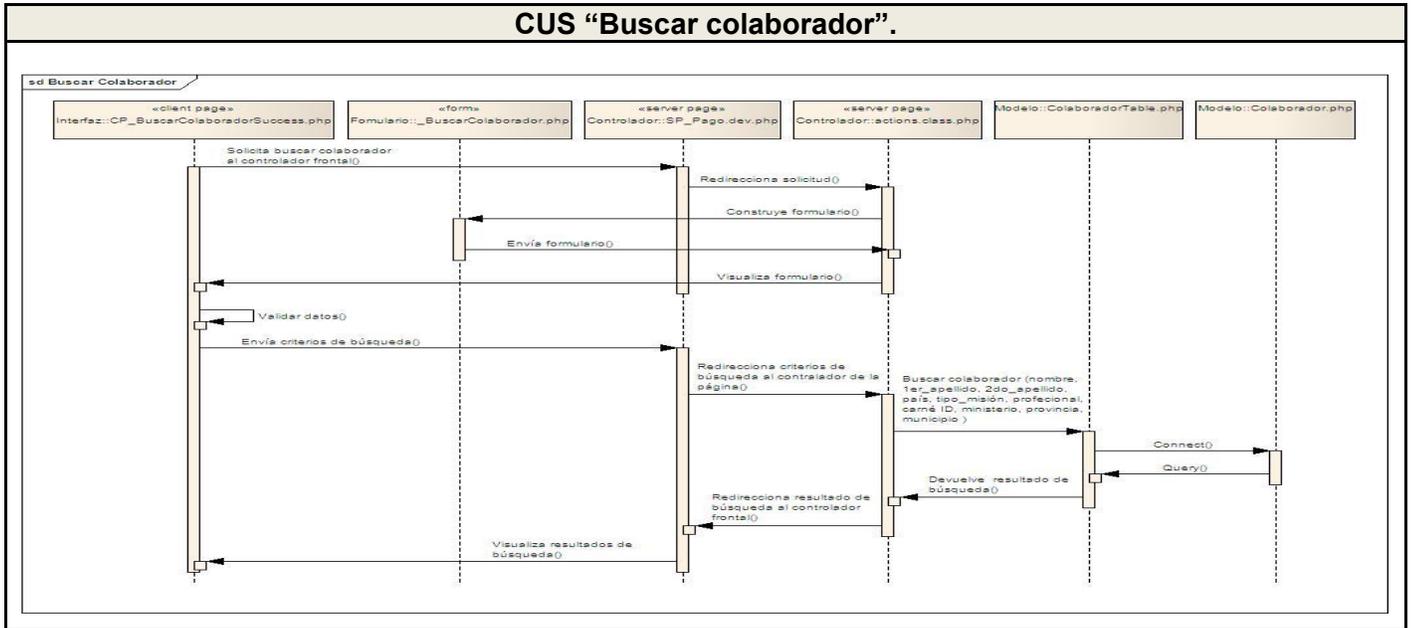


Figura #21: Diagrama de secuencia del CUS “Buscar colaborador”.

CUS “Ajustar cuenta especial”.

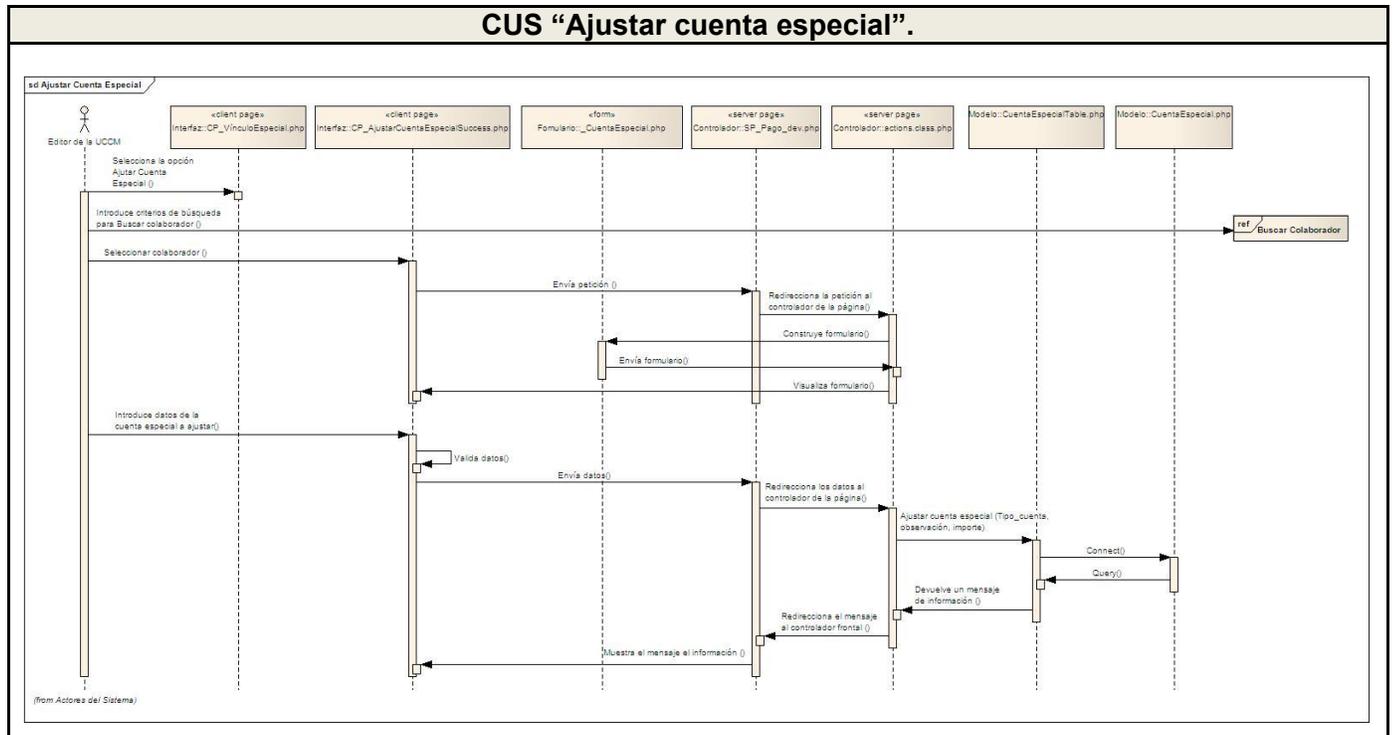


Figura #22: Diagrama de secuencia del CUS “Ajustar cuenta especial”.

CUS “Generar ajustes”.

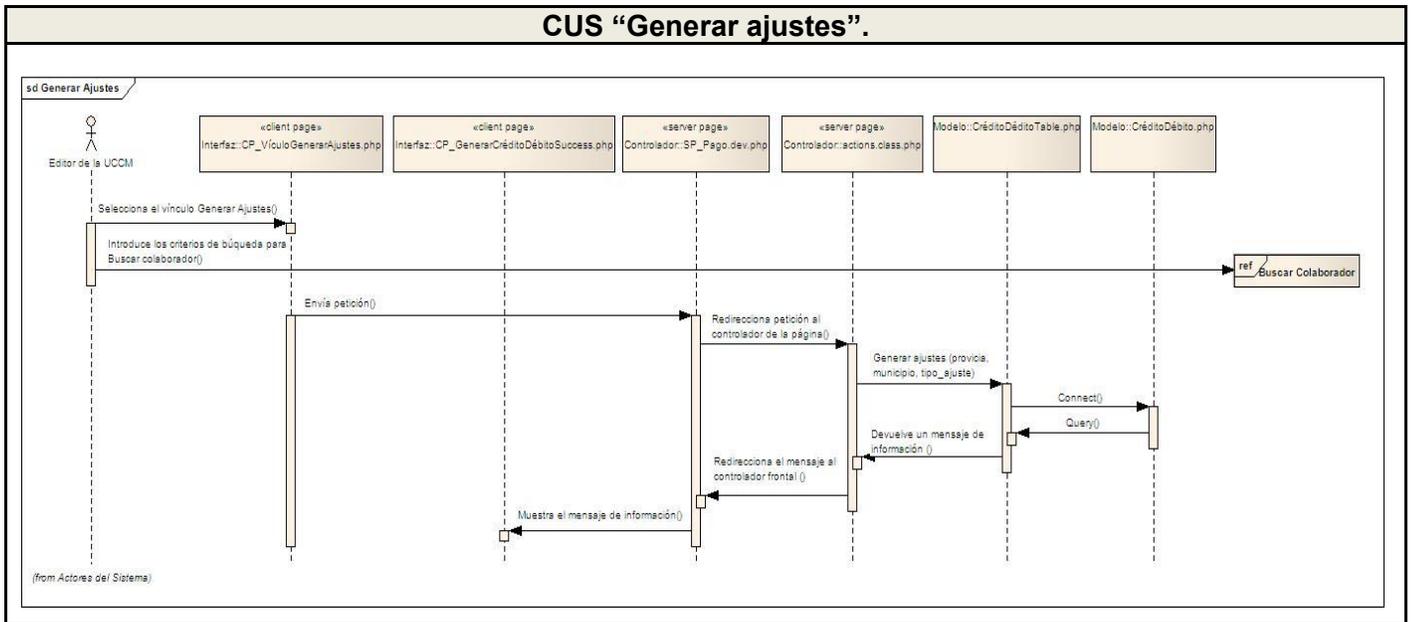


Figura #23: Diagrama de secuencia del CUS “Generar ajustes”.

CUS “Generar personalizaciones”.

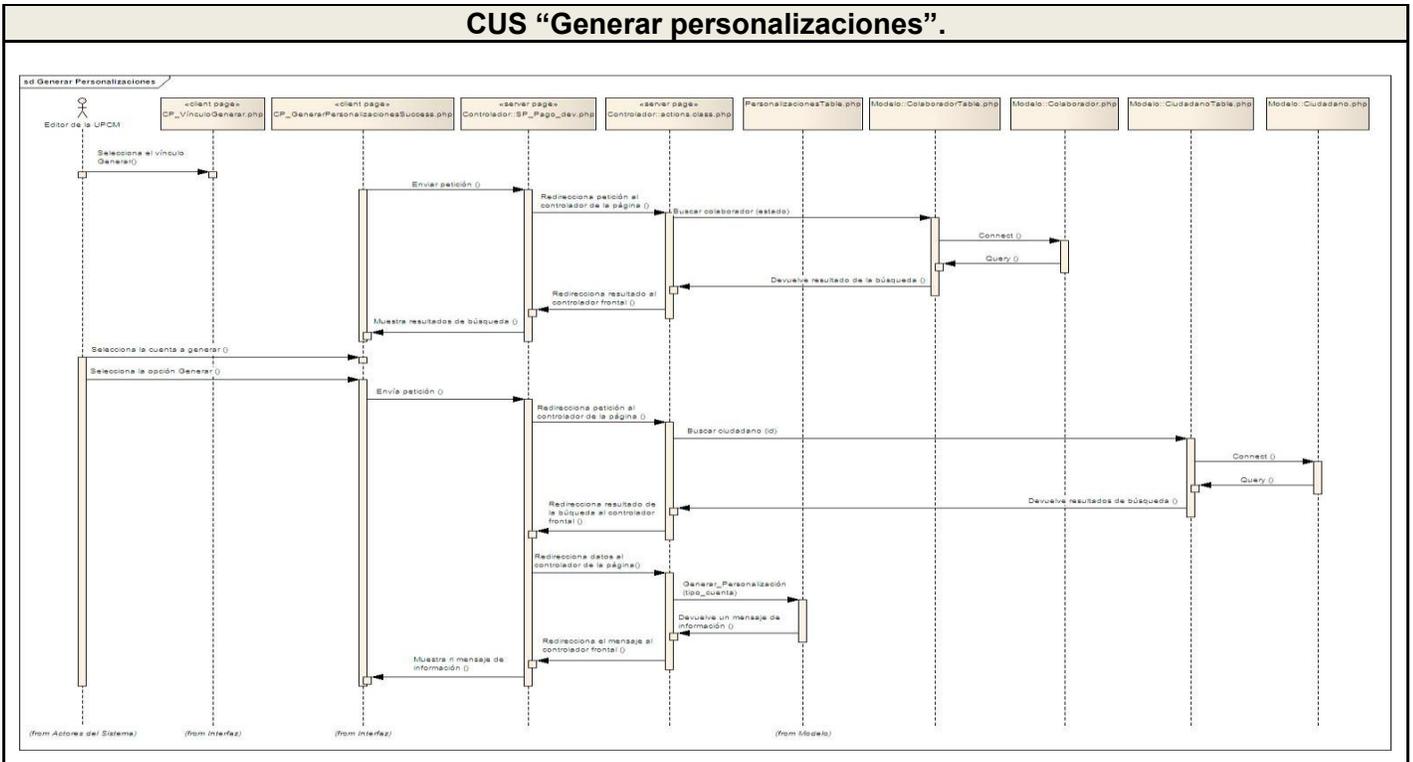


Figura #24: Diagrama de secuencia del CUS “Generar personalizaciones”.

CUS “Generar cuenta especial”.

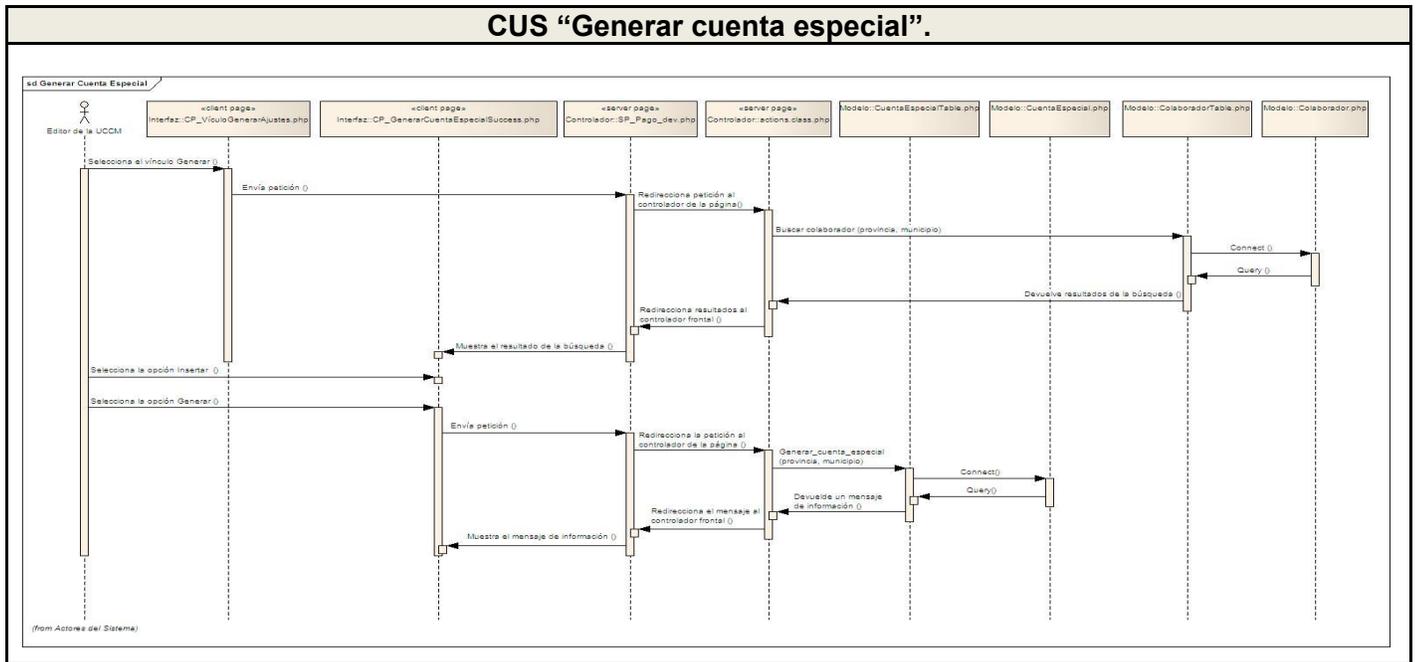


Figura #25: Diagrama de secuencia del CUS “Generar cuenta especial”.

CUS “Generar nómina de la cuenta de ayuda familiar”.

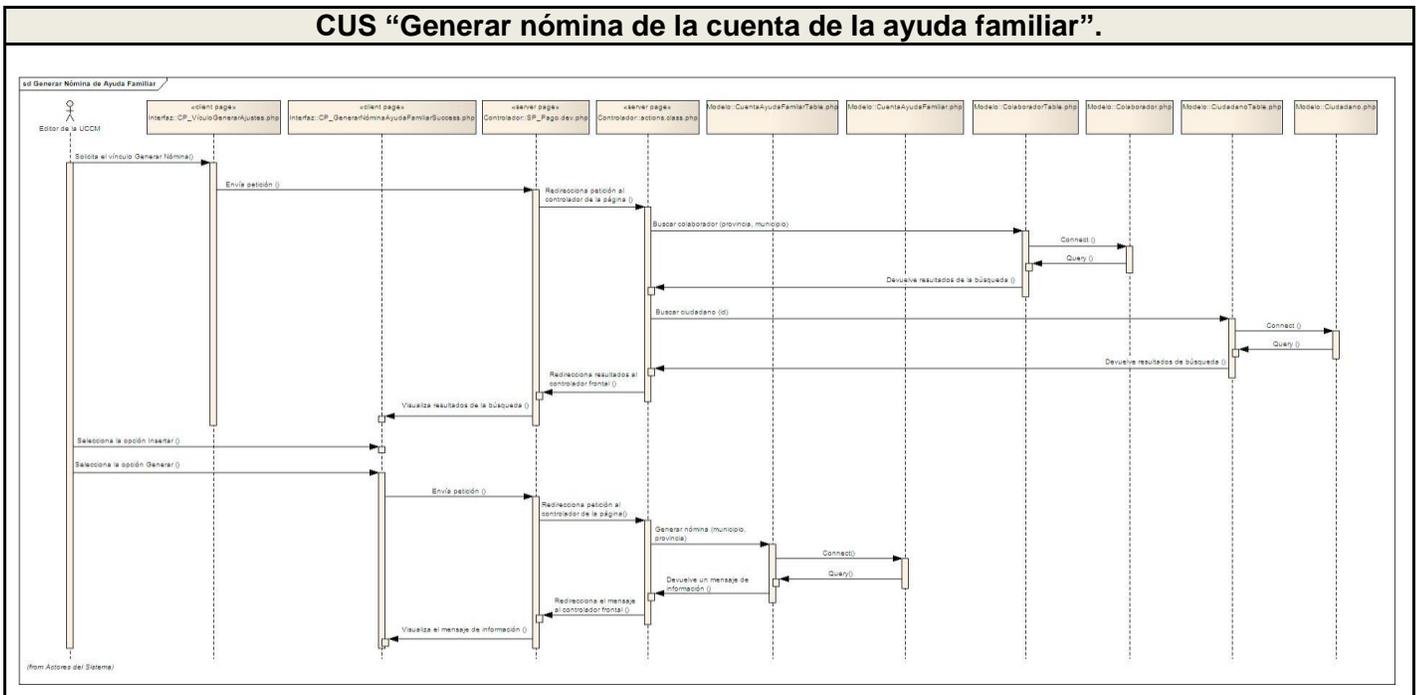


Figura #26: Diagrama de secuencia del CUS “Generar la nómina de la cuenta de ayuda familiar”.

CUS “Liberar colaborador”.

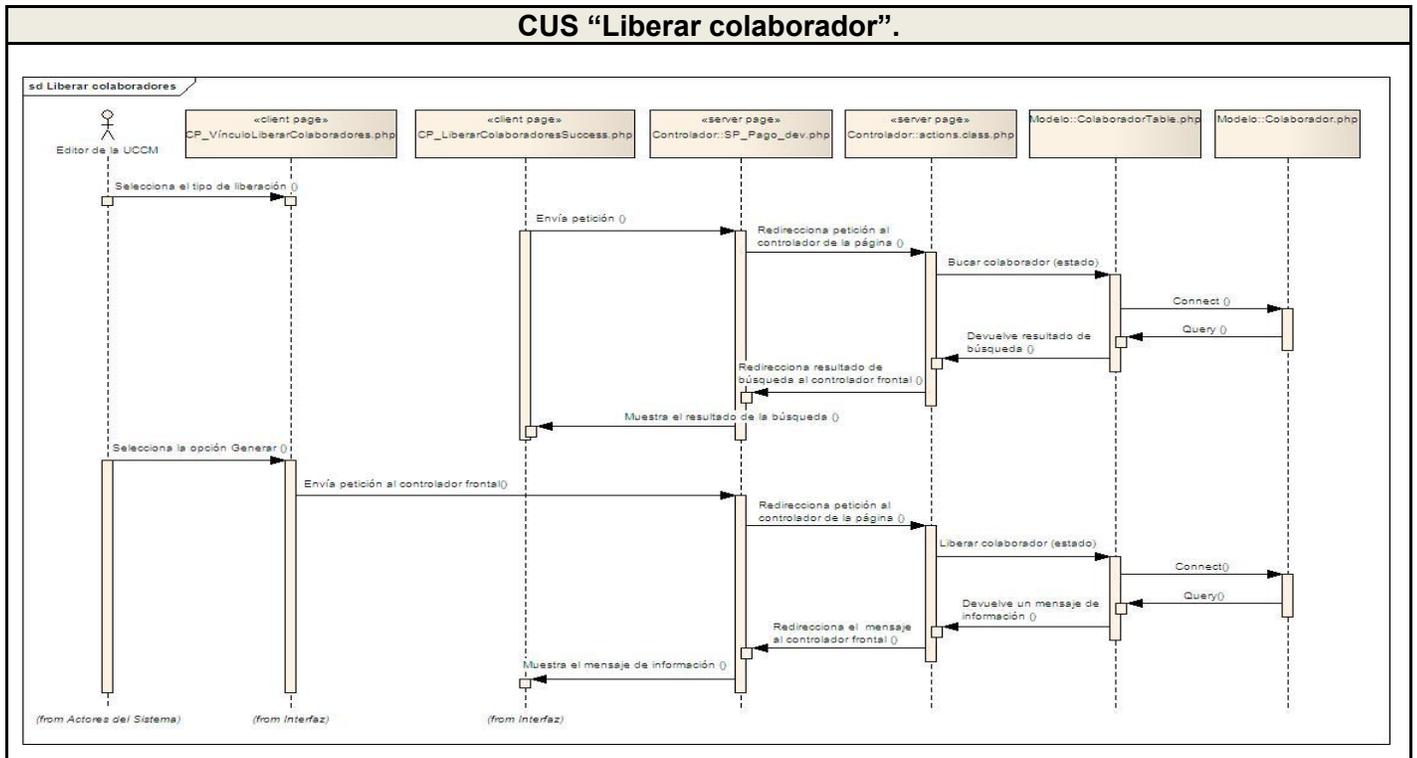


Figura #27: Diagrama de secuencia del CUS “Liberar colaborador”.

3.4.5 Descripción de las clases del diseño.

Clase Autenticar.

Nombre: CP_AutenticarSuccess.php	
Tipo de clase: interfaz	
Atributo	Tipo
Usuario	TextField
Contraseña	TextField
Aceptar	Button
Para cada responsabilidad:	
Nombre:	
Descripción:	
Nombre: actions.class.php	
Tipo de clase: controladora	

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA.

Atributo	Tipo
Para cada responsabilidad:	
Nombre:	executeAutenticar(usuario, contraseña)
Descripción:	Permite el acceso a las funcionalidades del sistema.

Tabla #18. Descripción textual de la clase del diseño del CUS "Autenticar".

Clase Buscar colaborador.

Nombre: CP_BuscarColaboradorSuccess.php	
Tipo de clase: interfaz	
Atributo	Tipo
CI	TextField
Nombre	TextField
Apellido 1	TextField
Apellido 2	TextField
No Registro	TextField
Estado	TextField
Profesional	TextField
Especialidad	TextField
Buscar	Button
Para cada responsabilidad:	
Nombre:	
Descripción:	
Nombre: actions.class.php	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	executeBuscarColaborador(id_colaborador; nombre; 1er_apellido; 2do_apellido; especialidad; tipo_estado; no_registro; profesional.)
Descripción:	Permite realizar la búsqueda de un colaborador dado un identificador.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA.

Tabla #19. Descripción textual de la clase del diseño del CUS “Buscar colaborador”.

Clase Ajustar cuenta especial.

Nombre: CP_AjustarCuentaEspecialSuccess.php	
Tipo de clase: interfaz	
Atributo	Tipo
Débito	RadioButton
Crédito	RadioButton
Importe	TextField
Observaciones	TextArea
Insertar	Button
Para cada responsabilidad:	
Nombre:	
Descripción:	
Nombre: actions.class.php	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	executeAjusteCuentaEspecial()
Descripción:	Permite introducir los cambios necesarios en el crédito o débito de la cuenta especial, así como una observación de lo realizado.
Nombre:	executeGenerarNominaCuentaEspecial()
Descripción:	Permite generar la nómina de la cuenta especial a todos aquellos colaboradores que estén de misión.

Tabla #20. Descripción textual de la clase del diseño del CUS “Ajustar cuenta especial”.

Clase Generar ajustes.

Nombre: CP_GenerarCréditoDébitoSuccess.php	
Tipo de clase: interfaz	
Atributo	Tipo
Tipo Ajuste	Select

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA.

Generar	Button
Para cada responsabilidad:	
Nombre:	
Descripción:	
Nombre: action.class.php	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	executeGenerarCréditoDébito()
Descripción:	Permite generar los ajustes por débito y crédito una vez hayan sido insertados.

Tabla #21. Descripción textual de la clase del diseño del CUS "Generar ajustes".

Clase Generar cuenta especial.

Nombre: CP_GenerarCuentaEspecialSuccess.php	
Tipo de clase: interfaz	
Atributo	Tipo
Aprobar Cuenta	Button
Exportar Cuenta	Button
Para cada responsabilidad:	
Nombre:	
Descripción:	

Tabla #22. Descripción textual de la clase del diseño del CUS "Generar cuenta especial".

Clase Generar nómina de la cuenta de la ayuda familiar.

Nombre: CP_GenerarNominaAyudaFamiliarSuccess.php	
Tipo de clase: interfaz	
Atributo	Tipo
Aprobar Nómina	Button
Exportar Nómina	Button
Para cada responsabilidad:	
Nombre:	

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA.

Descripción:	
--------------	--

Tabla #23. Descripción textual de la clase del diseño del CUS “Generar nómina de la cuenta de ayuda familia”.

Clase Liberar colaborador.

Nombre: CP_LiberarColaboradoresSuccess.php	
Tipo de clase interfaz	
Atributo	Tipo
Guardar archivo	Button
Generar PDF	Button
Para cada responsabilidad:	
Nombre:	
Descripción:	
Nombre: actions.class.php	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	ExecuteBuscarColaborador (CI, Nombre, 1er Apellido, 2do Apellido, No. Registro, estado, Profesional, Especialidad)
Descripción:	Permite buscar un colaborador dado parámetros de búsqueda.
Nombre:	executeLiberarColaboradorFinMision()
Descripción:	Permite liberar colaboradores cuando estos terminan la misión.
Nombre:	executeLiberarColaboradorVacaciones()
Descripción:	Permite liberar colaboradores cuando estos se encuentran de vacaciones.

Tabla #24. Descripción textual de la clase del diseño del CUS “Liberar colaborador”.

En el presente capítulo, después de realizar los diagramas de análisis, diseño y secuencia, se mostró la estructura del funcionamiento del sistema internamente. El análisis y diseño de un software brinda la visión de lo que puede ser la solución en el desarrollo del mismo. Este flujo de trabajo constituyó una abstracción para la implementación del sistema haciendo el trabajo mucho más fácil.

Capítulo 4. Implementación y Prueba.

En el presente capítulo se representarán gráficamente los diagramas de despliegue y de componentes. Además, se explicarán las estrategias de codificación, tratamiento de errores y seguridad.

4.1 Modelo de datos.

Un modelo de datos es básicamente, una descripción de un contenedor de datos, así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son elementos físicos, son abstracciones que permiten la implementación de un sistema eficiente de base de datos. Por lo general, se refieren a algoritmos y conceptos matemáticos; y aportan la base conceptual para diseñar aplicaciones que hacen uso intensivo de datos.

Para el funcionamiento del sistema se diseñó un modelo de datos de clases persistentes que describe los distintos datos, de la información que se manejará en el software. Este modelo contiene tablas con la información referente a los colaboradores y sus distintas tipologías, nómina de ayuda familiar, nómina congelada, vitalicio, personalizaciones y sucursales. [66]

4.2 Implementación.

Diagrama de despliegue.

El siguiente diagrama muestra la configuración hardware del sistema y los nodos físicos que lo componen.

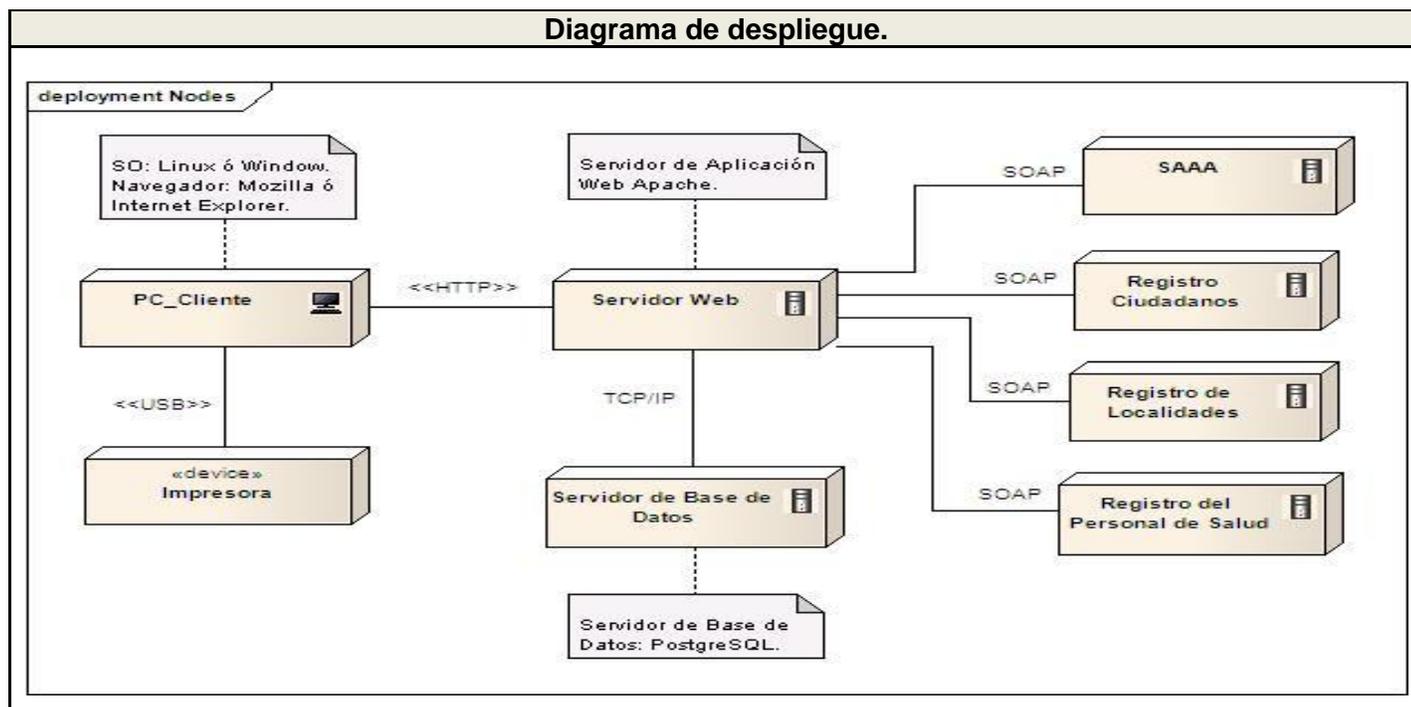


Figura #28: Diagrama de despliegue.

Diagrama de componentes.

El diagrama de componentes describe los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable.

A continuación se presenta un diagrama de componentes general.

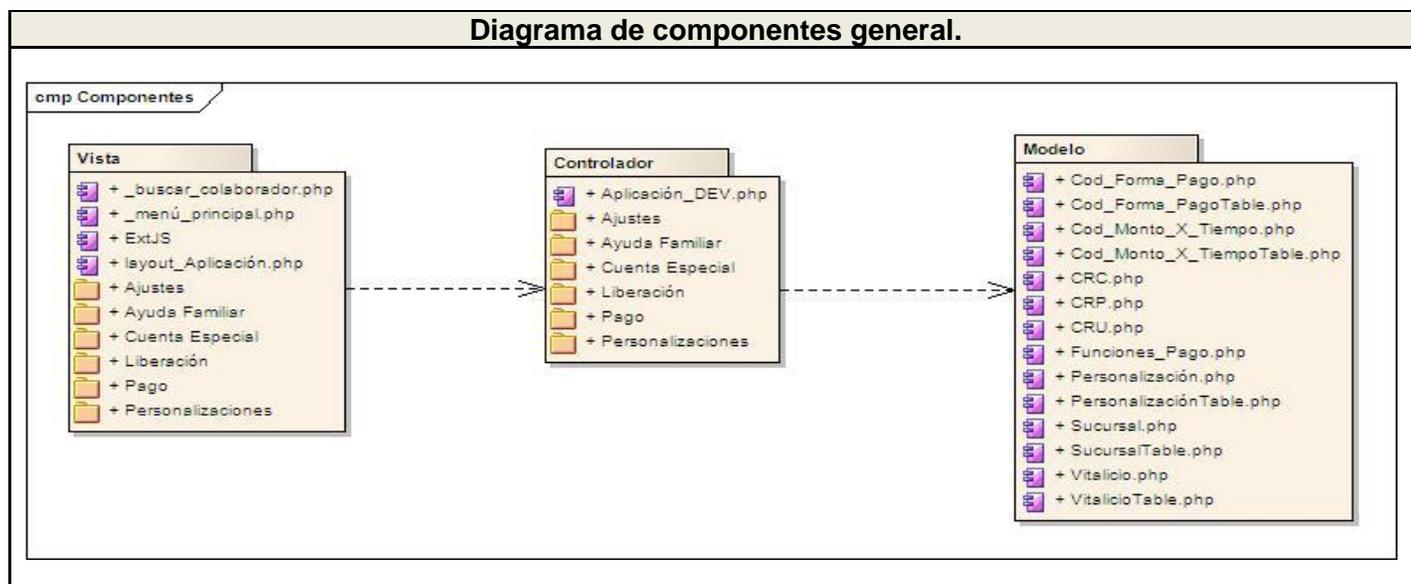


Figura #29: Diagrama de componentes.

4.3 Tratamiento de errores.

El tratamiento de errores es un paso fundamental para una buena elaboración del sistema, debido a que garantiza la integridad y confidencialidad de la información que se maneja en él. Para vencer estos errores uno de los métodos utilizados fue el trabajo con el lenguaje de programación JavaScripts, ya que mediante el mismo quedan informados mucho de estos errores.

En este sistema propuesto serán emitidos los mensajes de errores en el momento en que el usuario cometa alguna equivocación.

4.4 Seguridad.

La seguridad e integridad del sistema se garantizan una vez que se creen los usuarios con los permisos necesarios para la gestión de la información en la base de datos del sistema.

Para lograr la seguridad total del sistema propuesto se realizará la autenticación del mismo mediante el SAAA, de esta manera el usuario obtendrá los permisos necesarios para acceder al sistema. También el sistema de gestor de base datos debe mantenerse en un lugar restringido y asegurado.

4.5 Estrategias de codificación. Estándares y estilos a utilizar.

Para un mejor entendimiento del código en la implementación del sistema es necesario establecer un estándar de codificación a usar, en este caso se ha tenido en cuenta el estilo de código propuesto para la implementación en lenguaje PHP. En la política seguida al respecto, todas las variables y nombres de funciones a utilizar se definieron en idioma español. En el caso de los objetos que se utilizan incluyen el nombre asociado al valor que va en el contenido. Los nombres de las variables utilizadas comienzan en minúscula y son cortos, claros y describen su propósito.

4.6 Tiempo de respuesta de la solución implementada.

El tiempo de respuesta del sistema implementado fue calculado mediante la herramienta Jmeter, la cual permite realizar pruebas de rendimiento a aplicaciones Web. Para la utilización de dicha herramienta se definió un test de 10 hilos de ejecución por cada segundo, el cual simula 10 accesos de usuarios en las versiones existentes de la aplicación. Además, se definió una lista de enlaces a la que se le simuló el acceso aleatorio, seguidamente se recolectaron los datos necesarios para su interpretación.

Los resultados finales se muestran a través del reporte Informe Agregado, utilizado para las funcionalidades Buscar colaborador, Liberación por vacaciones y Generar cuenta especial:

Prueba de tiempo de respuesta en el Sistema de Gestión para la Colaboración Médica v1.0.

Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
TOTAL	2000	109	9	29	1	18401	0,00%	64,4/sec	401,87

Figura #30: Tiempo de respuesta de la versión 1.0.

Prueba de tiempo de respuesta en el Sistema para la Colaboración Médica v2.0.

Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
TOTAL	1800	547	541	630	94	1758	0,00%	17,8/sec	327,64

Figura #31: Tiempo de respuesta de la versión 2.0.

Como se observa en los reportes, el tiempo de respuesta de la versión actual es menor que el de la versión anterior, contribuyendo a la rapidez en el manejo de la información con el desarrollo del Módulo de Pago del Sistema para la Colaboración Médica en la versión 2.0.

En el capítulo se analizó la estructura de la implementación del sistema, permitiendo dar a conocer la importancia que presenta el tratamiento de errores, los estándares de codificación y la seguridad del sistema. Además, se comprobó que el tiempo de respuesta al usuario disminuyó considerablemente en comparación con la versión anterior.

Conclusiones.

La realización del presente trabajo ha cumplido con los objetivos propuestos y contribuye a mejorar la problemática planteada en el proceso de pago a los colaboradores de la salud, por lo que se puede concluir que:

- Con la implementación del Módulo de Pago del Sistema para la Colaboración Médica v2.0, se obtuvo una arquitectura más robusta, mejorando la trazabilidad y auditoría en la gestión del pago a los colaboradores de la salud.
- El módulo desarrollado proporciona un menor tiempo de respuesta al usuario, aumentando su rendimiento.

Recomendaciones.

Para un mejor desempeño de la aplicación se recomienda:

- Definir nuevos requerimientos que permitan adicionar funcionalidades al Módulo de Pago perteneciente a la versión 2.0 del Sistema para la Colaboración Médica, para lograr un mejor control del pago de los colaboradores de la salud en nuevas versiones; como por ejemplo:
 - ✓ Configurar los archivos .dbf, con el objetivo de poder realizar los cambios de la estructura interna en los mismos.
 - ✓ Generar los reportes en formato .pdf con el objetivo de facilitar la portabilidad de la información almacenada de los colaboradores.
- Implementar el módulo de forma genérica que se ajuste a cualquier cambio en la resolución de pago a lo largo tiempo.
- Brindar servicios Web al Portal del Colaborador, sobre la información almacenada del pago mensual y el histórico del salario durante las misiones realizadas por el colaborador.

Referencias Bibliográficas.

1. —. Definición. [En línea] 2008. [Citado el: 22 de octubre de 2009.] <http://definicion.de/informatica/>.
2. —. La informatica en la medicina. [En línea] 22 de octubre de 2007. [Citado el: 22 de octubre de 2009.] <http://informaticaenlamedicina.blogspot.com>.
3. —. Informaciones. [En línea] 03 de 10 de 2009. [Citado el: 14 de octubre de 2009.] <http://www.infodev.org/en/Project.38.html>.
4. —. Colaboración Médica Cubana. [En línea] 9 de octubre de 2007. [Citado el: 16 de octubre de 2009.] <http://colaboracion.sld.cu/.5>. Infomed. [En línea] 2010. <http://www.sld.cu/sitios/bmn/temas.php?idv=12389>.
5. Igual a la referencia bibliográfica número 4.
6. —. Infomed. [En línea] 16 de octubre de 2009. [Citado el: 16 de octubre de 2009.] <http://www.sld.cu/sitios/bmn/temas.php?idv=12389>.
7. Igual a la referencia bibliográfica número 6.
8. —. Definición. [En línea] 2009. [Citado el: 16 de octubre de 2009.] <http://www.definiciones.com/Ministerio-de-Salud/1/>.
Ministerio de Relaciones Exteriores de la República de Cuba. Cuba. [En línea] 2005. [Citado el: 16 de octubre de 2009.] <http://embacu.cubaminrex.cu/Default.aspx?tabid=4709>.
9. —. Sistema de salud. [En línea] 1999 - 2000. [Citado el: 16 de octubre de 2009.] http://www.sld.cu/sistema_de_salud/componentes.html.
—. Congreso Internacional de Información. [En línea] 1999. [Citado el: 4 de noviembre de 2009.] <http://www.congreso-info.cu/>.
10. —. La inversión extranjera en Cuba. [En línea] 2007. [Citado el: 16 de octubre de 2009.] <http://embacuba.cubaminrex.cu/Default.aspx?tabid=6077>.
11. —. Cooperación Internacional. [En línea] 2008. [Citado el: 16 de octubre de 2009.] <http://www.cendeisss.sa.cr/cooperacion/coope1a.html>.
12. —. Mozilla Firefox. [En línea] 2005. [Citado el: 17 de octubre de 2009.] <http://firefoxperu.blogspot.com/2005/12/qu-es-mozilla-firefox.html>.
13. —. Mozilla Firefox e Internet Explorer. [En línea] 9 de enero de 2003. [Citado el: 17 de octubre de 2009.] <http://www.naiandei.net/mozilla101/>.
14. —. Sistema de Nóminas y Recursos Humanos. [En línea] [Citado el: 6 de enero de 2010.] <http://www.saisamx.com/nomina.html>.

REFERENCIAS BIBLIOGRÁFICAS.

15. —. Sistemas de Procesamiento Digital. [En línea] [Citado el: 6 de enero de 2010.] http://mexico.acambiode.com/empresa_6952110061655494856526648514549.html.
16. —. Sistema de Nómina. [En línea] 1999 - 2006. [Citado el: 6 de enero de 2010.] <http://www.eniac.com.mx/eniac/productos.shtml?pag=nomina>.
- 17.—. Productos. [En línea] Empresa Servicios Informáticos, Consultorías y Sistemas, 2008. [Citado el: 6 de enero de 2010.] <http://www.sics.cu/productos.aspx>.
18. —. Sistema Integral Económica Administrativo. [En línea] 2002 - 2009. [Citado el: 6 de enero de 2010.] <http://www.rodasxxi.cu/>.
19. —. Gestión Jet. [En línea] 2004. [Citado el: 6 de enero de 2010.] <http://www.google.com/search?q=.+http://www.JetN%C3%B3minas.cu/&hl=es&lr=&filter=0>.
20. —. Sistema de Recursos Humanos. [En línea] 2004. [Citado el: 6 de enero de 2010.] <http://assets.co.cu/humanos.asp>.
21. Directivos de la UCCM. 11 de noviembre de 2009.
22. —. Mejora de procesos. [En línea] 2004. [Citado el: 16 de enero de 2010.] <http://www.sergiovillagra.com>.
- . Entorno Virtual de Aprendizaje. [En línea] 2009. [Citado el: 16 de enero de 2010.] <http://eva.uci.cu/course/view.php?id=102>.
23. García Murugarren, Joaquin. Prácticas y métodos para mejorar el desarrollo de proyectos de software. [En línea] 2003. [Citado el: 16 de enero de 2010.] <http://www.ingenierosoftware.com/>.
- . Monografías. [En línea] [Citado el: 17 de enero de 2010.] <http://www.monografias.com/trabajos43/patron-modelo-vista/patron-modelo-vista.shtml#patron>.
24. —. Entorno Virtual de Aprendizaje. [En línea] 2009. [Citado el: 16 de enero de 2010.] <http://eva.uci.cu/course/view.php?id=102>.
- . Rubaugh, J., Jacobson, I y Booch, G. El proceso unificado de desarrollo de software. Madrid. 2000.
25. Igual a la referencia bibliográfica número 22.
26. Igual a la referencia bibliográfica número 24.
27. —. Entorno Virtual de Aprendizaje. [En línea] 2009. [Citado el: 16 de enero de 2010.] <http://eva.uci.cu/course/view.php?id=102>.
28. Igual a la referencia bibliográfica número 27.
29. —. Definición de Servicios Web. [En línea] 1997 - 2009. [Citado el: 7 de noviembre de 2009.] <http://diccionario.babylon.com/servicio%20web>.

REFERENCIAS BIBLIOGRÁFICAS.

- . Guía Breve de Servicios Web. [En línea] 1994 - 2005. [Citado el: 4 de noviembre de 2009.]
<http://www.w3c.es/divulgacion/guiasbreves/ServiciosWeb>.
30. —. Apache. [En línea] [Citado el: 16 de enero de 2010.]
<http://www.facilnet.net/matriz/web2/apache.html>.
31. —. PostGreSQL vs. MySQL. [En línea] [Citado el: 18 de enero de 2010.]
http://danielpecos.com/docs/mysql_postgres/index.html.
32. —. Ajax. [En línea] 2007. [Citado el: 20 de enero de 2010.]
<http://www.monografias.com/trabajos43/ajax/ajax2.shtml>.
33. Igual a la referencia bibliográfica número 32.
34. Potencier, F. y Zaninotto, F. Symfony la guía definitiva. New York : s.n., 2009.
35. Igual a la referencia bibliográfica número 34.
36. WordPress y Green Park. ExtJS. [En línea] 2010. [Citado el: 18 de enero de 2010.]
<http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>.
37. —. PHP. [En línea] 2009. [Citado el: 20 de enero de 2010.]
<http://www.desarrolloweb.com/articulos/392.php>.
- . PHP. [En línea] 2010. [Citado el: 20 de enero de 2010.]
http://www.ideocentro.com/recursos.php?id=14&estudio_diseno_web_ideocentro=PHP--PHP-Hypertext-Pre-processor-.
38. —. JavaScript. [En línea] 2010. [Citado el: 18 de enero de 2010.]
<http://www.ulpgc.es/otros/tutoriales/JavaScript/cap1.htm>.
39. Igual a la referencia bibliográfica número 38.
40. —. JMeter. [En línea] 2003 - 2010. [Citado el: 3 de junio de 2010.]
<http://adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=jmeter>.
41. Igual a la referencia bibliográfica número 23.
42. —. Arquitectura Cliente/Servidor. [En línea] [Citado el: 22 de enero de 2010.]
<http://www.csae.map.es/csi/silice/Global71.html>.
43. —. Ventajas de la Arquitectura Cliente/Servidor. [En línea] [Citado el: 22 de enero de 2010.]
<http://www.csae.map.es/csi/silice/Global75.html>.
44. —. Enterprise Architect. [En línea] 2000 - 2008. [Citado el: 23 de enero de 2010.]
<http://www.sparxsystems.es/New/products/index.html#EA>.
45. Igual a la referencia bibliográfica número 44.

REFERENCIAS BIBLIOGRÁFICAS.

46. —. NetBeans. [En línea] 2009. [Citado el: 16 de enero de 2010.] <http://blogcultura.com/java/netbeans-6-8-liberado/>.
47. —. NetBeans. [En línea] 1994 - 2010. [Citado el: 20 de enero de 2010.] <http://es.sun.com/sunnews/press/2009/20091214.jsp>.
48. —. Entorno Virtual de Aprendizaje. [En línea] 2009. [Citado el: 16 de enero de 2010.] <http://eva.uci.cu/course/view.php?id=102>.
49. Igual a la referencia bibliográfica número 48.
50. Igual a la referencia bibliográfica número 48.
51. Igual a la referencia bibliográfica número 48.
52. Igual a la referencia bibliográfica número 48.
53. Igual a la referencia bibliográfica número 48.
54. —. El proceso unificado del desarrollo del software. [En línea] 2007. [Citado el: 22 de enero de 2010.] <http://www.monografias.com/trabajos22/desarrollo-software/desarrollo-software.shtml>.
55. —. Entorno Virtual de Aprendizaje. [En línea] 2009. [Citado el: 16 de enero de 2010.] <http://eva.uci.cu/course/view.php?id=102>.
56. Igual a la referencia bibliográfica número 50.
57. Igual a la referencia bibliográfica número 50.
58. —. Patrones Grasp. [En línea] 2007. [Citado el: 22 de enero de 2010.] <http://jorgesaavedra.wordpress.com/category/patrones-grasp/>.
59. —. Entorno Virtual de Aprendizaje. [En línea] 2009. [Citado el: 16 de enero de 2010.] <http://eva.uci.cu/mod/resource/view.php?id=22668>.
60. Igual a la referencia bibliográfica número 59.
61. Igual a la referencia bibliográfica número 59.
62. —. Entorno Virtual de Aprendizaje. [En línea] 2009. [Citado el: 16 de enero de 2010.] <http://eva.uci.cu/mod/resource/view.php?id=14069>.
63. Igual a la referencia bibliográfica número 62.
64. Igual a la referencia bibliográfica número 59.
65. Igual a la referencia bibliográfica número 59.
66. —. Repositorio. [En línea] 2010 [Citado el 2 de junio de 2010.] http://10.36.7.200/sas_pm/Colaboracionv2.0.

Bibliografía.

1. —. Ajax. [En línea] 2007. [Citado el: 20 de enero de 2010.] <http://www.monografias.com/trabajos43/ajax/ajax2.shtml>.
2. —. Apache. [En línea] [Citado el: 16 de enero de 2010.] <http://www.facilnet.net/matriz/web2/apache.html>.
3. —. Arquitectura Cliente/Servidor. [En línea] [Citado el: 22 de enero de 2010.] <http://www.csae.map.es/csi/silice/Global71.html>.
4. —. Colaboración Médica Cubana. [En línea] 9 de octubre de 2007. [Citado el: 16 de octubre de 2009.] <http://colaboracion.sld.cu/>.
5. —. Conceptos básicos del servidor web. [En línea] [Citado el: 4 de noviembre de 2009.] http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php.
6. —. Congreso Internacional de Información. [En línea] 1999. [Citado el: 4 de noviembre de 2009.] <http://www.congreso-info.cu/>.
7. —. Cooperación Internacional. [En línea] 2008. [Citado el: 16 de octubre de 2009.] <http://www.cendeiss.sa.cr/cooperacion/coope1a.html>.
8. —. Definición. [En línea] 2008. [Citado el: 22 de octubre de 2009.] <http://definicion.de/informatica/>.
9. —. Definición. [En línea] 2009. [Citado el: 16 de octubre de 2009.] <http://www.definiciones.com/Ministerio-de-Salud/1/>.
10. —. Definición de Servicios Web. [En línea] 1997 - 2009. [Citado el: 7 de noviembre de 2009.] <http://diccionario.babylon.com/servicio%20web>.
11. —. Enterprise Architect. [En línea] 2000 - 2008. [Citado el: 23 de enero de 2010.] <http://www.sparxsystems.es/New/products/index.html#EA>.
12. —. Entorno Virtual de Aprendizaje. [En línea] 2009. [Citado el: 16 de enero de 2010.] <http://eva.uci.cu/course/view.php?id=102>.
13. —. Entorno Virtual de Aprendizaje. [En línea] 2009. [Citado el: 16 de enero de 2010.] <http://eva.uci.cu/mod/resource/view.php?id=14069>.
14. —. Gestión de Proyectos. [En línea] 2004. [Citado el: 10 de enero de 2010.] <http://www.sergiovillagra.com>.
15. —. Gestión Jet. [En línea] 2004. [Citado el: 6 de enero de 2010.] <http://www.google.com/search?q=+http://www.JetN%C3%B3minas.cu/&hl=es&lr=&filter=0>.

16. —. Guía Breve de Servicios Web. [En línea] 1994 - 2005. [Citado el: 4 de noviembre de 2009.] <http://www.w3c.es/divulgacion/guiasbreves/ServiciosWeb>.
17. —. Infomed. [En línea] 16 de 10 de 2009. [Citado el: 16 de octubre de 2009.] <http://www.sld.cu/sitios/bmn/temas.php?idv=12389>.
18. —. Informaciones. [En línea] 03 de 10 de 2009. [Citado el: 14 de octubre de 2009.] <http://www.infodev.org/en/Project.38.html>.
19. —. La informática en la medicina. [En línea] 22 de octubre de 2007. [Citado el: 22 de octubre de 2009.] <http://informaticaenlamedicina.blogspot.com>.
20. —. La inversión extranjera en Cuba. [En línea] 2007. [Citado el: 16 de octubre de 2009.] <http://embacuba.cubaminrex.cu/Default.aspx?tabid=6077>.
21. —. JavaScript. [En línea] 2010. [Citado el: 18 de enero de 2010.] <http://www.ulpgc.es/otros/tutoriales/JavaScript/cap1.htm>.
22. —. JMeter. [En línea] 2003 - 2010. [Citado el: 3 de junio de 2010.] <http://adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=jmeter>.
23. —. Ministerio de Educación y Ciencia. 2009.
24. —. Monografias. [En línea] [Citado el: 17 de enero de 2010.] <http://www.monografias.com/trabajos43/patron-modelo-vista/patron-modelo-vista.shtml#patron>.
25. —. Mozilla Firefox. [En línea] 2005. [Citado el: 17 de octubre de 2009.] <http://firefoxperu.blogspot.com/2005/12/qu-es-mozilla-firefox.html>.
26. —. Mozilla Firefox e Internet Explorer. [En línea] 9 de enero de 2003. [Citado el: 17 de octubre de 2009.] <http://www.naiandei.net/mozilla101/>.
27. —. NetBeans. [En línea] 1994 - 2010. [Citado el: 20 de enero de 2010.] <http://es.sun.com/sunnews/press/2009/20091214.jsp>.
28. —. NetBeans. [En línea] 2009. [Citado el: 16 de enero de 2010.] <http://blogultura.com/java/netbeans-6-8-liberado/>.
29. —. Patrones Grasp. [En línea] 2007. [Citado el: 22 de enero de 2010.] <http://jorgesaaavedra.wordpress.com/category/patrones-grasp/>.
30. —. PHP. [En línea] 2009. [Citado el: 20 de enero de 2010.] <http://www.desarrolloweb.com/articulos/392.php>.

31. —. PHP. [En línea] 2010. [Citado el: 20 de enero de 2010.] http://www.ideocentro.com/recursos.php?id=14&estudio_diseno_web_ideocentro=PHP--PHP-Hypertext-Pre-processor-.
32. —. PostGreSQL vs. MySQL. [En línea] [Citado el: 18 de enero de 2010.] http://danielpecos.com/docs/mysql_postgres/index.html.
33. —. El proceso unificado del desarrollo del software. [En línea] 2007. [Citado el: 22 de enero de 2010.] <http://www.monografias.com/trabajos22/desarrollo-software/desarrollo-software.shtml>.
34. —. Productos. [En línea] Empresa Servicios Informáticos, Consultorías y Sistemas, 2008. [Citado el: 6 de enero de 2010.] <http://www.sics.cu/productos.aspx>.
35. —. Repositorio. [En línea] 2010 [Citado el 2 de junio de 2010.] http://10.36.7.200/sas_pm/Colaboracion v2.0.
36. —. Sistema de Nómina. [En línea] 1999 - 2006. [Citado el: 6 de enero de 2010.] <http://www.eniac.com.mx/eniac/productos.shtml?pag=nomina>.
37. —. Sistema de Nóminas y Recursos Humanos. [En línea] [Citado el: 6 de enero de 2010.] <http://www.saisamx.com/nomina.html>.
38. —. Sistemas de Procesamiento Digital. [En línea] [Citado el: 6 de enero de 2010.] http://mexico.acambiode.com/empresa_6952110061655494856526648514549.html.
39. —. Sistema de Recursos Humanos. [En línea] 2004. [Citado el: 6 de enero de 2010.] <http://assets.co.cu/humanos.asp>.
40. —. Sistema de salud. [En línea] 1999 - 2000. [Citado el: 16 de octubre de 2009.] http://www.sld.cu/sistema_de_salud/componentes.html..
41. —. Sistema Integral Económica Administrativo. [En línea] 2002 - 2009. [Citado el: 6 de enero de 2010.] <http://www.rodasxxi.cu/>.
42. —. Ventajas de la Arquitectura Cliente/Servidor. [En línea] [Citado el: 22 de enero de 2010.] <http://www.csae.map.es/csi/silice/Global75.html>.
43. Directivos de la UCCM. 11 de noviembre de 2009.
44. García Murugarren, Joaquín. Prácticas y métodos para mejorar el desarrollo de proyectos de software. [En línea] 2003. [Citado el: 16 de enero de 2010.] <http://www.ingenierosoftware.com/>.
45. H. Cámara de Diputados. Ley 1.032/96. [En línea] 22 de agosto de 1996. [Citado el: 16 de octubre de 2009.] http://www.cej.org.py/games/Leyes_por_Materia_juridica/SALUD/LEY%20N%C2%BA%201032.pdf.

46. Ministerio de Relaciones Exteriores de la República de Cuba. Cuba. [En línea] 2005. [Citado el: 16 de octubre de 2009.] <http://embacu.cubaminrex.cu/Default.aspx?tabid=4709>.
47. Potencier, F. y Zaninotto, F. Symfony la guía definitiva. New York: s.n., 2009.
48. Rubaugh, J., Jacobson, I y Booch, G. El proceso unificado de desarrollo de software. Madrid. 2000.
49. WordPress y Green Park. ExtJS. [En línea] 2010. [Citado el: 18 de enero de 2010.] <http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>.

Glosario de Términos.

Framework: Es una estructura de soporte definida en la cual un proyecto de software puede ser organizado y desarrollado.

GNU/GPL: La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU (General Public License), es una licencia creada por la fundación de software libre a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

Informatizar: Proceso de aplicar sistemas o equipos informáticos al tratamiento de la información.

Linux: Es un sistema operativo tipo Unix (también conocido como GNU/Linux) que se distribuye bajo la GNU y el GPL, es decir que es software libre. Su nombre proviene del núcleo de Linux, desarrollado en 1991 por Linus Torvalds.

SOAP: (Siglas de Simple Object Access Protocol) es un protocolo estándar creado por Microsoft, IBM y otros. Es uno de los protocolos utilizados en los servicios Web.

Software: Conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema.

Software Libre: Es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

Subsistema: Agrupación de elementos, de los que algunos constituyen una especificación del comportamiento ofrecido por los elementos contenidos.

TCP/IP: Conjunto de protocolos de red en la que se basa Internet y que permiten la transmisión de datos entre redes de computadoras.

Anexos.

Anexo 1. Diagrama de actividades.

Diagrama de actividades del caso de uso del negocio "Realizar misión".

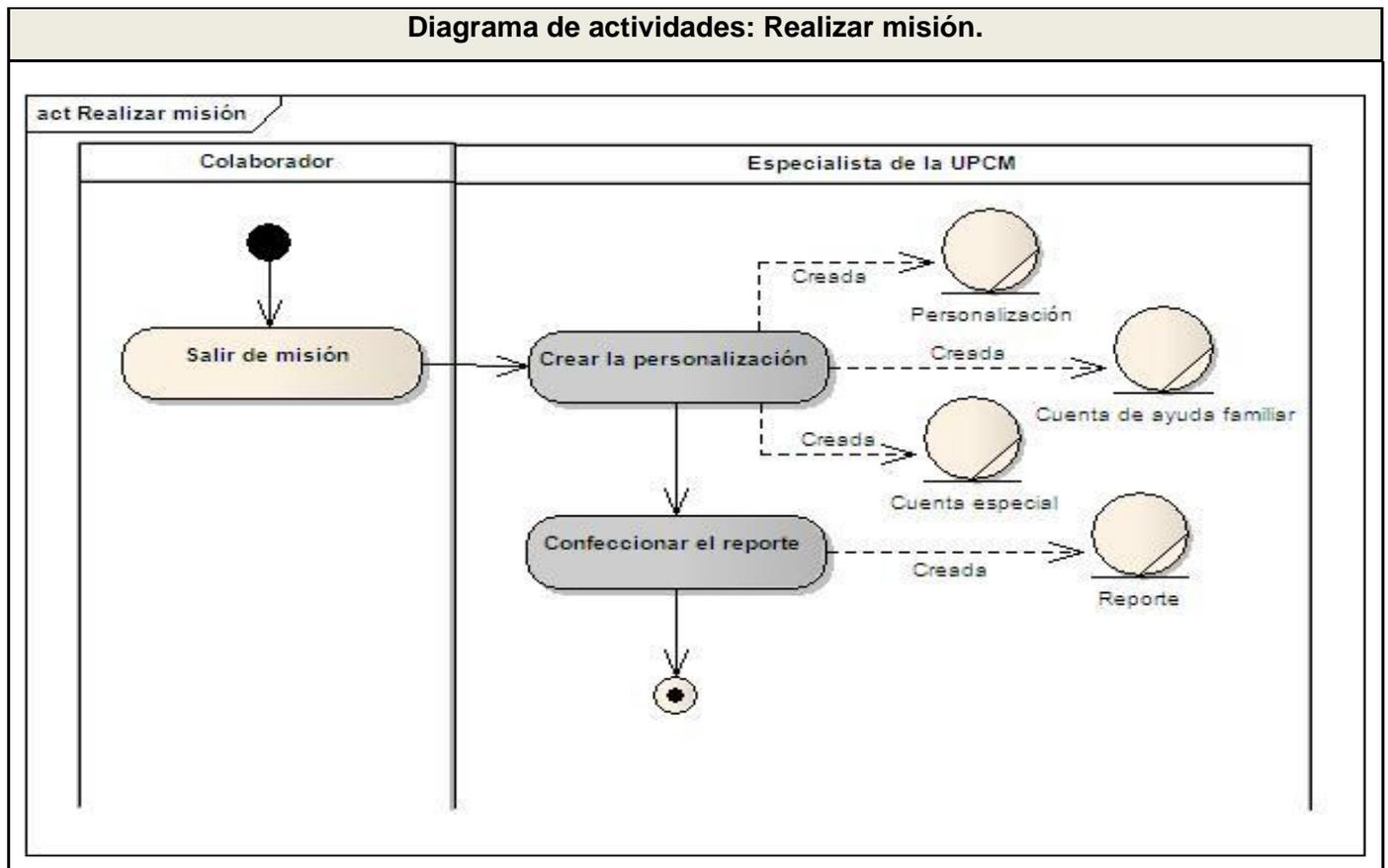


Figura #34: Diagrama de actividad del CUN "Realizar misión".

Diagrama de actividades del caso de uso del negocio “Cobrar salario”.

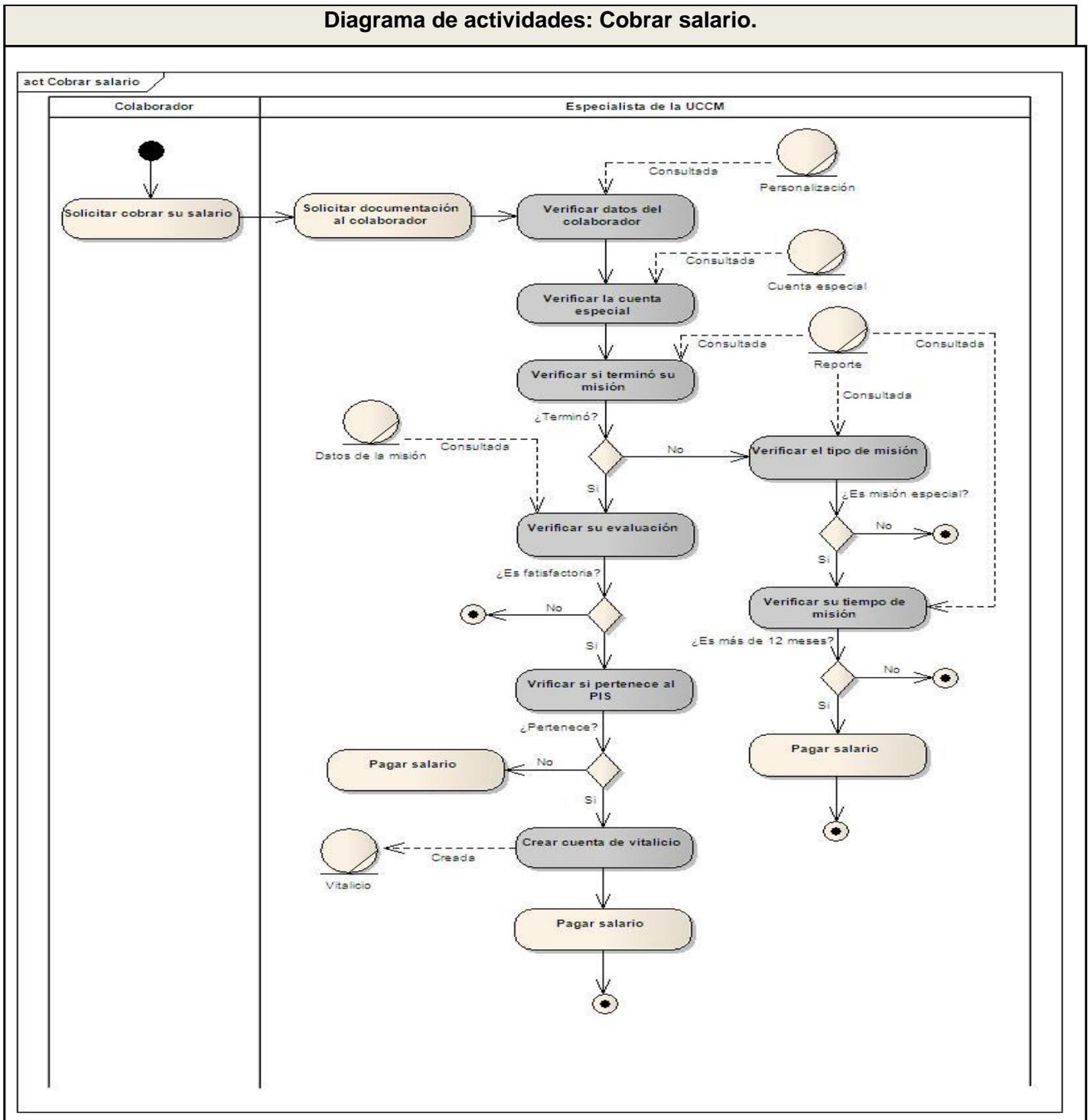


Figura #32: Diagrama de actividad del CUN “Cobrar salario”.

Diagrama de actividades del caso de uso del negocio “Cobrar salario de ayuda familiar”.

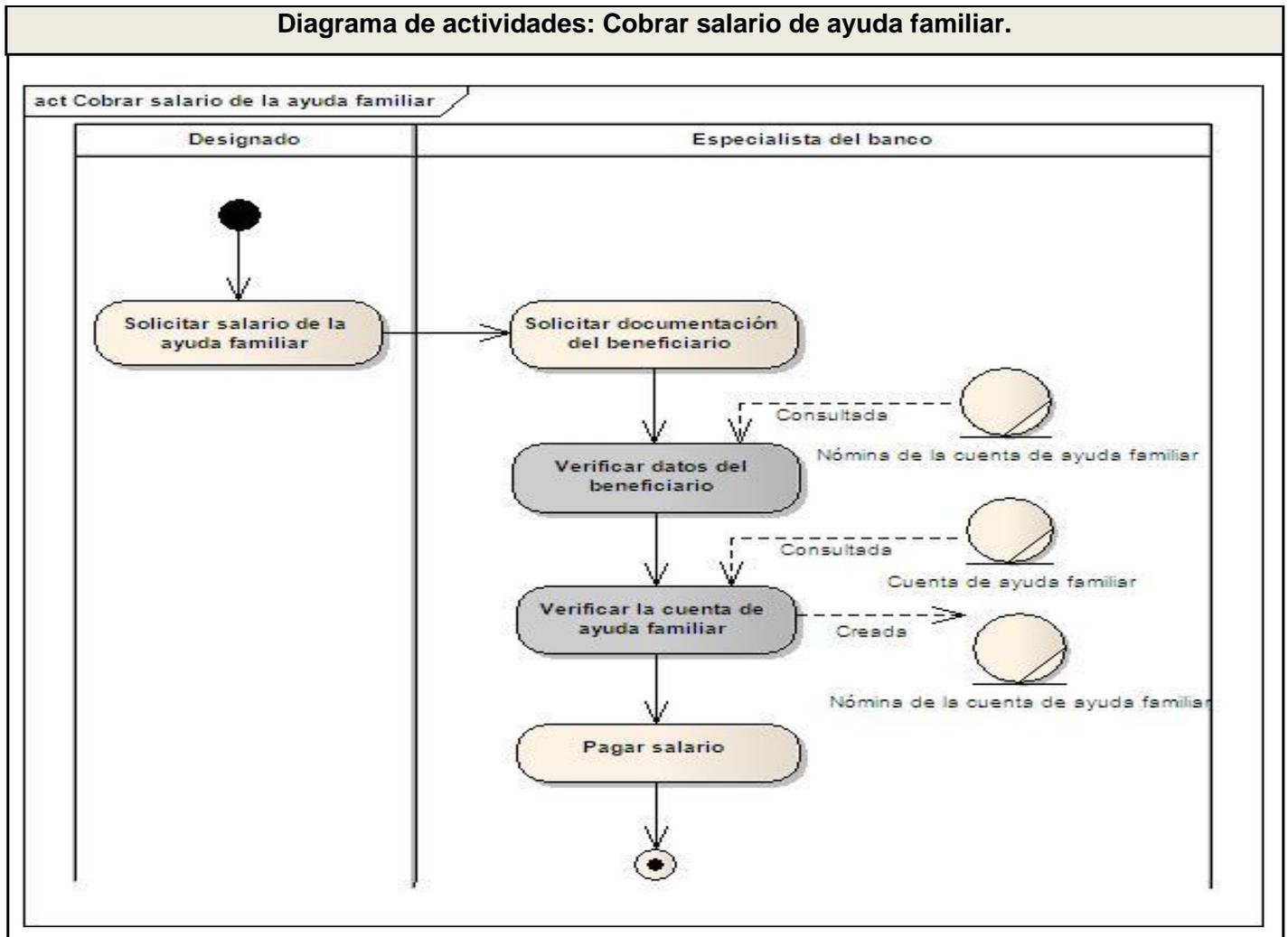


Figura #33: Diagrama de actividad del CUN “Cobrar salario de ayuda familiar”.