



**Universidad de las Ciencias Informáticas**

**FACULTAD 7**

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

**Título: Módulo para la gestión de la información de los  
colaboradores de la salud v2.0**

**Autores:**

Diainys González Benítez

Yuandy Riverón González

**Tutora:**

Ing. Yisel Reyes Cardoso

**Ciudad de La Habana, junio del 2010**

**“Año 52 de la Revolución”**

## **Datos del contacto**

**Nombre: 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 Álgebra Lineal. Actualmente pertenece al Departamento de Sistemas de Apoyo a la Salud del Centro de Informática Médica (CESIM).

Correo: [ycardoso@uci.cu](mailto:ycardoso@uci.cu)

## Resumen

Con el objetivo de mantener el control de los datos de los colaboradores en la Unidad Central de Cooperación Médica (UCCM), se realiza el Módulo para la Gestión de la Información de los Colaboradores, el cual debe satisfacer las necesidades del Sistema Nacional de Salud y de la UCCM soportando y controlando el flujo de información que en el centro se maneja.

El sistema se desarrolló utilizando el lenguaje de programación PHP v5.2.6. Para una mejor estructuración del código fuente se utilizó el framework Symfony v1.4.3. Como Sistema Gestor de Base de Datos se utilizó PostgreSQL v8.3, se eligió Apache como el servidor Web sobre el que correrá la aplicación. Se trabajó con la herramienta Enterprise Architect v7.1 para realizar la modelación haciendo uso de la notación UML v2.0 (Lenguaje Unificado de Modelado) y el Sistema Operativo en el Servidor es Linux Distribución Ubuntu v9.4.

El sistema propuesto es de gran ayuda para el control de las colaboraciones médicas cubanas, ya que permite la gestión de la información de todo el personal en misiones de la salud, facilitando un mejor control de los recursos humanos. Tiene un alcance a los diferentes niveles del sistema nacional de salud y podrán acceder a la información, garantizando la integridad, manejabilidad y confiabilidad de los datos mediante la integración a los registros disponibles en el Registro Informatizado de la Salud (RIS).

**Palabras claves:** Colaboradores, Colaboración médica, Recursos humanos, UCCM.

## Índice

|   |           |
|---|-----------|
| <b>Introducción</b> .....   | <b>1</b>  |
| <b>Capítulo 1: Fundamentación teórica</b> .....                                   | <b>6</b>  |
| 1.1 Conceptos fundamentales .....   | 6         |
| 1.2 Estudio del Arte .....  | 7         |
| 1.2.1 Nivel internacional.....  | 7         |
| 1.2.2 Nivel nacional.....   | 8         |
| 1.2.3 Universidad de las Ciencias Informáticas .....                              | 10        |
| 1.3 Tecnologías, Metodologías, Herramientas y Lenguajes utilizados.....           | 11        |
| 1.3.1 Integración de Modelos de Madurez de Capacidades (CMMI) .....               | 11        |
| 1.3.2 Proceso Unificado de Desarrollo (RUP) .....                                 | 12        |
| 1.3.4 Leguaje Unificado de Modelado (UML) .....                                   | 13        |
| 1.3.5 Enterprise Architect v7.1 .....   | 13        |
| 1.3.6 NetBeans .....  | 14        |
| 1.3.7 Hypertext Pre-processor (PHP).....  | 14        |
| 1.3.8 JavaScript .....  | 14        |
| 1.3.9 Ajax.....   | 15        |
| 1.3.10 Arquitecturas del Sistema .....  | 15        |
| 1.3.11 Sistema Operativo.....   | 16        |
| 1.3.12 Sistema Gestor de Base de Datos (SGBD) .....                               | 17        |
| 1.3.13 Servidor Web .....   | 17        |
| 1.3.14 Navegadores.....   | 18        |
| 1.3.15 Servicios Web .....  | 18        |
| 1.3.16 Framework de desarrollo.....   | 19        |
| <b>Capítulo 2: Características del sistema</b> .....                              | <b>23</b> |
| 2.2 Propuesta del Sistema.....  | 23        |
| 2.3 Modelo de negocio .....   | 24        |
| 2.4 Especificación de los requisitos .....  | 27        |
| 2.5 Descripción del Sistema .....   | 32        |
| 2.6 Diagrama de Casos de Uso del Sistema .....                                    | 33        |
| 2.7 Especificación de los Casos de Uso .....                                      | 33        |
| <b>Capítulo 3: Diseño del sistema</b> .....                                       | <b>37</b> |
| 3.2 Descripción de la arquitectura .....  | 37        |
| 3.3 Análisis de componentes o módulos rehusados. Estrategias de integración ..... | 38        |
| 3.3.1 Componentes reutilizados.....   | 38        |
| 3.3.2 Estrategia de integración .....   | 39        |
| 3.4 Análisis.....   | 39        |
| 3.4.1 Modelo de Análisis .....  | 40        |
| 3.4.2 Clases de Análisis.....   | 40        |
| 3.4.3 Diagrama de clase del análisis.....   | 40        |
| 3.5 Diseño .....  | 43        |
| 3.5.1 Modelo de Diseño .....  | 43        |
| 3.5.2 Diagrama de Clases de Diseño del Sistema .....                              | 43        |
| 3.5.3 Diagrama de Secuencia.....  | 46        |

---

|   |           |
|---|-----------|
| 3.5.4 Descripción de las clases del diseño.....                       | 52        |
| <b>Capítulo 4: Implementación .....</b>                               | <b>57</b> |
| 4.2 Modelo de datos.....  | 57        |
| 4.3 Diagrama de despliegue.....                                       | 57        |
| 4.4 Diagrama de componentes.....                                      | 58        |
| 4.5 Tratamiento de errores .....                                      | 59        |
| 4.6 Seguridad.....  | 59        |
| 4.7 Estrategias de codificación. Estándares y estilos utilizados..... | 59        |
| 4.8 Tiempo de respuesta de la solución implementada .....             | 60        |
| <b>Beneficios.....</b>  | <b>61</b> |
| <b>Conclusiones Generales .....</b>                                   | <b>61</b> |
| <b>Recomendaciones .....</b>  | <b>63</b> |
| <b>Referencias Bibliográficas .....</b>                               | <b>64</b> |
| <b>Bibliografía .....</b>   | <b>67</b> |
| <b>Glosario de Términos .....</b>                                     | <b>69</b> |
| <b>Anexos.....</b>  | <b>70</b> |
| Anexo 1. Imágenes de propuesta del sistema.....                       | 70        |
| Anexo 2. Diagramas de actividades.....                                | 71        |

## Introducción

En Cuba, antes del triunfo de la revolución no se brindaba ningún tipo de colaboración médica, pero ya en el año 1960 ofrece su primera Ayuda Médica Internacional enviando una Brigada de colaboradores a Chile, después del terremoto que azotó a ese país. A fines de 1962, en el acto inaugural de la Facultad de Ciencias Básicas y Preclínicas “Victoria de Playa 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. Dicha 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. Está confirmado que durante estas cuatro décadas un total de 79 487 colaboradores han brindado su ayuda, distribuidos aproximadamente en 97 países por todo el mundo, es decir, 32 074 en 34 países de África, 39 334 en 37 países de América y 277 en 9 países de Europa. (1)

Durante ese período, el personal de salud cubano realizó 66 millones de consultas: de estas, más de 13 millones fueron realizadas en viviendas ubicadas mayoritariamente en comunidades pobres y de difícil acceso, así como también superaron los 607 mil partos asistidos. Otro logro de esa colaboración internacionalista cubana es la realización de casi dos millones de intervenciones quirúrgicas y el desarrollo de programas de lucha contra el Virus de la Inmunodeficiencia Humana (VIH) mediante actividades de capacitación.

Cuba es hoy el país que más ha hecho por compartir sus conocimientos, experiencias médicas y científicas, sin importar diferencias ideológicas ni políticas, barreras idiomáticas, sacrificios u obstáculos, con más de cinco décadas de internacionalismo en la salud, salvando más de un millón de vidas, abre nuevos horizontes en el camino de la solidaridad. Debido a la importancia que tiene para el país prestar su ayuda solidaria, se crea la Unidad Central de Cooperación Médica (UCCM), con el objetivo de almacenar toda la información posible de la persona que se encuentra cumpliendo misión, es la encargada de gestionar todo lo relacionado con el colaborador cubano dentro y fuera del territorio nacional. (2)

En sus inicios todo el proceso se realizaba de forma manual, provocando la pérdida de valiosa información, además de horas de engorroso trabajo y papeleo, de aquí la importancia que tiene para el país la informatización de la sociedad, que se define en Cuba como el proceso de utilización ordenada y masiva de las Tecnologías de la Informática y las Comunicaciones (TIC) para satisfacer las necesidades

de información y conocimiento de todas las personas y esferas de la sociedad. Este proceso busca lograr más eficiencia, que permita una mayor generación de riquezas y haga sustentable el aumento sistemático de la calidad de vida de los cubanos. La dirección del país ha priorizado en esta tarea el sector de la salud, que tiene como objetivo principal abarcar todas las ramas de los diferentes niveles de atención médica del Sistema Nacional de Salud.

Con la aplicación de las TIC, en la UCCM se decide desarrollar un pequeño sistema para agilizar el control de la colaboración médica cubana, tratando de evitar que la información se manejara manualmente para mayor seguridad y eficiencia. Inicialmente se implementó una aplicación en Santiago de Cuba, en el año 2004, utilizando Delphi como lenguaje de programación, con una base de datos en Microsoft Access 97, resolviendo el problema en ese momento.

Posteriormente en la UCCM surgieron cambios en cuanto a las políticas del negocio, incrementando sus necesidades, lo que trae consigo que el sistema que existía ya no era óptimo para cumplir con las expectativas requeridas además de que resultaba obsoleto por las tecnologías que se utilizaban, dándole la tarea de desarrollar un nuevo sistema a la Universidad de las Ciencias Informáticas (UCI) en colaboración con la empresa especializada en Soluciones Informáticas (SOFTTEL). Se implementó un nuevo sistema utilizando tecnología Web. Pero este no se adapta a los lineamientos de arquitectura definidos por el Centro de Informática Médica, además de presentar las siguientes deficiencias:

- Existe una gran dependencia del sistema gestor de base de datos MySQL porque posee mucha lógica del negocio dentro del mismo, además se pueden encontrar clases de acceso a la información en cualquier capa del sistema. Sin dejar de mencionar que el uso de este gestor se encuentra limitado para el país.
- La aplicación no cuenta con las mejores prácticas y patrones de diseño propuestos para la Web.
- 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 Módulo de Gestión 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.

- La trazabilidad y la auditoría son elementos que el sistema no proporciona, los cuales son de gran importancia por la integridad de la información.
- El tiempo de respuesta del sistema se torna demasiado lento por la sobrecarga del almacenamiento temporal de datos (caché) con el uso excesivo de la variable Sesión.

Luego de analizar la problemática planteada y las dificultades existentes en la forma en que se gestiona la información en la UCCM se identifica como **problema científico**: ¿Cómo mejorar la trazabilidad y la auditoría, siguiendo los lineamientos de la arquitectura definidos por el Centro de Informática Médica en el Módulo para la Gestión de la información los Colaboradores, en la versión 1.0 del Sistema de Gestión para la Colaboración Médica en la UCCM?

El problema planteado delimita el **objeto de estudio**: Gestión de la información en el Departamento de Recursos Humanos de la UCCM. El objeto de estudio se enmarca en el **campo de acción**: La coordinación de la información de los colaboradores de la salud en el Departamento de Recursos Humanos de la UCCM.

Por lo antes planteado se definió como **objetivo de la investigación**: Desarrollar el Módulo de Gestión de la Información de los Colaboradores 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.

Para lograr el objetivo de la investigación planteado se trazaron las siguientes **tareas de la investigación**:

- Investigar sobre los sistemas existentes relacionados con la gestión de información de recursos humanos, para obtener un mejor punto de partida en la realización del nuevo sistema.
- Analizar la coordinación 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 arquitecturas definidas por el Departamento Sistemas de Apoyo a la Salud (SAS) para la implementación de la nueva solución para mantener una homogeneidad en el uso de las herramientas utilizadas en el mismo.

- Examinar los servicios públicos Registro del Ciudadano (RC), Registro de Personal de la Salud (RPS), Sistema de Administración (SAAA), Registro de Localidades (RL) del Registro Informatizado de Salud (RIS), para establecer la integración requerida con los mismos.
- Desarrollar los artefactos correspondientes a los flujos de trabajo del Proceso Unificado de Desarrollo (RUP).
- Implementar la versión 2.0 del Módulo para la Gestión de la Información de los Colaboradores de la Salud.

Dándole cumplimiento a estas tareas, se resolverá la problemática existente en el Dpto. de Recursos Humanos de la UCCM, donde el usuario va a controlar la información necesaria de todos los colaboradores del país. Se mejorará en rapidez en la gestión de la información relacionada con los colaboradores, permitiendo un mejor control de los Recursos Humanos para la toma de decisiones.

El documento está estructurado de la siguiente forma:

**Capítulo I: Fundamentación Teórica.** Se realiza un análisis sobre los diferentes aspectos teóricos relacionados con el desarrollo del tema tratado. Teniendo en cuenta las necesidades y las características del entorno donde se aplicará la solución propuesta. Se hará una descripción de todas las herramientas, tecnologías y metodologías a utilizar para dar solución al problema planteado.

**Capítulo 2: Características del Sistema.** Este capítulo se enfoca en todo lo relacionado con las características del sistema, plantea una propuesta para el desarrollo del mismo, se identifican los actores, trabajadores y casos de uso del negocio. Se definen los requisitos funcionales, no funcionales y se modela el sistema. Se expone además, el estudio realizado acerca del Módulo para la gestión de la información de los colaboradores del Sistema de Gestión para la Colaboración Médica. Se detalla la forma en que se realizan las labores que serán objeto de automatización para determinar con mayor claridad las responsabilidades llevadas a cabo.

**Capítulo 3: 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 representan a la aplicación, además de una

explicación sobre los principios de diseño que se tuvieron en cuenta para la realización del sistema, así como la descripción de la interfaz, la seguridad y la concepción de la ayuda para un mejor manejo de la aplicación.

**Capítulo 4: 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. Se indica cómo los elementos del diseño se implementarán y se organizarán de acuerdo con el diagrama de despliegue. Además, se realiza el diagrama de componentes general que organiza las dependencias entre un conjunto de componentes, y el diagrama de despliegue que permite la configuración de nodos de procesamiento y los componentes que residen en ellos; todo dando lugar al buen control de la aplicación.

### **Capítulo 1: Fundamentación teórica**

Este capítulo está dedicado a realizar un análisis sobre los diferentes aspectos teóricos relacionados con el desarrollo del tema tratado a nivel nacional. Teniendo en cuenta las necesidades y las características del entorno donde se aplicará la solución propuesta.

Se describen los sistemas integrados y distribuidos, los antecedentes del sistema, las herramientas que se utilizarán para el desarrollo del mismo, la metodología a utilizar para el análisis y diseño, teniendo en cuenta las facilidades que puede aportar al trabajo. Se hace un estudio de las tecnologías que puedan ser adecuadas para el desarrollo del sistema, profundizando en las propuestas por el Ministerio de Salud Pública (MINSAP) y el Dpto. de Sistemas de Apoyo a la Salud, del Centro de Informática Médica.

#### **1.1 Conceptos fundamentales**

##### **Unidad Central de Cooperación Médica:**

Desde el año 1960 el país brindó su primera ayuda médica cuando envía una brigada emergente y varias toneladas de equipos e insumos a Chile, país afectado por un intenso terremoto. Pero se considera como fecha de inicio de la Colaboración Médica Internacional Cubana con brigadas permanentes, el 23 de Mayo de 1963, con el envío de una brigada a Argelia, compuesta por 55 colaboradores que brindaron sus servicios por un año.

Debido a la importancia que tiene para Cuba prestar ayuda solidaria a países hermanos, se crea la UCCM, con el objetivo de almacenar toda la información posible de la persona que está cumpliendo la misión, es la encargada de gestionar todo lo relacionado con el colaborador cubano dentro y fuera del territorio nacional. (3)

##### **Registro Informatizado de Salud (RIS):**

Es la solución informática integral para la Salud Pública acorde con los objetivos de la informatización de la sociedad cubana. Constituido por un conjunto de aplicaciones independientes que se interconectan según las necesidades del flujo de información. Es además la herramienta que permite a los usuarios autorizados combinar la información de los diferentes módulos que lo componen, para obtener una

información integral en tiempo real para la toma de decisiones en los diferentes niveles de dirección, la docencia, investigación y la gestión en salud.

El RIS, es un sistema disponible en la intranet médica cubana, conocida como INFOMED, la Red Telemática de Salud. Es la solución propuesta por el MINSAP para la informatización de las instituciones de la Salud Pública. Más que un software, es una plataforma de aplicaciones, abierta, con una interfaz de programación que permite incorporar nuevos módulos que sean compatibles con ella. Es portable a diferentes sistemas operativos, tanto en los servidores como en los clientes y replicable en otros entornos. Es una plataforma en constante desarrollo, que crece en la medida en que se implementan nuevos módulos. (4)

## 1.2 Estudio del Arte

### 1.2.1 Nivel internacional

En el mundo existen diversos sistemas que gestionan la información del personal que labora en una empresa determinada, entre estos se pueden citar los siguientes sistemas: Empleados MSD Multiusuario versión 3.2, Wikimetal RRHH versión 3.0, @GestRRHH versión 2.0 y GestPeople versión 2.0, estos sistemas fueron diseñados para mejorar el alcance de los objetivos de los empleados de una empresa. Sus características son:

**Empleados MSD Multiusuario v3.2:** Avanzado gestor de información de los empleados de una empresa que permite manejar datos personales, familiares, historial, documentos y titulaciones entre otros servicios. Este sistema está estructurado por cuatro módulos: Módulo Empleados, Módulo Información de Empleados, Módulo Resumen, Módulo Estadísticas. Es una aplicación fácil de utilizar y permite configurar muchas de sus características para adaptarse a los gustos del usuario, además de permitirle un alto grado de privacidad y de aportarle herramientas de suma utilidad tales como copias de seguridad, procesador de textos, calculadora y otras. (5)

**Wikimetal RRHH v3.0:** Sistema destinado a la gestión del personal de una empresa, es una aplicación simple y de fácil manejo que ha sido creada con el fin de ayudar a las empresas en la difícil tarea del control de sus empleados. Entre otras funciones, brinda la posibilidad de gestionar las vacaciones de los empleados de forma mensual, bimensual o trimestral. También se puede efectuar un control por

departamentos e incluso visualizar inmediatamente los retrasos, ausencias o sanciones de los empleados. (6)

**@GestRRHH v2.0:** Sistema diseñado para la gestión de Recursos Humanos compuesto de los siguientes módulos: Agenda, Plantilla, Actividades y Nóminas. Sus principales funcionalidades son: permite la realización de un gran número de informes, cada módulo posee un color individual para mejorar la información y permite la copia de seguridad automática. (7)

No es recomendable utilizar estos sistemas, porque son propietarios y no son multiplataforma, ellos, ofrecen una visión de cómo es gestionada la información de los recursos humanos de forma general y ayuda al entendimiento de los procesos que estos implementan, pero no se ajustan a las políticas y necesidades del sistema a desarrollar.

### **1.2.2 Nivel nacional**

En Cuba existen sistemas de Registro y Control de Personal para tener bien definido todo su capital humano entre los que se pueden citar los siguientes: Sistema para la Gestión de los Colaboradores de la Salud, Sistema de Registro Personal de Salud, Sistema de Selsoft y el Sistema Fastos. A continuación se explica algunas características de ellos.

**Selsoft:** Producto diseñado para la evaluación integral de los requisitos exigidos por el perfil psicológico en el proceso de Selección de Personal y Evaluación de los puestos de trabajo. La elaboración de los perfiles psicológicos se realiza con el usuario. La herramienta ha sido desarrollada utilizando el Lenguaje Visual Basic Versión 3.0 y Bases de Datos en Access 2.0, soportados sobre el sistema Windows 95. (6)

**Registro Personal de la Salud:** Basado en software libre y en las nuevas tecnologías de Internet como XML, PHP y Servicios Web que garantizan una explotación integrada y compatible. Utiliza como soporte de esta aplicación la infraestructura de INFOMED. Cumple con la arquitectura definida por el Grupo de Arquitectura MINSAP–Ministerio de Informática y las Comunicaciones (MIC), desarrollando productos informáticos sobre una arquitectura basada en componentes y orientada a servicios, haciendo uso de la plataforma de servicios (plaser) y los Servicios Web. (8)

Este sistema tiene las siguientes características:

- Almacena su información en una base de datos única, sólida y confiable.
- Realiza la actualización del RPS (profesionales y técnicos) de manera rápida y segura.
- Permite gestionar al personal de salud pendiente de ubicar en una unidad de salud, abarcando los movimientos de traslado, jubilación y baja, así como los movimientos de los recién graduados.
- Realiza búsquedas de profesionales por sus características, que pueden ser: nombre, tipo de profesional, nivel técnico, etc.

**Sistema Fastos:** Es otro sistema en el ámbito nacional, que tiene como función la gestión de los recursos humanos, conformado por módulos que ejecutan los procesos y operaciones de: registro de empleados, control de la plantilla, control de asistencia, informes y modelos. (9)

Estas aplicaciones no se utilizan porque algunas no son multiplataforma, además no cumplen con las características del sistema a desarrollar, pero es bueno aclarar que se usa el RPS como sistema externo que brinda cierta información acerca del personal de salud que se gestiona.

**Sistema para la Gestión de los Colaboradores de la Salud:** Este sistema tiene como función la gestión de la información de los colaboradores que van a cumplir misión médica a un país determinado. Es utilizado por la UCCM y es una aplicación desarrollada con un lenguaje de programación Delphi, con la base de datos implementada en Access 97 sobre una plataforma Windows, fue creado en el 2004 por un equipo de Informática de la provincia Santiago de Cuba. La aplicación permite gestionar misiones, pero no con datos tan específicos, como son: las localidades donde presta servicio el colaborador, agrupado por áreas geográficas y grupos de áreas geográficas, el fin de misión no maneja todas las causas ni características del proceso, de igual forma ocurre en los movimientos. Todo esto trae consigo que algunos reportes que genera la aplicación muestran datos incorrectos e incompletos. (10)

En el caso de esta aplicación es importante destacar que es uno de los antecedentes del sistema a desarrollar, es la única aplicación con que cuenta la UCCM en la actualidad, tiene estrecha relación con la situación problemática planteada, pero no es óptimo utilizarlo porque no se ajusta a las necesidades actuales de la UCCM, las tecnologías utilizadas son obsoletas, no se integra a los servicios del RIS, además no es multiplataforma.

### 1.2.3 Universidad de las Ciencias Informáticas

En la universidad existen algunos sistemas que gestionan información de personas, dentro de los que se pueden citar:

**ASSETS (Sistema de Registro Integral):** Sistema que registra todo el personal que es plantilla en la UCI, este sistema es del tipo de Planificación de Recursos Empresariales (ERP), de factura foránea, se introdujo en Cuba desde el año 1997. Controla plenamente los recursos laborales, dígase: empleados, estructura organizativa de la entidad y plantilla. Es válido aclarar que este sistema no se desarrolló en la UCI, sino que se modificó teniendo en cuenta el personal que se gestiona en la universidad que es propio solo para la misma.

**El Sistema de Registro de Personal:** Está enmarcado en el control del personal externo de la UCI. Fue desarrollado con lenguaje de programación PHP, PostgreSQL y Apache. Las funcionalidades que brinda el sistema son:

- Autenticar usuario.
- Gestionar Persona.
- Gestionar Áreas Tercerizadas.
- Gestionar Atributos por categorías.
- Gestionar categorías de personas.
- Gestionar Funcionalidades.
- Gestionar Roles.
- Mostrar Roles.
- Verificar Personas Registradas.

También se desarrolló un trabajo de tesis en la Universidad específicamente en la facultad 7 en colaboración con la empresa especializada en Soluciones Informáticas (SOFTTEL), utilizando tecnología Web. Pero este no se adapta a los lineamientos de arquitectura definidos por el Centro de Informática Médica. Esta aplicación es otro los antecedentes del sistema a desarrollar, y mantiene un modo de interactuar muy similar al que se necesita, solo que no posee la trazabilidad y la auditoría necesaria, además de todos los problemas planteados anteriormente en la introducción.

### **1.3 Tecnologías, Metodologías, Herramientas y Lenguajes utilizados**

#### **1.3.1 Integración de Modelos de Madurez de Capacidades (CMMI)**

CMMI es un modelo de mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software. 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, siendo esta última la más común y la utilizada para el desarrollo del presente trabajo.

Los niveles de CMMI son:

1. Inicial.
2. Administración básica de proyectos.
3. Proceso estandarizado.
4. Proceso administrado cuantitativamente.
5. Proceso en mejora continua. (11)

El objetivo que persigue el nivel 2 de madurez de CMMI que será el nivel utilizado para el desarrollo del presente trabajo, es 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). (12)

El área de proceso que se utilizará para el desarrollo del trabajo es REQM, 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. (13)

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, formando el Proceso Unificado de Desarrollo (RUP), parte de este grupo, ya que es una eficiente metodología probada y muy utilizada en la universidad. El trabajo en conjunto de CMMI como plan de mejora, y RUP como metodología de apoyo de este, guían a la organización para la realización del producto. RUP por su parte utiliza UML como lenguaje para modelar todos los artefactos de los flujos de trabajo.

### **1.3.2 Proceso Unificado de Desarrollo (RUP)**

El Proceso Unificado de Desarrollo o RUP (Rational Unified Process) es un proceso de desarrollo de software, una forma disciplinada de asignar tareas y responsabilidades en una empresa de desarrollo (quién hace qué, cuándo y cómo) que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, tipos de organizaciones, niveles de amplitud y tamaños de proyectos. El Proceso Unificado está basado en componentes y utiliza el UML (Lenguaje Unificado de Modelado) para preparar todos los esquemas de un sistema de software. UML es una parte esencial de RUP, sus desarrollos fueron paralelos. No obstante, los verdaderos aspectos definitorios del proceso unificado se resumen en características fundamentales:

Dirigido por casos de uso: Los Casos de Uso son el instrumento para validar la arquitectura del Software y extraer los casos de prueba.

Centrado en la arquitectura: Los modelos son proyecciones del análisis y el diseño constituye la arquitectura del producto a desarrollar.

Iterativo e incremental: Durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo. (14)

### **1.3.4 Leguaje Unificado de Modelado (UML)**

UML v2.0 es un lenguaje de modelado visual, su función principal es especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. UML usa procesos de otras metodologías, aprovechando la experiencia de sus creadores, eliminó los componentes que resultaban de poca utilidad práctica y añadió nuevos elementos. (15)

UML es un lenguaje más expresivo, claro y uniforme que el resto definidos para el diseño Orientado a Objetos (OO), que no garantiza el éxito de los proyectos pero si mejora sustancialmente el desarrollo de los mismos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios, razón por la cual se escoge para el modelado del sistema a desarrollar.

### **1.3.5 Enterprise Architect v7.1**

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. Fue construido sobre la base del excepcional éxito de las versiones previas con un completo soporte para el estándar UML 2.1. Soporta los 13 diagramas de UML 2.1. (16)

Se utilizará esta herramienta para el modelado del sistema ya que brinda beneficios significativos para ayudar a construir modelos de sistemas de software rigurosos en un ambiente fácil de usar, rápido y flexible.

### **1.3.6 NetBeans**

El Entorno Integrado de Desarrollo (IDE) NetBeans v6.8 es una herramienta pensada para escribir, compilar, depurar y ejecutar programas. Consta de una gran comunidad de usuarios en constante crecimiento, lo que le ha permitido, al igual que muchos otros sistemas libres, el progreso paulatino de sus prestaciones y la eliminación de errores que pudiesen existir. NetBeans permite que las aplicaciones se desarrollen a partir de un conjunto de módulos o componentes de software. Brinda una barra de navegación para el acceso rápido a funciones en una clase muy extensa, detecta errores de sintaxis en tiempo real, además de un completamiento de código fuente eficiente y seguro. (17)

Esta herramienta fácil de instalar y usar será utilizada para el desarrollo del presente sistema, ya que es un producto libre y gratuito sin restricciones de uso. Permite crear aplicaciones Web con PHP 5 y además incluye soporte para Symfony y AJAX, lo que la hace aun más popular entre los desarrolladores.

### **1.3.7 Hypertext Pre-processor (PHP).**

PHP es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y enorme volumen de documentación. Diseñado originalmente para la creación de páginas Web dinámicas. Permite soporte para gran cantidad de sistemas gestores de bases de datos como InterBase, MySQL, PostgreSQL, Oracle, Informix, DB2 y Microsoft SQL Server. Permite a la mayoría de los programadores crear aplicaciones complejas con un tiempo de aprendizaje muy corto. También les facilita involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones, puede ser interpretado por los servidores Web más populares. (18)

El sistema será desarrollado completamente en PHP v5.2.6, usando el Protocolo Simple de Acceso a Objetos (por sus siglas en inglés, SOAP) como mecanismo para el intercambio de información del cliente con el servidor. Se decidió utilizar dicho lenguaje por tener una mayor facilidad de uso con respecto a PHP v5.1.4 utilizado en el sistema anterior, y presenta además grandes ventajas con respecto a Delphi, que fue el lenguaje utilizado en el primer sistema probado en la UCCM.

### **1.3.8 JavaScript**

JavaScript es un lenguaje que no requiere de compilación ya que funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos. JavaScript puede crear diferentes efectos e

interactuar con los usuarios. 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. Se decide la utilización de JavaScript como lenguaje del lado del cliente por las siguientes razones:

- Es interpretado (no compilado) por el cliente, es decir, directamente del programa fuente se pasa a la ejecución de dicho programa, con lo que al contrario de los lenguajes compilados no se genera ni código objeto ni ejecutable.
- Está basado en objetos. No es un lenguaje de programación orientada a objetos (OOP). JavaScript no emplea clases ni herencia, ni otras técnicas típicas de la OOP.
- No es necesario declarar los tipos de variables que van a utilizarse ya que realiza una conversión automática de tipos.
- Las referencias a objetos se comprueban en tiempo de ejecución. Esto es consecuencia de que JavaScript no es un lenguaje compilado.
- No puede escribir automáticamente al disco duro. Por eso se dice que JavaScript es un lenguaje seguro para el entorno de Internet en el que se aplicará. (19)

### 1.3.9 Ajax

El acrónimo AJAX viene de Asynchronous JavaScript And XML, es un conjunto de tecnologías que permite al navegador Web obtener datos del servidor y refrescar partes de una página Web asíncronamente. Utiliza otras tecnologías que ya existen: HTML, Java Script y XML. Básicamente el objetivo de utilizar AJAX es para realizar Web más usables y dinámicas. Se pueden hacer aplicaciones Web en tiempo real. Se ha demostrado también que, Ajax, es válida para múltiples navegadores dado que está basado en estándares abiertos como Java Script y Document Object Model (DOM). Se decide la utilización de Ajax como lenguaje del lado del cliente para validaciones y envío de datos a través de las páginas. (20)

### 1.3.10 Arquitecturas del Sistema

**MVC (Modelo-Vista-Controlador):** El MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde:

- La vista 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.
- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. (21)

Se decide la utilización del MVC por la facilidad que brinda para la creación de sistemas Web, además de la organización que provee en el código fuente por lo que se consigue un mantenimiento más sencillo de las aplicaciones.

**Cliente-Servidor:** La arquitectura Cliente/Servidor es aquella en la que influyen una serie de aplicaciones basadas en dos categorías que cumplen funciones diferentes (una requiere servicios y la otra los brinda) pero que a la vez, pueden realizar tanto actividades en forma conjunta como independientemente ya que el cliente también tiene sus propias aplicaciones, archivos y bases de datos y que además, pueden estar en la misma plataforma o en plataformas diferentes. Por otra parte, el servidor puede brindar varios servicios a la vez, tanto al mismo cliente como a clientes múltiples. Como ventajas principales del uso de esta arquitectura se tiene el aumento de la productividad, se disminuyen los costes de operación y mejora en el rendimiento de la red. (22)

### 1.3.11 Sistema Operativo

**Linux:** Linux es un sistema operativo libre tipo Unix para computadoras personales (PC), servidores, y estaciones de trabajo. Como sistema operativo, Linux tiene un buen diseño, es multitarea, multiusuario y multiprocesador.

**Ubuntu:** Es una distribución GNU/Linux que ofrece un sistema operativo predominantemente enfocado a ordenadores de escritorio aunque también proporciona soporte para servidores. Basada en Debian GNU/Linux, Ubuntu concentra su objetivo en la facilidad y libertad de uso, los lanzamientos regulares (cada 6 meses) y la facilidad en la instalación. Está totalmente basado en los principios del desarrollo de software de código abierto; se anima a que la gente use, mejore y distribuya software de código abierto razones por las que se utilizará esta distribución en su versión 9.4 para el desarrollo del presente sistema. (23)

### 1.3.12 Sistema Gestor de Base de Datos (SGBD)

Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso, puede ser considerado una colección de datos variables en el tiempo. Un SGBD es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad.

**PostgreSQL:** Es un sistema gestor de base de datos relacional orientada a objetos y libre. Es un potente motor de bases de datos, que tiene prestaciones y funcionalidades equivalentes a muchos gestores de bases de datos comerciales.

Dentro de sus principales ventajas se encuentran:

- Soporta distintos tipos de datos
- Posee una gran escalabilidad, haciéndolo idóneo para su uso en sitios Web que atienden un gran número de solicitudes.
- Puede ser instalado un número ilimitado de veces sin temor de sobrepasar la licencia.
- Posee estabilidad y confiabilidad legendaria.
- Es extensible a través del código fuente disponible sin costos adicionales.
- Es multiplataforma, disponible en Linux, \*nix, Mac Os X y Windows, entre otros sistemas operativos.

Se seleccionó PostgreSQL v8.3 teniendo en cuenta la necesidad de utilizar herramientas libres, para el desarrollo, además de que es un gestor multiplataforma, confiable, estable, con gran escalabilidad, control de concurrencia y funcionalidades que lo destacan como uno de los SGBD más potentes en la actualidad.

(24)

### 1.3.13 Servidor Web

Un servidor Web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. El servidor Web se encarga de contestar a estas peticiones de forma adecuada, integrando como resultado una página Web.

**Apache:** Es uno de los servidores Web más populares del mercado, y el más utilizado actualmente, de código abierto y gratuito, disponible para Windows y GNU/Linux, entre otros. Apache presenta entre otras

características, mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. Apache tiene amplia aceptación en la red: desde 1996, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios Web en el mundo. (25)

En el presente sistema se utilizará como servidor Web Apache 2.2, porque 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.

### **1.3.14 Navegadores**

**Internet Explorer:** Tiene una forma rápida y eficiente de navegar por la Web, que le permite abrir varias páginas en una misma ventana dentro de pestañas separadas. Abre 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. Solo puede ejecutarse en Windows y en Mac, pero no soporta las mismas características en ambas plataformas.

**Mozilla Firefox:** Es un navegador de Internet, libre, de código abierto y con gran potencia, con interfaz gráfica de usuario, desarrollado por la Corporación Mozilla y un gran número de voluntarios externos. Cualquier usuario corriente de PC puede usar el navegador sin tener la necesidad de cambiar su sistema operativo. Firefox, es multiplataforma y está disponible en versiones para Microsoft Windows, Mac OS X y GNU/Linux. (26)

El sistema a realizar puede ser visualizado en cualquiera de los dos navegadores estudiados anteriormente pero es recomendable utilizar Mozilla Firefox ya que se encuentra disponible para distintas plataformas, utilizando las mismas características en cada una de ellas.

### **1.3.15 Servicios Web**

Los Servicios Web permiten que las aplicaciones compartan información y que además invoquen funciones de otras aplicaciones independientemente de cómo se hayan creado estas, cuál sea el sistema

operativo o la plataforma en que se ejecutan y cuáles son los dispositivos utilizados para obtener acceso a ellas.

Ventajas de los Servicios Web:

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Permiten la interoperabilidad entre plataformas diferentes por medio de protocolos estándar.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Los Servicio Web permiten agregar a tus programas, funciones imposibles de contemplar bajo el uso de rutinas de librerías, como por ejemplo, incorporar un buscador de páginas Web, además la carga de CPU que supone la ejecución de una rutina, desaparece al usar Servicios Web. (27)

### 1.3.16 Framework de desarrollo

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los procesos más utilizados para resolver tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

**CodeIgniter:** Es un framework utilizado por una gran comunidad de usuarios. Construido para codificadores PHP que necesitan una herramienta de desarrollo fácil para crear aplicaciones Web simples y elegantes. Es rápido pero no muy fácil, ya que carece de algunas librerías que los otros framework poseen.

Ventajas:

- Amplia documentación disponible.
- Ligerero, y sin instalación (para comenzar a desarrollar una aplicación basta con copiar los archivos, y ponerse a trabajar).
- Compatibilidad con una amplia variedad de servidores y configuraciones (la aplicación se concebiría para ejecutarse en un hosting compartido con otros clientes, y con relativamente poca posibilidad de configuración).

- Flexibilidad, ya que no obliga a tener una determinada estructura de tablas, nombres de campos, ni adherirse a una forma de programar concreta.

Desventajas:

- Curva de aprendizaje: necesidad de aprender nuevas funciones, estructuras y métodos de programación.
- Dificultad para adaptar el código escrito en PHP tradicional. (28)

**CakePHP:** Es un framework para PHP que permite programar más rápido evitando escribir código tedioso de tareas muy comunes.

Ventajas:

- Tiene una comunidad muy activa. Licencia flexible. CakePHP está distribuido bajo la MIT LicenseIP limpia .Cada línea de código está escrita por el equipo de desarrollo de CakePHP. Extremadamente simple.
- Desarrollo rápido.
- Buenas prácticas. Cake es muy fácil de entender y cumple los estándares en seguridad y autenticación, manejo de sesiones y muchas otras características.
- Orientado a Objetos.

Desventajas:

- Ninguna oferta de formación está disponible para este framework.
- No ofrecen más apoyo que foros, listas de correo y Google Grupos.
- Ninguna protección por definición de la licencia MIT. (29)

**Symfony:** Es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación Web. Symfony 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 plataformas Windows. Soporta AJAX y plantillas.

Sus características son:

- Fácil de instalar y configurar en la mayoría de plataformas.
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la Web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo. (30)

Se decide usar Symfony v1.4.3 como framework para el desarrollo del sistema ya que su licencia es de tipo software libre, es uno de los framework PHP más populares entre los usuarios y las empresas, ya que permite que los programadores sean mucho más productivos y a la vez se crea código de más calidad y más fácil de mantener y actualizar. Symfony es maduro, estable, profesional y está muy bien documentado.

**ExtJS:** Para la realización del sistema se utilizará EXTJS 3.0.1 como framework de diseño visual, puesto que posee código reutilizable, es un framework de código limpio, usando tecnología AJAX, escrito en Java Script con soporte para la mayoría de los navegadores, presenta cientos de funcionalidades que hacen de él una herramienta indispensable, ya que junto con el ahorro de tiempo y de líneas de código hace que el usuario se sienta a gusto pues presenta una interfaz amigable. Contiene estilos CSS predefinidos aunque es totalmente configurable. Posee 2 licencias, una comercial y otra Open Source.

Ventajas de EXT:

- Código reutilizable.
- Independiente o adaptable a framework diferentes (prototipo, jquery, YUI)
- Orientada a la programación de interfaces tipo desktop en el Web.

- El API es homogeneizado independientemente del adaptador usado. Los controles siempre se verán igual.
- Soporte comercial.
- Una extensa comunidad de usuarios. (31)

### **Conclusiones**

Una vez concluido este capítulo, después de haber realizado un profundo estudio teórico y del estado del arte, referente a las tecnologías y las tendencias del mundo actual para respaldar la labor a realizar, se llegó a las siguientes conclusiones:

- Los métodos de trabajo, las tecnologías y software utilizados deben ser adaptados a las necesidades del centro.
- De los lenguajes de programación para construir una aplicación Web, PHP es una buena variante y junto a JavaScript ofrece grandes ventajas y potencialidades, para el almacenamiento de los datos se aprobó PostgreSQL por las potencialidades que ofrece la combinación con el lenguaje seleccionado.
- Después del análisis realizado acerca de los sistemas existentes relacionadas con la problemática planteada, ninguno cumple con las características que se requieren para ser utilizado.
- Para llevar a cabo la automatización del proceso de gestión de los colaboradores de la salud en la UCCM; se hace necesario la implementación de una herramienta Web haciendo uso de las herramientas y tecnologías mencionadas que permita gestionar y almacenar toda esta información satisfactoriamente.

### **Capítulo 2: Características del sistema**

En el presente capítulo se realiza un análisis de todo lo relacionado con las características del sistema, primeramente se realiza una propuesta del sistema, se identifican los actores, trabajadores y casos de uso del negocio. Se definen los requisitos funcionales, no funcionales y se modela el sistema. Se expone además, el estudio realizado acerca del Módulo de la Gestión de la Información de los colaboradores. Se detalla la forma en la que se realizan las labores que serán objeto de automatización para determinar con mayor claridad las responsabilidades llevadas a cabo.

#### **2.2 Propuesta del Sistema**

El producto a desarrollar tiene como objetivo gestionar todo el personal médico y no médico que pudiera salir a cumplir misión hacia el exterior, para ello, la UCCM mantiene una relación estrecha con todos los municipios y provincias del país. Razón que hace necesario integrarse a los diferentes registros ya disponibles en el RIS, entre ellos se encuentran, RC, RPS, RL y SAAA. La información que se gestionará previamente a través del sistema permitirá organizar las misiones por países, y el control de los movimientos realizados a los colaboradores en el transcurso de las mismas (salida de misión, baja de misión, vacaciones, deserciones).

Se propone desarrollar una aplicación que mantenga un modo de interactuar lo más similar posible a la de la aplicación que se usa actualmente en la UCCM, mejorando la trazabilidad y la auditoría con una arquitectura más robusta, tomando en cuenta que la tecnología Web presupone algunas limitaciones en cuanto a la rapidez con que el usuario interactúa con la interfaz.

Los diferentes niveles tales como el municipio, la provincia y la nación, u otro autorizado, podrán conectarse a los servidores centrales alojados en INFOMED para recuperar información sobre las misiones o colaboraciones, de acuerdo con los permisos y privilegios de acceso en cada caso.

#### **Descripción de la propuesta del sistema**

El sistema que se propone en el trabajo se encontrará brindando servicio sobre la red telemática de salud cubana INFOMED, a través de ella todos los niveles (municipio, provincia, nación) accederán al sistema. Para mayor comprensión ver imágenes de: [Anexo #1](#).

### 2.3 Modelo de negocio

El modelo de negocio describe las funcionalidades que la empresa desea desarrollar y la relación de los clientes con él, es decir, ayuda a comprender cómo interactúan los actores del negocio con los casos de uso del mismo.

#### Actores del negocio

El actor del negocio es aquella persona, sistema o entidad externa que interactúa con el negocio y se beneficia de sus resultados. El término actor significa el rol que algo o alguien juega cuando interactúa con el negocio, por tanto, no representa un usuario físico.

| Actor       | Descripción   |
|-------------|---|
| Aspirante   | Es el principal beneficiado de los resultados del negocio .El más interesado en la realización del expediente y la asignación de la misión. |
| Colaborador | Es el principal beneficiado de los resultados del negocio .El más interesado en la realización de algún movimiento.                         |

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

#### Trabajadores del negocio

Los trabajadores del negocio son aquellas personas o sistemas dentro de la frontera del negocio que son las que realizan las actividades que están comprendidas dentro de los casos de uso y no se benefician del resultado de las mismas.

| Trabajador   | Descripción  |
|--|--|
| Especialista de colaboración médica en el municipio. | Se encarga de entregar al Dpto. de RRHH de la UPCM, parte de los datos del recurso humano. |
| Especialista de colaboración médica de la Provincia  | Esta es la persona encargada de confeccionar el expediente del aspirante a colaborador.    |
| Especialista de colaboración médica en la Nación.    | Está encargado de realizar todas las actividades pertinentes a un aspirante a colaborador. |

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

**Diagrama de casos de uso del negocio**

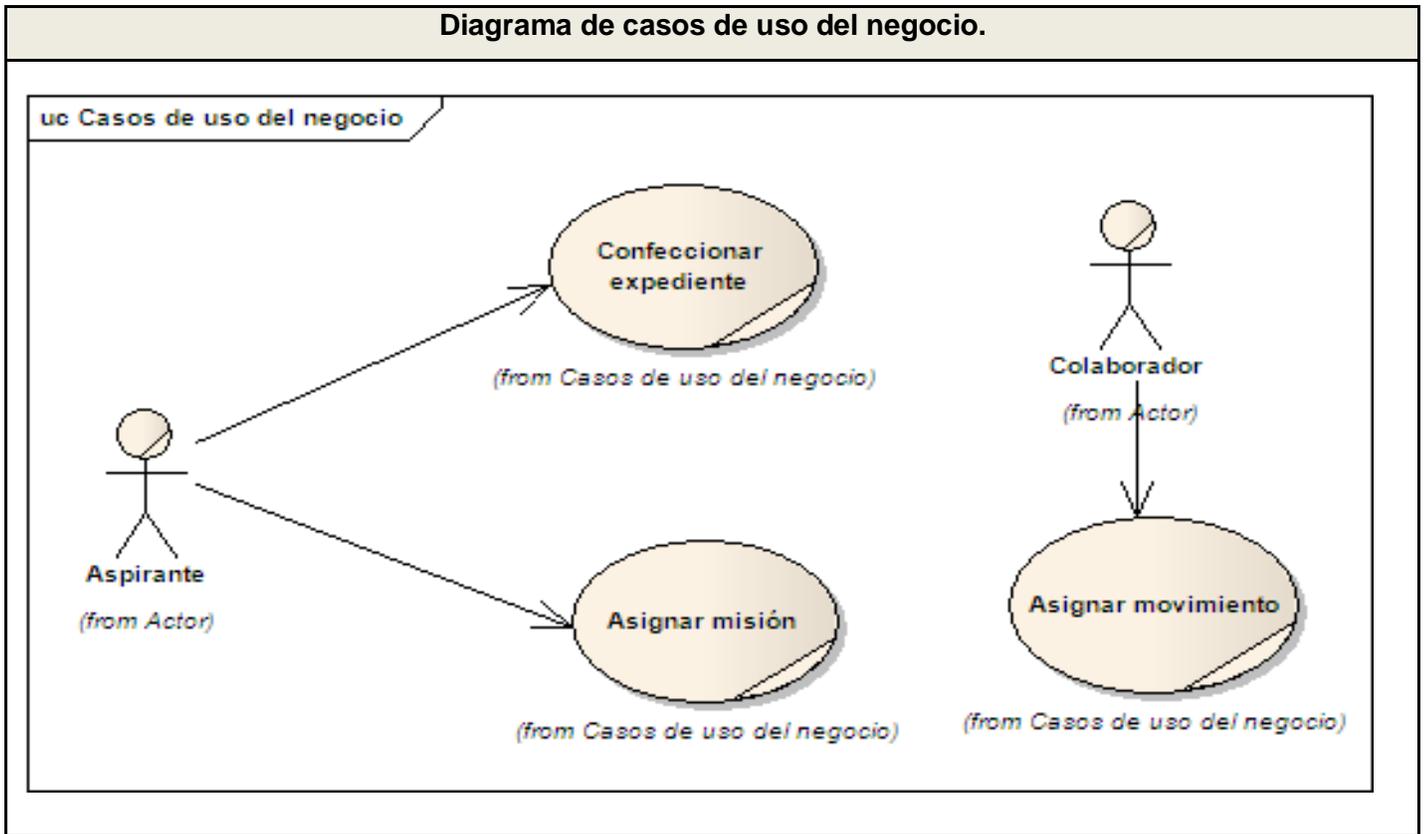


Fig. 1 Diagrama de casos de uso del negocio.

**Descripción de los casos de uso del negocio**

| Caso de Uso del Negocio | Confeccionar Expediente del Aspirante a Colaborador.   |
|-------------------------|--|
| Actor                   | Aspirante  |
| Resumen                 | El Caso de Uso inicia cuando el Aspirante solicita la confección de su expediente, y el especialista de colaboración médica del municipio envía todos los datos de este a la provincia si estos datos están correctos es expediente es confeccionado por el especialista de colaboración médica en la provincia. |
| Casos de Uso asociados  |  |

Tabla #3. Descripción del CU Confeccionar Expediente del Aspirante a Colaborador.

| Caso de Uso del Negocio       | Asignar Misión  |
|-------------------------------|---|
| <b>Actor</b>                  | Aspirante   |
| <b>Resumen</b>                | El Caso de Uso inicia cuando el aspirante solicita la asignación de una misión, esta es atendida por el especialista de colaboración médica en la nación, este revisa el expediente del aspirante solicitante y ejecuta la acción en caso de que el expediente esté correcto. |
| <b>Casos de Uso asociados</b> |   |

Tabla #4. Descripción del CU Asignar misión.

| Caso de Uso del Negocio       | Realizar Movimiento  |
|-------------------------------|--|
| <b>Actores</b>                | Colaborador  |
| <b>Resumen</b>                | El Caso de Uso inicia cuando el Colaborador solicita un movimiento, es atendido por el especialista de colaboración médica en la nación este revisa la solicitud para ver si puede realizar este movimiento al colaborador solicitante en caso correcto ejecuta la acción. |
| <b>Casos de Uso asociados</b> |  |

Tabla #5. Descripción del CU Manejar acciones del Aspirante.

### Diagramas de actividades del Negocio

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.

Los diagramas de actividades se muestran en el [Anexo #2](#) del documento.

### Modelo de Objetos

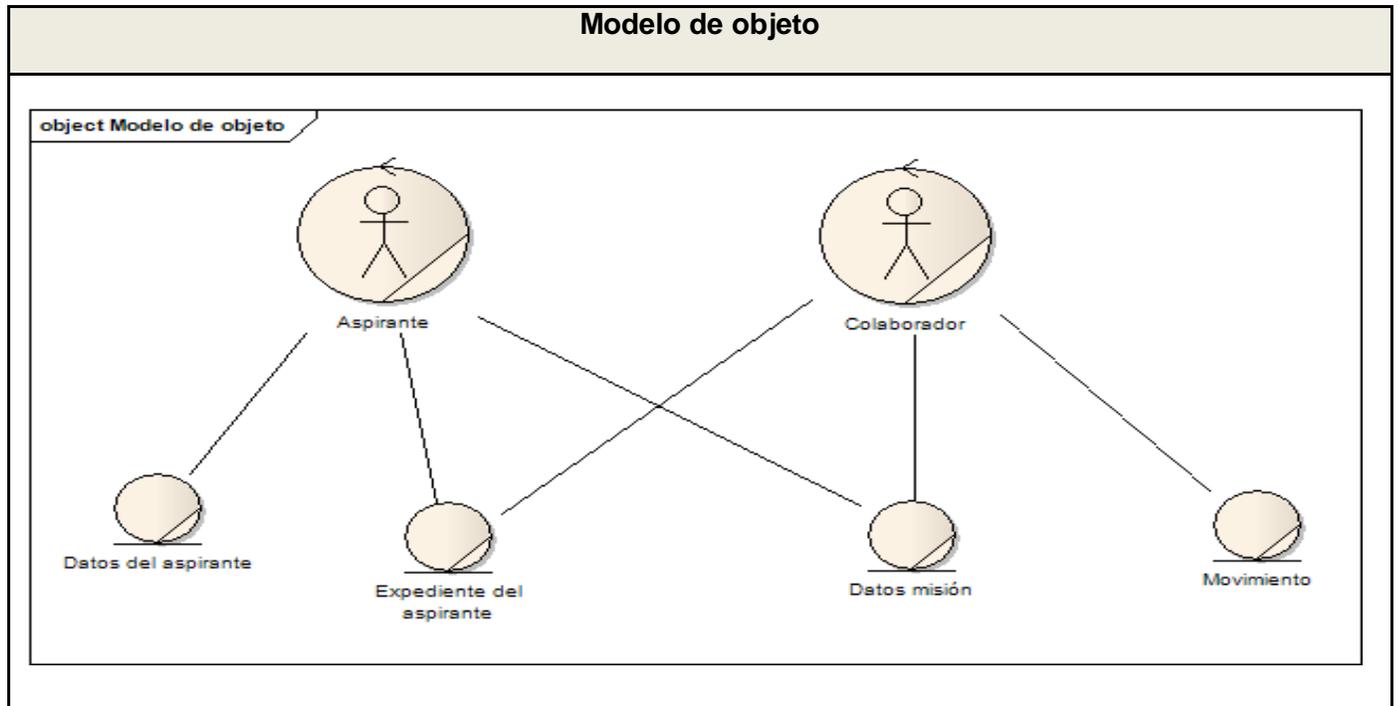


Fig. 2 Modelo de objeto.

### 2.4 Especificación de los requisitos

Los requisitos son cualidades o funcionalidades que el sistema debe cumplir para que tenga un correcto funcionamiento. Una buena captura de requisitos es fundamental para desarrollo exitoso de la aplicación, pues ellos proveen al equipo de desarrollo de un entendimiento específico y común de las cualidades del sistema. Los mismos tienen dos tipos de clasificación:

- **Requisitos funcionales:** Se basa en los casos de uso que describen de qué forma el usuario va a utilizar el sistema. Cada usuario requiere de varios casos de uso. Los analistas proponen cómo será la interfaz del sistema esbozando varias versiones para que el usuario decida.
- **Requisitos NO funcionales:** Especifica las propiedades del sistema que tienen que ver con rendimiento, velocidad, uso de memoria, plataforma, entre otras condiciones. Además, imponen condiciones a requisitos funcionales.

### Los requerimientos del sistema se resumen en:

- **RF 1** Autenticar.
  - El usuario introduce sus datos personales, en caso de ser correctos obtiene los permisos necesarios para la utilización del sistema y de los servicios Web consumidos por el mismo, en caso contrario, se le notifica para que corrija sus datos de entrada. Importante destacar que este servicio es brindado por el SAAA.
- **RF 2** Buscar colaborador.
  - Para buscar se debe especificar si ha cumplido alguna misión anteriormente, en caso de que si busca en la BD del sistema sino, en un registro en dependencia del ministerio al que pertenezca. Luego de seleccionar los criterios de búsqueda muestra un listado de colaboradores.
- **RF 3** Mostrar datos del colaborador.
  - Buscar colaborador y después de seleccionarlo muestra una ficha con todos los datos del colaborador, dando la posibilidad de imprimirla.
- **RF 4** Crear expediente del aspirante a colaborador.
  - Buscar colaborador y después de seleccionarlo en caso de que ya tenga expediente, se da la posibilidad de crear otro modificando los datos del anterior, en caso de que no posea ninguno, se debe mostrar el formulario para crearlo nuevo completamente.
- **RF 5** Actualizar expediente del colaborador.
  - Buscar colaborador, seleccionarlo y mostrar su expediente para modificar sus datos.
- **RF 6** Buscar expediente
  - Buscar colaborador, seleccionarlo y muestra los datos del expediente (sin derecho a modificarlo) con la opción de imprimir.
- **RF 7** Asignar misión.
  - Buscar colaborador y después de seleccionarlo se debe mostrar el formulario para asignar la misión. Incluye las funcionalidades del caso de uso buscar ciudadano.
- **RF 8** Actualizar misión.
  - Buscar colaborador y después de seleccionarlo se debe mostrar el formulario para actualizar la misión.
- **RF 9** Buscar misión.

- Buscar colaborador, seleccionarlo y muestra la lista de misiones asignadas al colaborador.
- **RF 10** Realizar movimientos.
  - Buscar colaborador, seleccionarlo y mostrar la interfaz para realizar el movimiento.
- **RF 11** Buscar movimientos.
  - Buscar colaborador, seleccionarlo y obtener el listado de las misiones, seleccionar la misión deseada y muestra todos los movimientos realizado en la misión.
- **RF 12** Modificar designado.
  - Buscar colaborador, seleccionarlo y muestra los datos del designado actual dando la posibilidad de modificarlo. Incluye las funcionalidades del caso de uso buscar ciudadano.
- **RF 13** Buscar designado.
  - Buscar colaborador, seleccionarlo y muestra los datos del designado actual dando la posibilidad de imprimir sus datos.
- **RF 14** Listar colaborador
  - Consiste en mostrar un grupo de colaboradores que cumplan con los criterios de una determinada búsqueda de los mismos que realizó.
- **RF 15** Buscar ciudadano
  - Consiste en buscar una persona en el registro de ciudadano que sería el designado correspondiente al colaborador.
- **RF 16** Buscar aspirante a colaborador
  - Consiste en buscar la persona a la que se le va a crear un nuevo expediente y que aún no es colaborador porque no se le ha asignado ninguna misión.

**Los requerimientos no funcionales quedan se resumen en:**

- **RNF 1 Requerimientos de Comunicación**
  - Protocolo de red TCP/IP.
- **RNF 2 Requerimientos de Software**

Para PC cliente:

- Los clientes tendrán acceso a través de cualquier navegador Web: Mozilla v3.6.2, Internet Explorer 4.0 o superior. Debe tener instalado el FlashPlayer v10.0 para visualizar las imágenes.

Para PC servidor:

- Se desarrollará con el lenguaje de programación PHP 5.0. Como Gestor de Base de Datos PostgreSQL, Apache como el servidor Web sobre el que correrá la aplicación. Sistema Operativo en el Servidor es Linux Distribución Debian v5.0.

- **RNF 3 Requerimientos de Rendimiento**

- Las páginas de la aplicación deberán reducir al máximo el uso de imágenes y componentes que ocupen gran capacidad dentro de la transmisión de la información, para garantizar una respuesta más rápida del sistema.

- **RNF 4 Requerimientos de Confidencialidad**

- Los administradores del sistema son los únicos que podrán transformar la información, los operadores solo podrán ver la información.

- **RNF 5 Requerimientos de Extensibilidad**

- Se garantizará que el sistema sea compatible con el resto de los módulos que se realizarán.

- **RNF 6 Requerimientos de Mantenimiento**

- El sistema estará bien documentado de forma tal que el tiempo de mantenimiento sea mínimo en caso de necesitarse.

- **RNF 7 Requerimientos de Interfaz externa**

- El diseño de la interfaz será sencillo y claro de usar, formal, serio y con una navegación sugerente, todo esto teniendo en cuenta el fin con el que se desarrollará la aplicación. Los colores usados serán tenues y agradables a la vista.

- **RNF 8 Requerimientos de Usabilidad**

- La aplicación será fácil de usar y fácil de aprender por la mayor parte de personas que interactúen con ella. El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y que esté autorizado a hacer uso del mismo.
- **RNF 9 Requerimientos de Hardware**
  - Se requiere de una PC con una memoria RAM de 128 o superior, de disco duro preferiblemente de 10 GB o más, procesador Pentium II o superior a 1.60GHz, impresora local o de red para imprimir los reportes.
- **RNF 10 Requerimientos de Seguridad**
  - 10.1 Se dispondrá de un mecanismo de seguridad basado en el modelo del SAAA. Actualmente, componente desarrollado y mantenido por la empresa SOFTEL.
  - 10.2 Confiabilidad: La información manejada por el sistema estará protegida de acceso no autorizado.
  - 10.3 Integridad: Se harán copias de respaldo que puedan restaurar el sistema en caso de pérdida de información.
  - 10.4 Disponibilidad: Se utilizará servicio de autenticación para acceso a varios servicios en el sistema. Solo los usuarios autorizados tendrán acceso a la información en todo momento.
- **RNF 11 Requerimientos Legales**
  - El sistema se basará en un estándar que se rige por leyes internacionales y cumple con las normas establecidas por el país.

### 2.5 Descripción del Sistema

| Actores                        | Justificación   |
|--------------------------------|---|
| Editor provincial              | Es la persona que ocupa el cargo de estadístico provincia el cual cuenta los privilegios de gestionar los datos del expediente de un aspirante a colaborador en el sistema.   |
| Editor nacional                | Gestiona toda la información posible del aspirante a colaborador, sus misiones y movimientos.   |
| Visualizador                   | Los visualizadores son especialistas en cada uno de los niveles que tendrán el privilegio de obtener información del sistema como Buscar Colaborador, Buscar Misión, Buscar Movimientos e Imprimir, correspondiente con el nivel al que pertenezca. |
| SAAA                           | Este registro dará la seguridad requerida al sistema, por el cual el usuario se va a poder autenticar, y tendrá la autorización y auditoria necesaria.  |
| Registro de Localidad.         | Este registro se utiliza para obtener la provincia, el municipio, localidades, calles y manzanas del país, ya sea para ubicar una unidad de salud determinada o para registrar la dirección de un colaborador.                                      |
| Registro del Ciudadano         | Este registro se utiliza para acceder a los datos personales de todos los ciudadanos del país.  |
| Registro Personal de la Salud. | Gestiona un conjunto de datos del personal que labora en el sector de la salud, ya sea profesional o no profesional.  |

Tabla #6. Descripción del Sistema.

2.6 Diagrama de Casos de Uso del Sistema

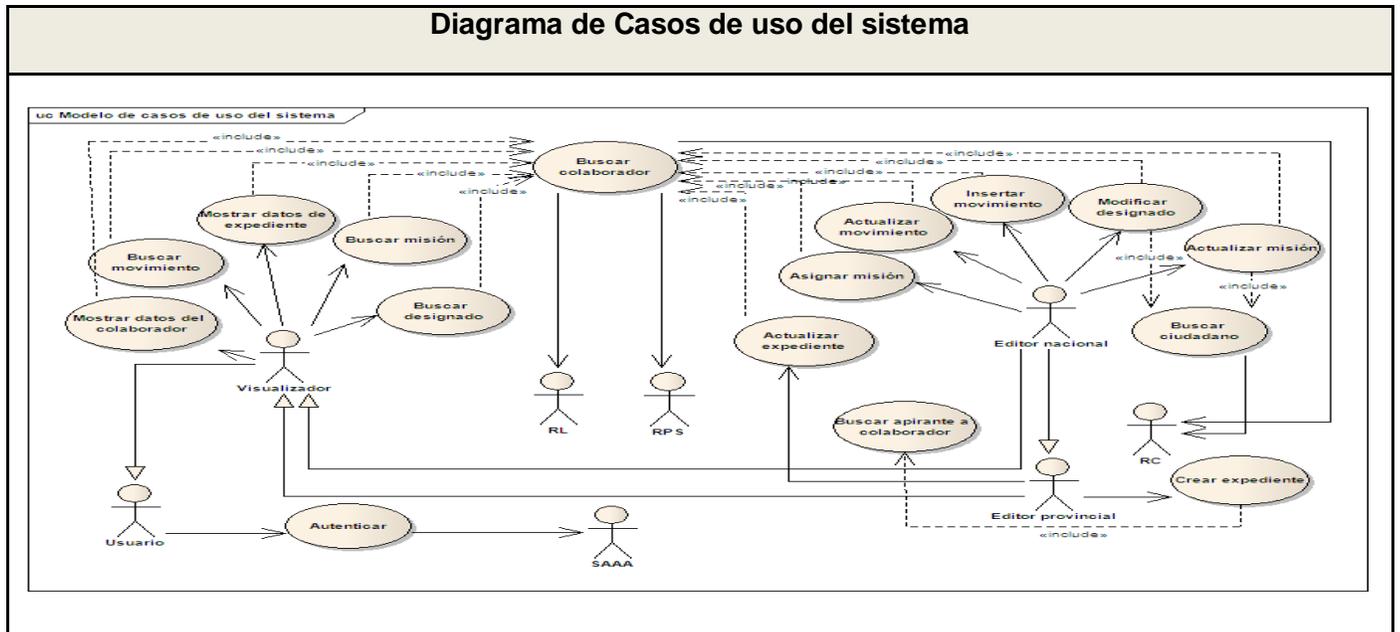


Fig. 3 Diagrama de CUS.

2.7 Especificación de los Casos de Uso

CUS Autenticar

|                               |  |
|-------------------------------|--|
| <b>Objetivo</b>               | Conocer los datos de la persona autenticada para validar sus permisos.   |
| <b>Actores</b>                | Usuario.   |
| <b>Resumen</b>                | El caso de uso se inicia cuando el usuario requiere acceso al sistema, permite que solamente los que tengan permiso puedan tener acceso a la aplicación. |
| <b>Complejidad</b>            | Alta   |
| <b>Prioridad</b>              | Crítico  |
| <b>Precondiciones</b>         | El usuario debe tener los permisos necesarios para poder acceder a la aplicación.  |
| <b>Poscondiciones</b>         | El usuario es autenticado en el sistema.   |
| <b>Requisitos Funcionales</b> | RF1  |

Tabla #7. Descripción del CUS Autenticar.

### CUS Buscar colaborador

|                               |   |
|-------------------------------|---|
| <b>Objetivo</b>               | Obtener los datos de un colaborador dado los parámetros de búsqueda para realizarle determinadas operaciones.   |
| <b>Actores</b>                | Visualizador  |
| <b>Resumen</b>                | Inicia cuando el usuario desea buscar un colaborador. 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 |
| <b>Complejidad</b>            | Alta  |
| <b>Prioridad</b>              | Crítico   |
| <b>Precondiciones</b>         | La información debe estar actualizada y disponible.   |
| <b>Poscondiciones</b>         | Se buscó colaborador dado criterios.  |
| <b>Requisitos Funcionales</b> | RF2   |

Tabla #8. Descripción del CUS Buscar colaborador.

### CUS Crear Expediente.

|                               |  |
|-------------------------------|--|
| <b>Objetivo</b>               | Crear un nuevo expediente para que todos los datos del colaborador queden especificados  |
| <b>Actores</b>                | Editor nacional  |
| <b>Resumen</b>                | El caso de usa inicia cuando el usuario va a crear un expediente. Busca el colaborador y después de seleccionarlo en caso de que ya tenga expediente o no, se debe mostrar el formulario para crearlo nuevo completamente o para crear otro modificando los datos de un expediente anterior. |
| <b>Complejidad</b>            | Alta   |
| <b>Prioridad</b>              | Crítico  |
| <b>Precondiciones</b>         | El colaborador debe ser un aspirante   |
| <b>Poscondiciones</b>         | Se creó el expediente con los datos del colaborador  |
| <b>Requisitos Funcionales</b> | RF4  |

Tabla #9. Descripción del CUS Crear expediente.

### CUS Asignar Misión.

|                 |  |
|-----------------|--|
| <b>Objetivo</b> | Asigna la misión para que el colaborador tenga su misión correspondiente |
|-----------------|--|

|                               |  |
|-------------------------------|--|
| <b>Actores</b>                | Editor Nacional  |
| <b>Resumen</b>                | Inicia cuando el usuario busca una persona que esté en la base de datos, luego de seleccionarlo, le asignará una misión asignando los datos correspondientes a la misma. |
| <b>Complejidad</b>            | Alta   |
| <b>Prioridad</b>              | Crítico  |
| <b>Precondiciones</b>         | El colaborador no debe tener ninguna misión asignada en ese momento.   |
| <b>Poscondiciones</b>         | Se le asignó la misión al colaborador.   |
| <b>Requisitos Funcionales</b> | RF7  |

Tabla #10. Descripción del CUS Asignar misión.

### CUS Realizar movimientos.

|                               |   |
|-------------------------------|---|
| <b>Objetivo</b>               | Cambiar algún dato de una misión asignada a un colaborador en caso de que estos se hayan entrado incorrectamente, o haya cambiado por alguna razón.           |
| <b>Actores</b>                | Editor Nacional   |
| <b>Resumen</b>                | Inicia cuando el usuario busca una persona que tenga una misión asignada, luego de seleccionarlo, le podrá actualizar los datos correspondientes a la misión. |
| <b>Complejidad</b>            | Alta  |
| <b>Prioridad</b>              | Crítico   |
| <b>Precondiciones</b>         | El colaborador debe tener una misión asignada.  |
| <b>Poscondiciones</b>         | Se actualizan los datos de la misión en la base de datos.   |
| <b>Requisitos Funcionales</b> | RF10  |

Tabla #11. Descripción del CUS Insertar movimiento.

### Conclusiones

En este capítulo se ha planteado la solución propuesta para llevar a cabo el desarrollo de la aplicación, se detalló la lógica del negocio y el funcionamiento de cada uno de sus procesos, de este análisis, se deriva que la implementación de una herramienta Web automatizada puede mejorar considerablemente las

deficiencias existentes en la forma en que se gestiona la información de los colaboradores de la salud en la UCCM.

A partir del análisis detallado de cada uno de los requerimientos funcionales fueron definidas las principales funcionalidades del sistema. La aplicación propuesta contará con varios tipos de usuarios que asumirán roles identificados en el diagrama de actores del sistema. Para que el sistema funcione adecuadamente debe cumplir con los requerimientos de software y hardware planteados.

### Capítulo 3: Diseño del sistema

Con el desarrollo de este capítulo se profundiza en el análisis y diseño del sistema, tiene como objetivos fundamentales: llevar los requerimientos definidos a una propuesta de diseño, que será la guía a seguir para la implementación del sistema, evolucionar hacia una arquitectura del software robusta y flexible ante la aplicación de nuevos cambios, y diseñar el sistema con un enfoque hacia el rendimiento y la eficiencia.

Se analizan los requisitos que fueron descritos, refinándolos y estructurándolos, con el objetivo de conseguir una comprensión más precisa de estos, una descripción que sea fácil de mantener y que ayude a estructurar todo el sistema, incluyendo su arquitectura.

Se toman decisiones estratégicas y tácticas para cumplir los requerimientos funcionales y no funcionales del sistema en cuestión. Es la parte del proceso de desarrollo de software cuyo propósito primario es decidir cómo se llevará a cabo la implementación del sistema.

#### 3.2 Descripción de la arquitectura

La Arquitectura de Software, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software. La arquitectura de software establece los fundamentos para que analistas, diseñadores, programadores, etc. trabajen en una línea común que permita alcanzar los objetivos del sistema, cubriendo todas las necesidades. (32)

El sistema propuesto cuenta una arquitectura basada en uno de los estilos más utilizados para el diseño Web, MVC, explicado en el Capítulo 1. El uso del framework Symfony propone construir el software bajo ese patrón. Brindándole al sistema una clara definición entre cómo se muestra la información, el manejo de las acciones que el usuario desea hacer sobre el sistema y cómo se realizan estas acciones modificando y validando la información.

La estructura de este patrón propuesta por el framework Symfony queda de la siguiente forma: la Vista cuenta con dos elementos fundamentales, la Capa Externa que es el archivo que contiene el código HTML de las interfaces de usuario que no varía para todas o parte de las páginas Web y las Plantillas que contienen los formularios y datos específicos de cada página. En el Controlador, se encuentra el

Controlador Frontal que constituye el único punto de entrada a la aplicación y cuya principal función es re-direccionar las peticiones realizadas por la Vista, hacia las Acciones que constituyen los controladores de los diferentes módulos que conforman la aplicación. En el Modelo se encuentran la capa de Acceso a Datos y la capa de Abstracción de la Base de Datos. La primera contiene todas las funcionalidades que responden a la lógica del negocio y la segunda provee una abstracción de la base de datos que posibilita que la aplicación no dependa de un sistema gestor determinado.

De esta forma cuando el usuario interactúa con la aplicación lo hace mediante la Vista, que envía las peticiones al Controlador Frontal, que las re-direcciona hacia la Acción correspondiente, encargándose esta de llamar a la función implementada en el Modelo, donde se obtiene la respuesta, que es enviada a través del Controlador hacia la Vista.

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 de crear 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. (33)

### **3.3 Análisis de componentes o módulos rehusados. Estrategias de integración**

#### **3.3.1 Componentes reutilizados**

El SISaludPlugin es un componente desarrollado por el Dpto. Sistemas Especializados de la facultad 7 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.

### **3.3.2 Estrategia de integración**

El sistema tiene la particularidad de consumir información gestionada por los diferentes servicios del RIS. Esta comunicación se realiza mediante el protocolo SOAP para la comunicación con los servicios Web, a continuación se especifican las diferentes funcionalidades de las cuales se obtiene dicha información:

#### **SAAA:**

Autenticar, el usuario obtiene todos los permisos necesarios para acceder a las funcionalidades del sistema y a los restantes Servicios Web

#### **RC**

- Buscar Total Ciudadano, recibe los datos del ciudadano que se desea buscar como parámetro y retorna un listado con las personas que coinciden con la información entrada.

#### **RL**

- Listar Provincias, mediante esta funcionalidad se obtiene toda la información referente a las provincias, municipios, localidades, manzanas y calles en dependencia de la profundidad de la petición.

#### **RPS**

- Buscar Profesionales, 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.
- Listar Nomencladores Especiales, retorna los datos con los codificadores de tipo de profesionales y sus respectivas especialidades.

### **3.4 Análisis**

La etapa de análisis en el proceso de construcción de un sistema no es más que el paso inicial hacia la transformación de los requisitos definidos con el cliente en software ejecutable.

### **3.4.1 Modelo de Análisis**

Se analizan los requisitos que fueron descritos anteriormente, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar todo el sistema, incluyendo su arquitectura.

### **3.4.2 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:

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

### **3.4.3 Diagrama de clase del análisis**

Un Diagrama de clases 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. (34)

CU Autenticar

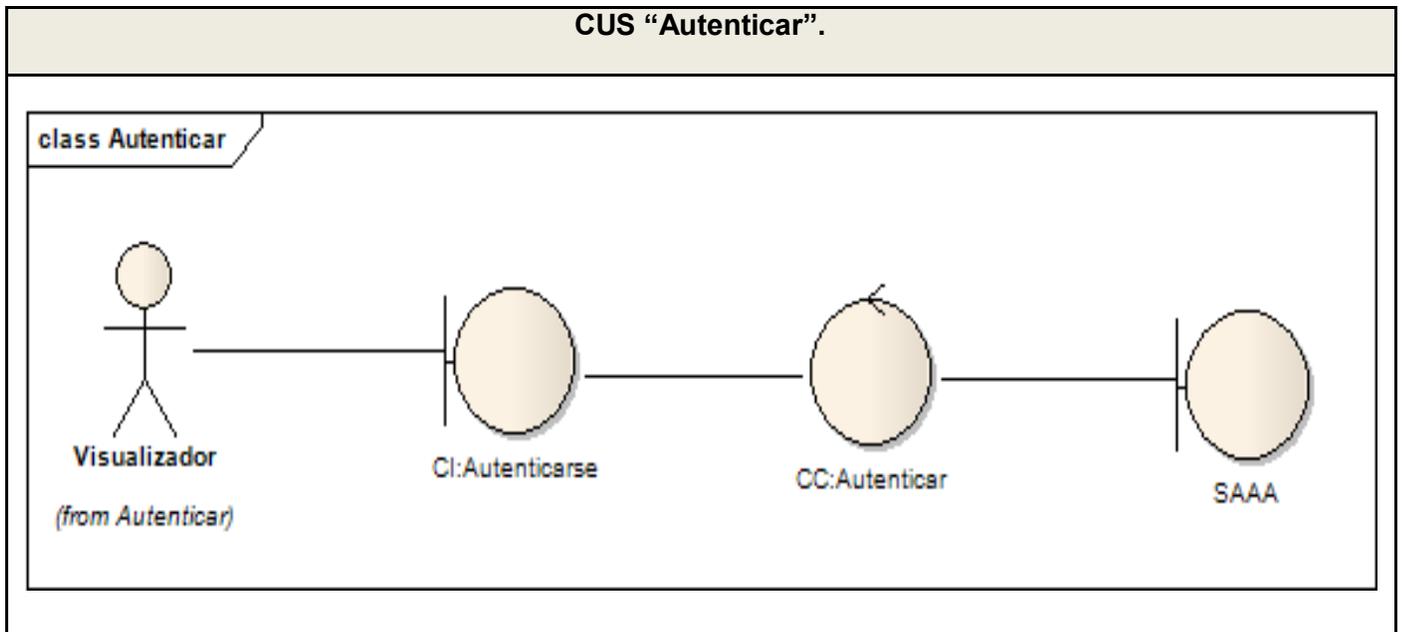


Fig. 4. Diagrama de clases de análisis: CU Autenticar.

CU Buscar Colaborador

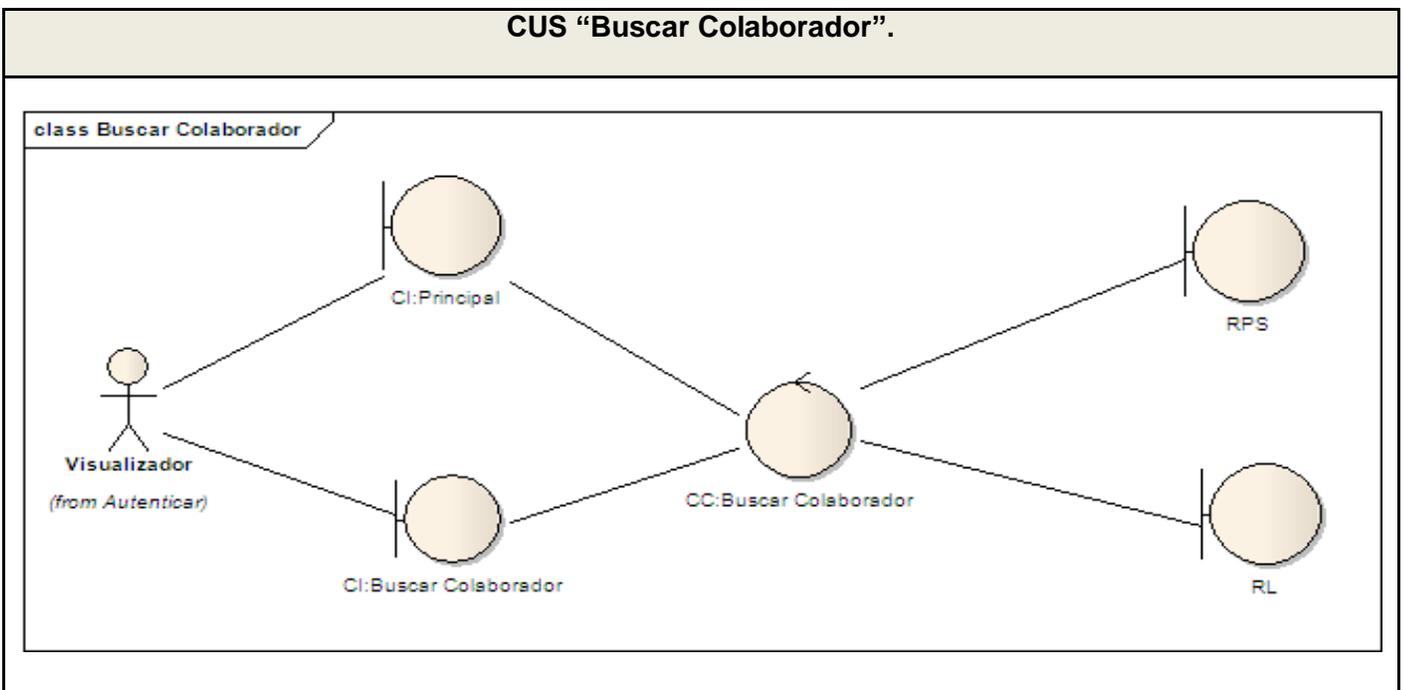


Fig. 5. Diagrama de clases de análisis: CU Buscar Colaborador.

CU Crear Expediente

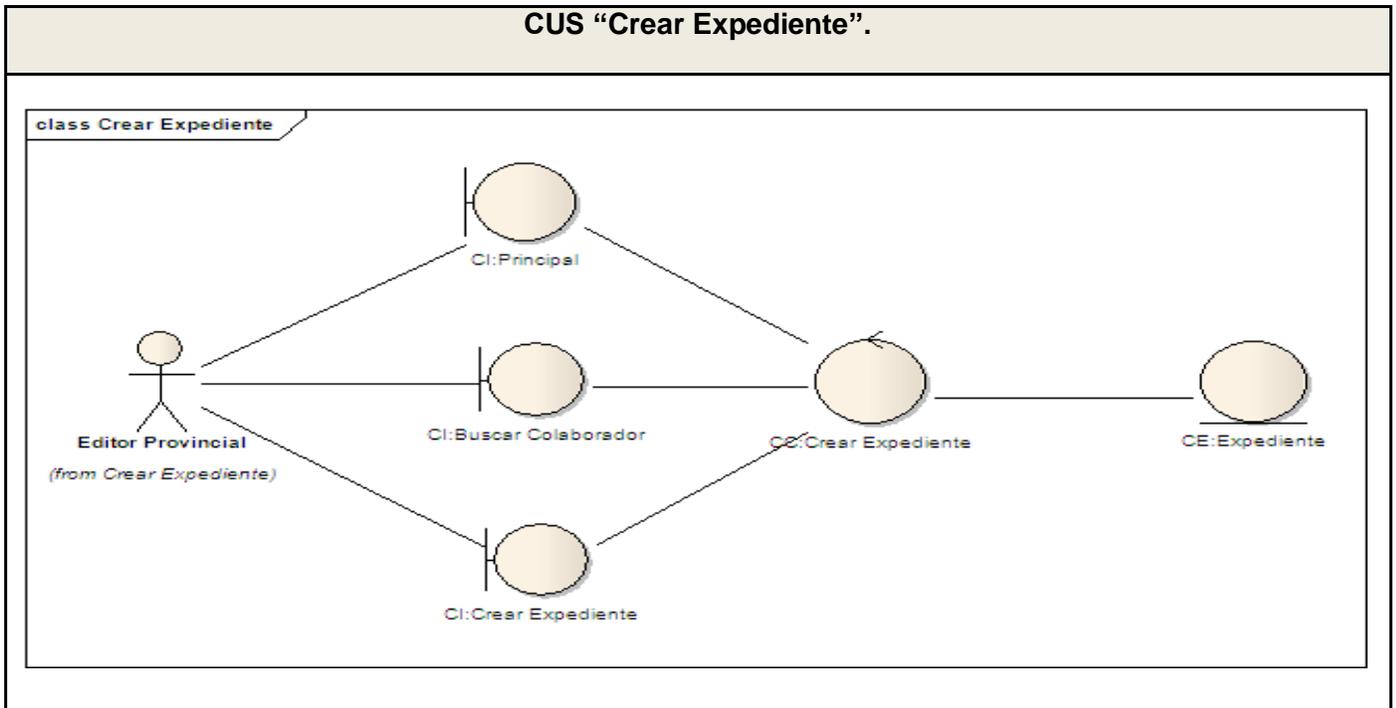


Fig. 6. Diagrama de clases de análisis: CU Crear Expediente.

CU Asignar Misión

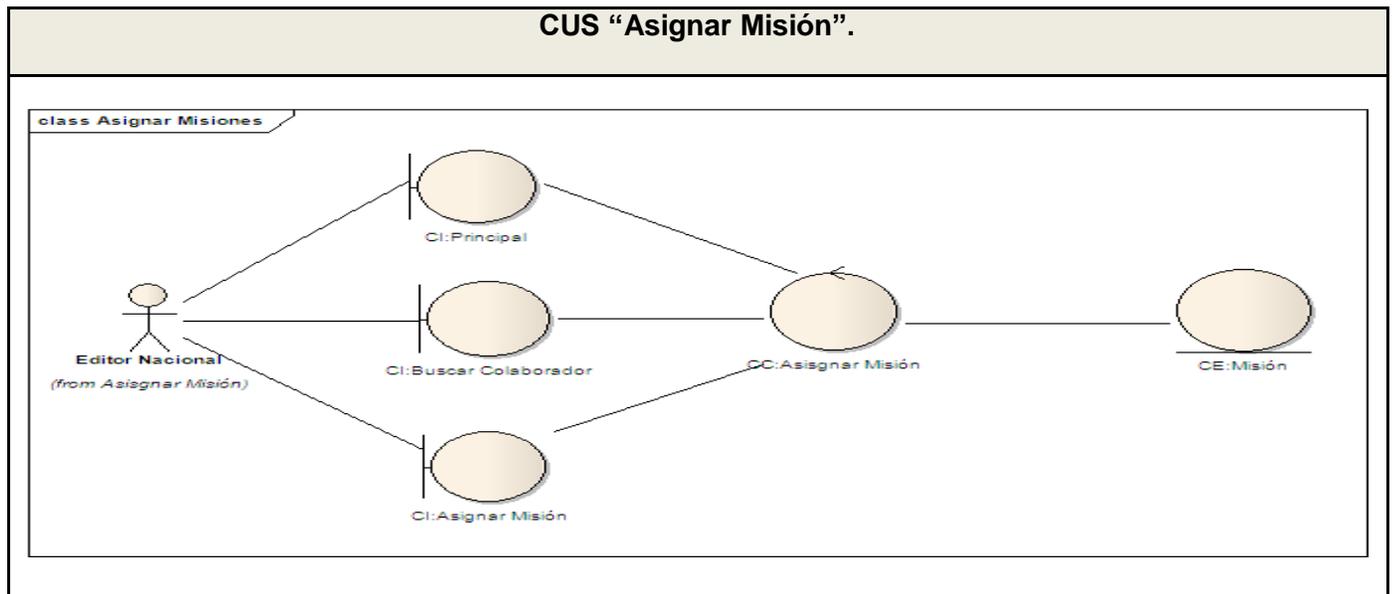


Fig. 7. Diagrama de clases de análisis: CU Asignar Misión.

**CU Insertar Movimiento**

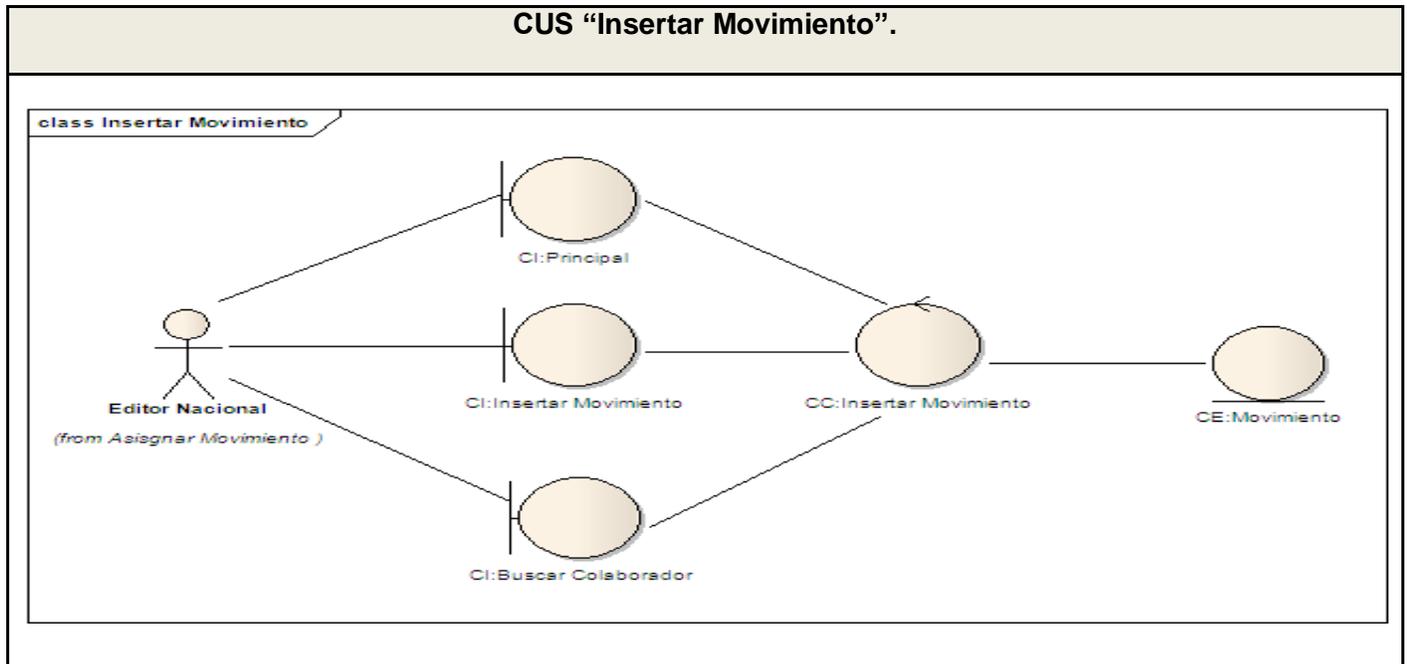


Fig. 8. Diagrama de clases de análisis: CU Insertar Movimiento.

**3.5 Diseño**

El diseño se realiza para crear especificaciones adicionales que enriquecen el modelo de análisis con detalles próximos a la implementación. Es una solución lógica, que describe como los requisitos funcionales y no funcionales tienen impacto en el sistema. (35)

**3.5.1 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. (36)

**3.5.2 Diagrama de Clases de Diseño del Sistema**

Describen gráficamente las especificaciones de las clases del software, los elementos básicos que se pueden encontrar son las clases y sus relaciones. (37)

CU Autenticar

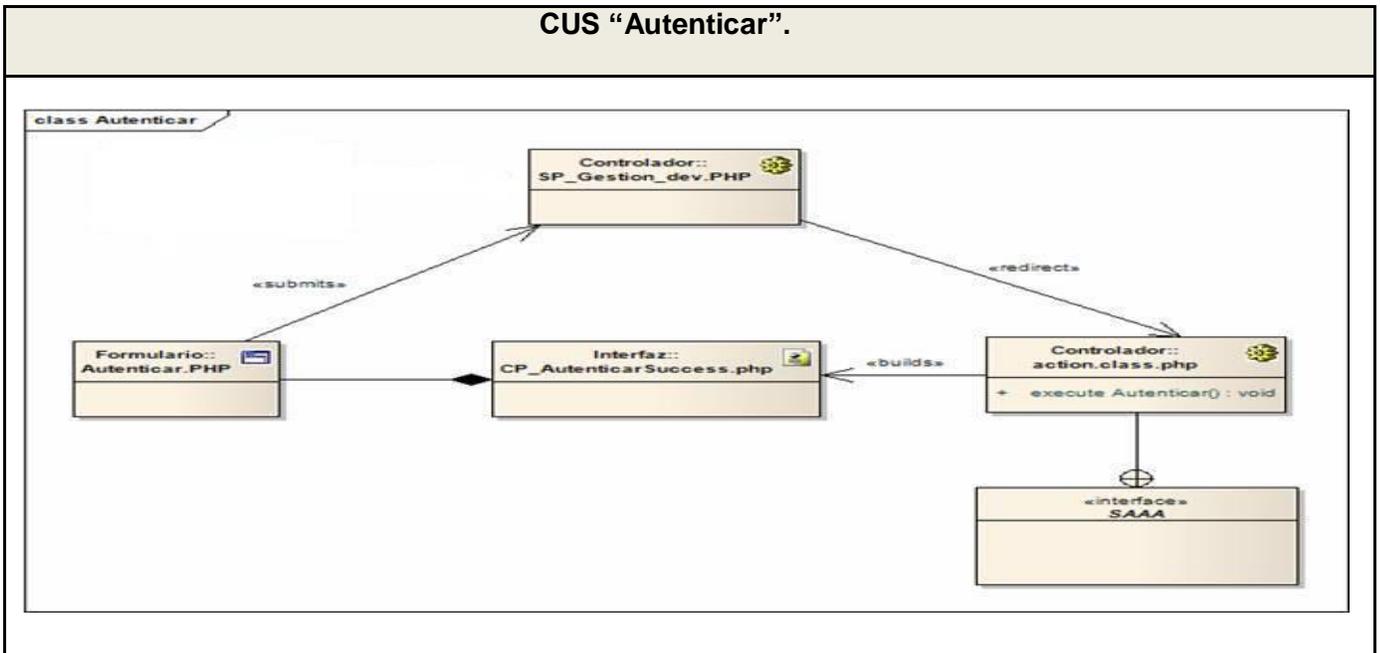


Fig. 9. Diagrama de clases del diseño: CU Autenticar.

CU Buscar Colaborador

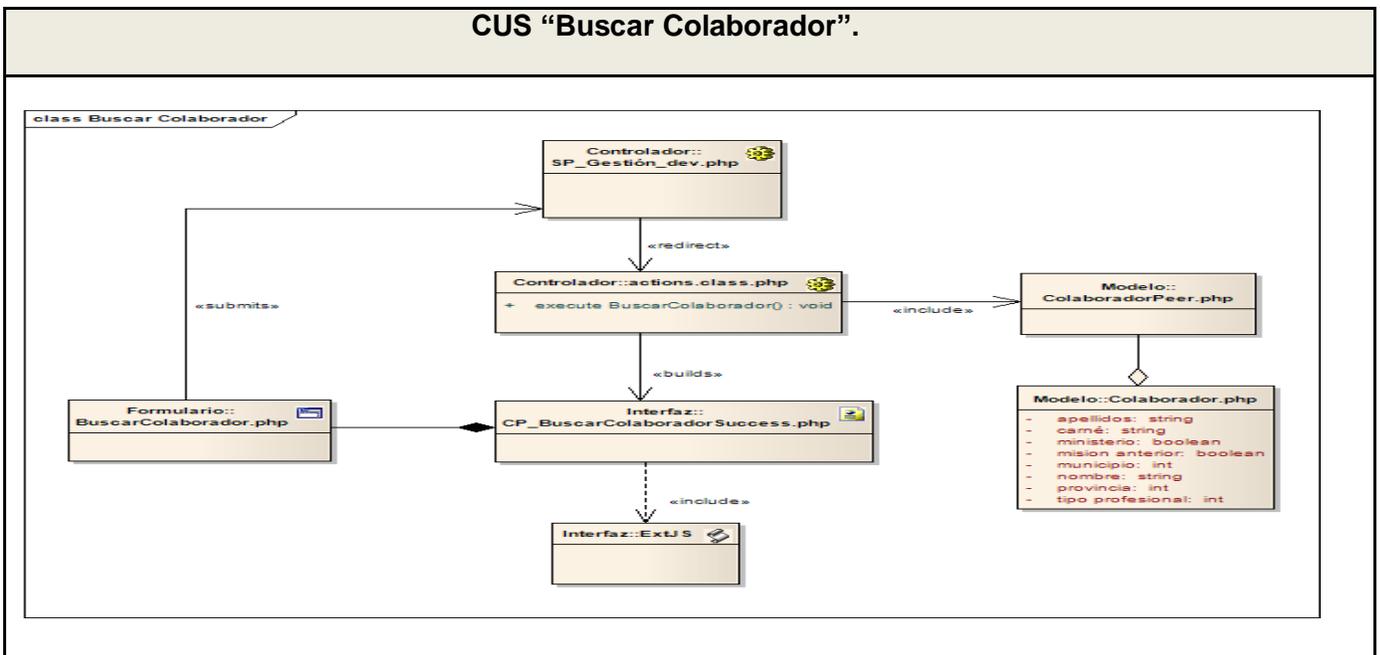


Fig. 10. Diagrama de clases del diseño: CU Buscar Colaborador.

CU Crear Expediente

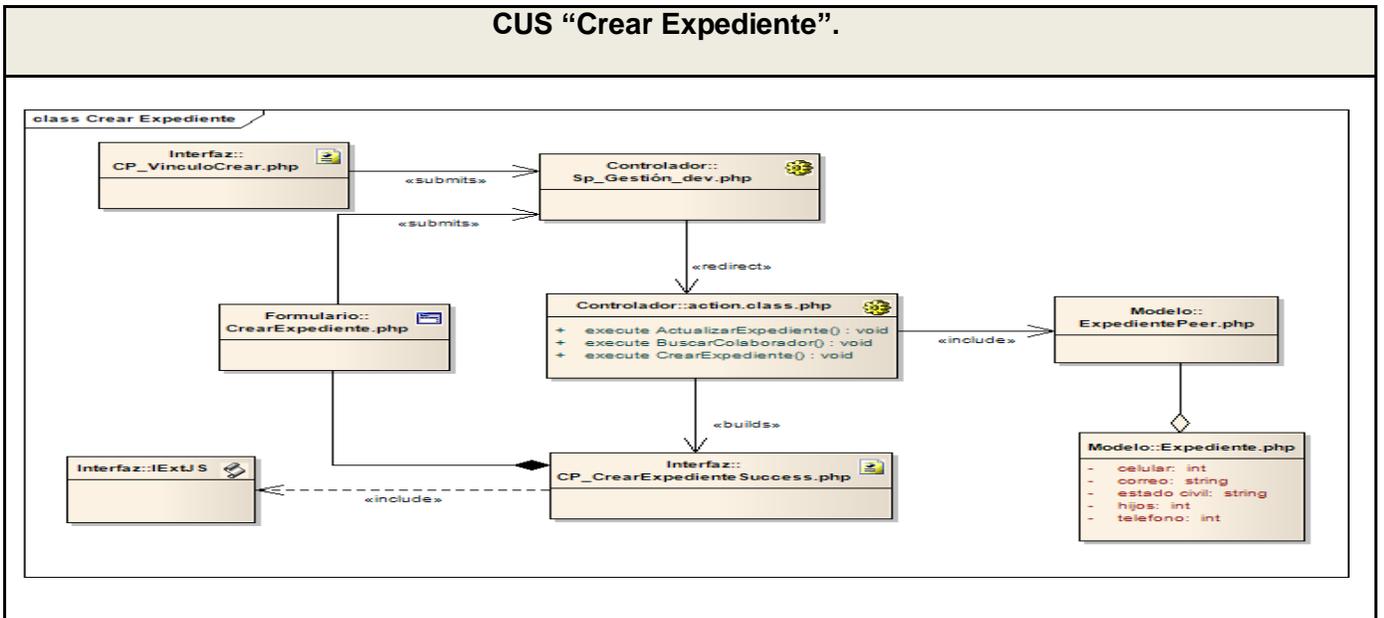


Fig. 11. Diagrama de clases del diseño: CU Crear Expediente.

CU Asignar Misión

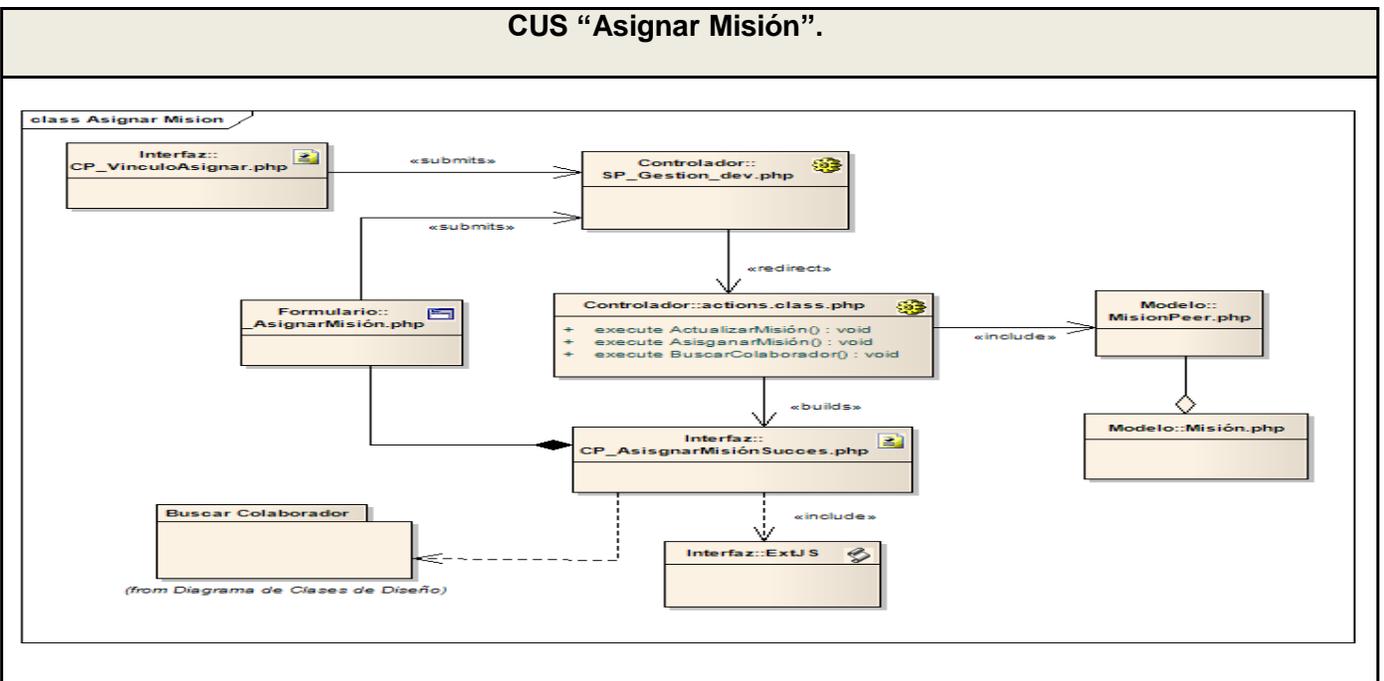


Fig. 12. Diagrama de clases del diseño: CU Asignar Misión.

CU Insertar Movimiento

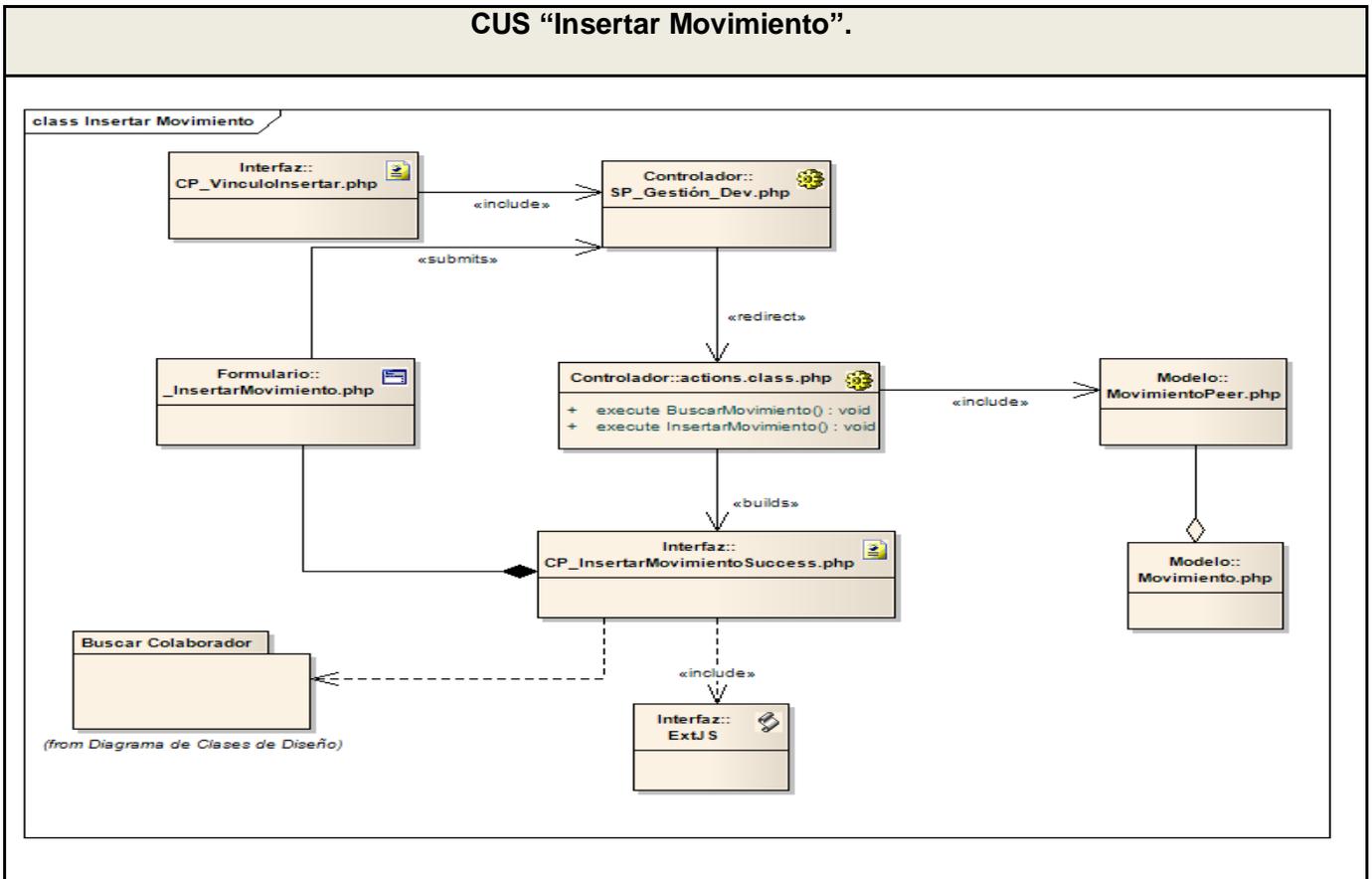


Fig. 13. Diagrama de clases del diseño: CU Insertar Movimiento.

3.5.3 Diagrama de Secuencia

Diagramas de interacción, es un término genérico que se aplica a varios tipos de diagramas que describen las interacciones entre objetos, de mayor utilización los diagramas de colaboración y secuencia. Ambos son isomorfos, es decir, se pueden transformar de uno al otro sin pérdida de información. Razón por la cual solo se hace referencia en el presente trabajo a uno de ellos en este caso el de secuencia.

De forma específica los diagramas de secuencia están formados por una serie de eventos de forma ordenada y ascendente que describen los pasos a realizar para la implementación de cada CU.

CU Autenticar

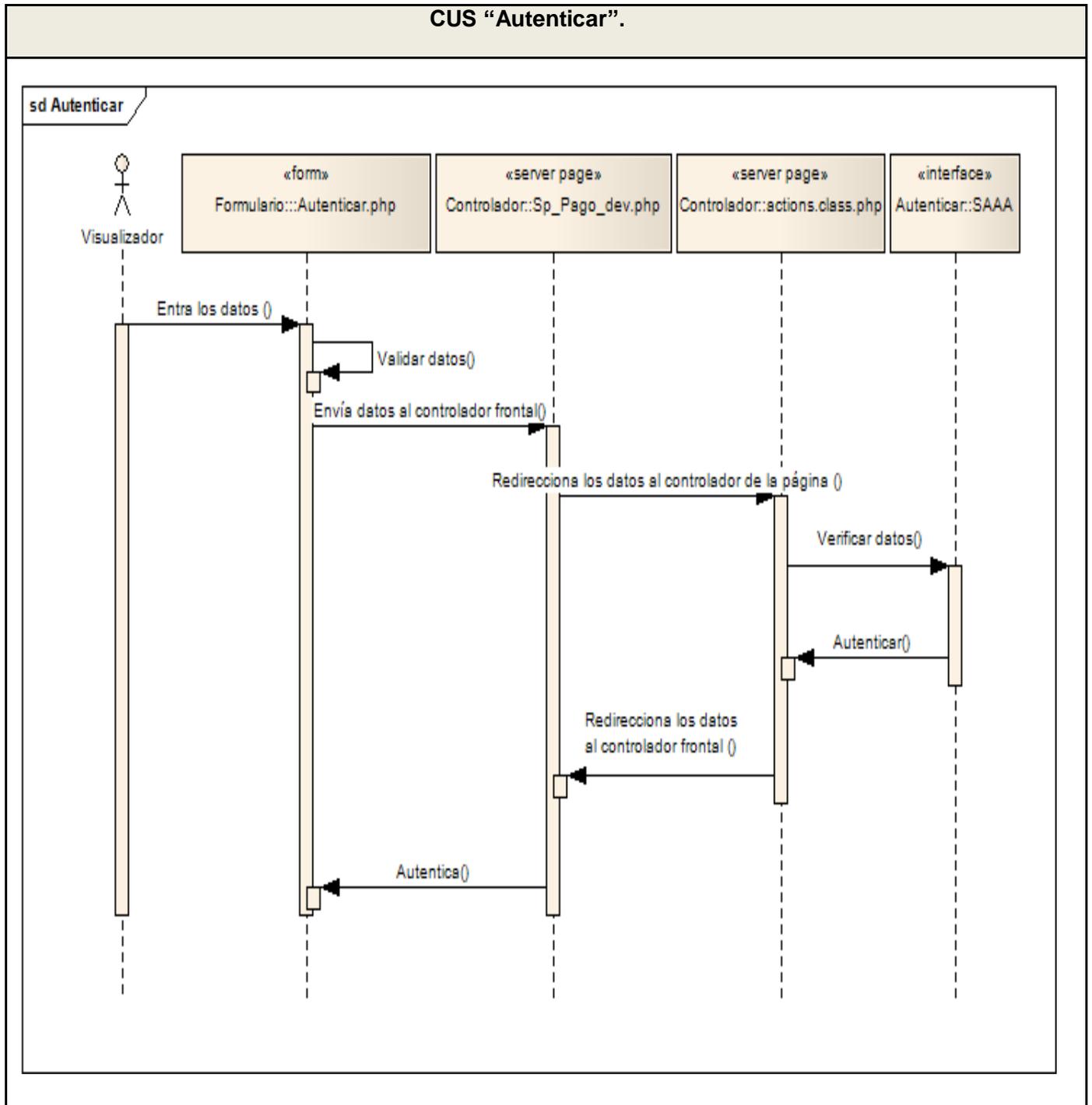


Fig. 14. Diagrama de Secuencia: CU Autenticar.

CU Buscar Colaborador

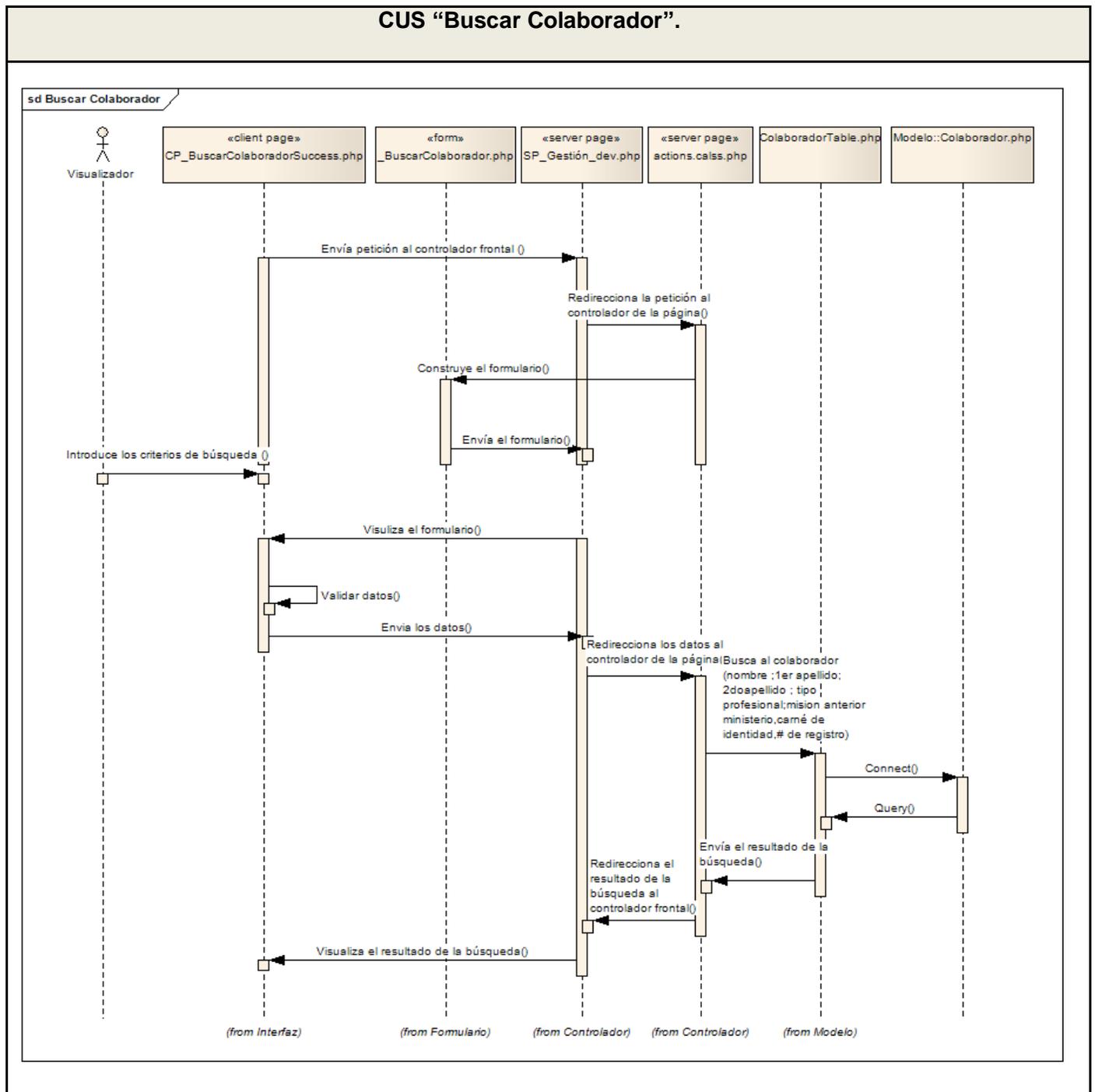


Fig. 15. Diagrama de Secuencia: CU Buscar Colaborador.

CU Crear Expediente

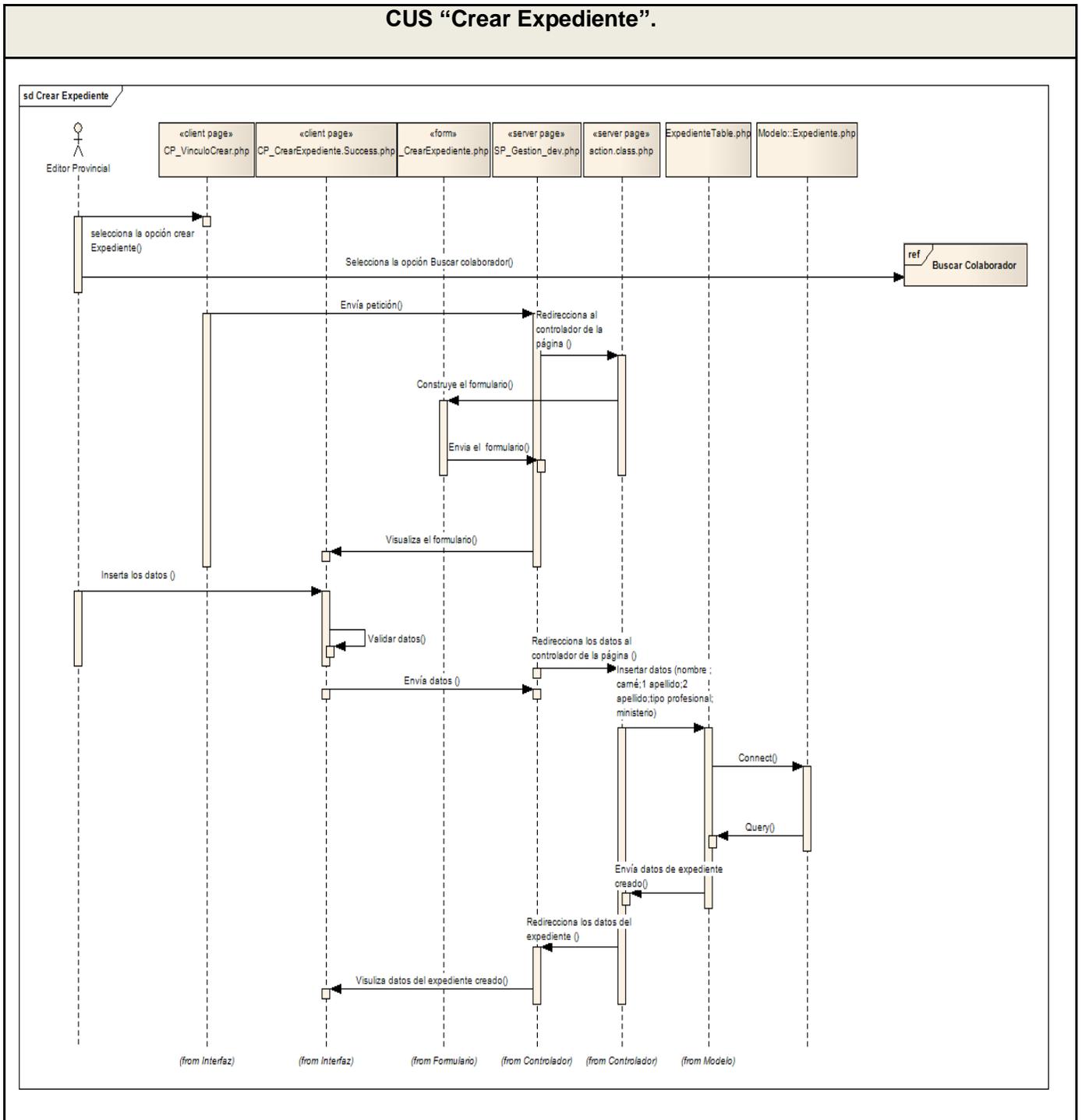


Fig. 16. Diagrama de Secuencia: CU Crear Expediente.

CU Asignar Misión

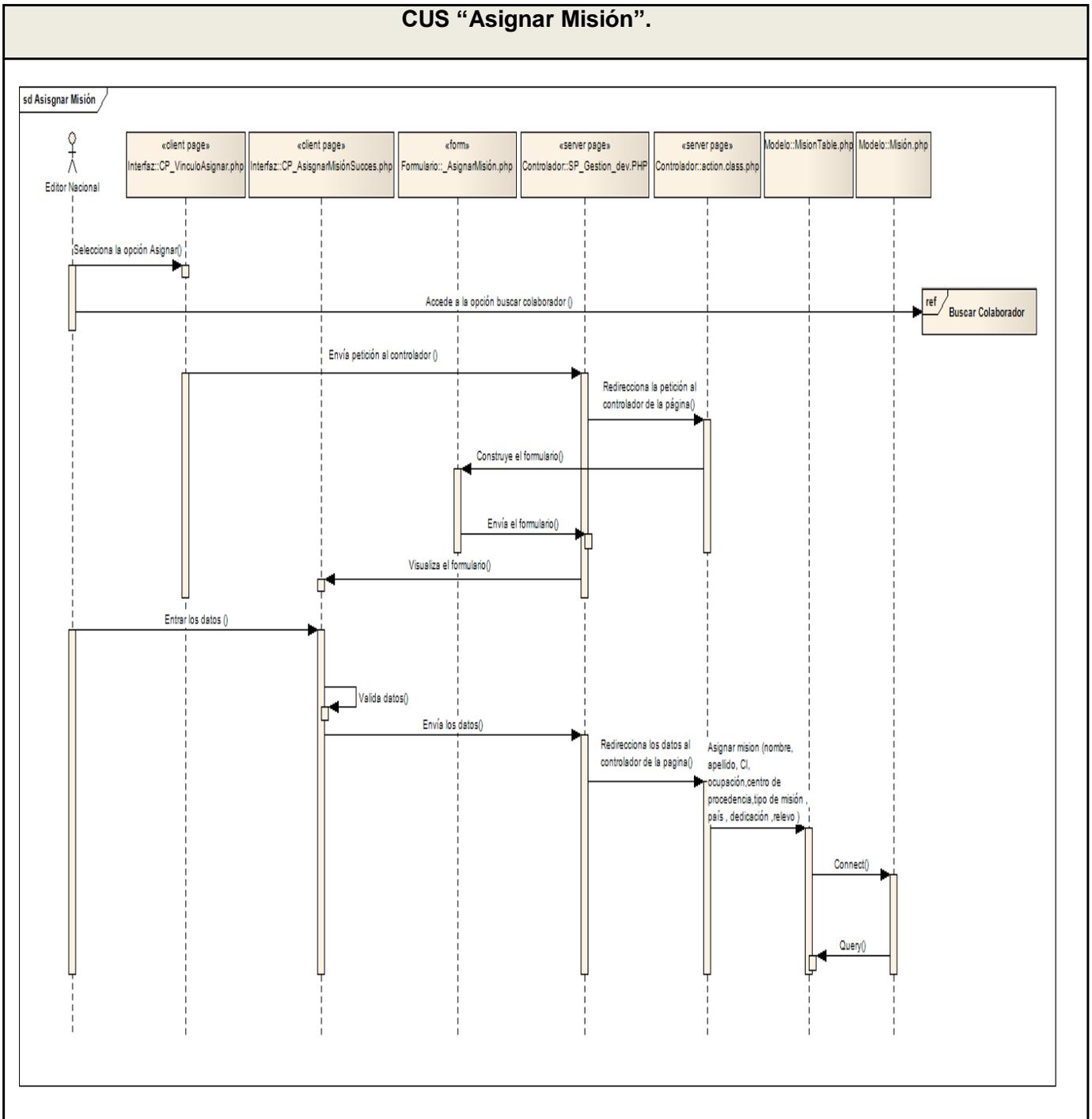


Fig. 17. Diagrama de Secuencia: Asignar Misión.

CU Insertar Movimiento

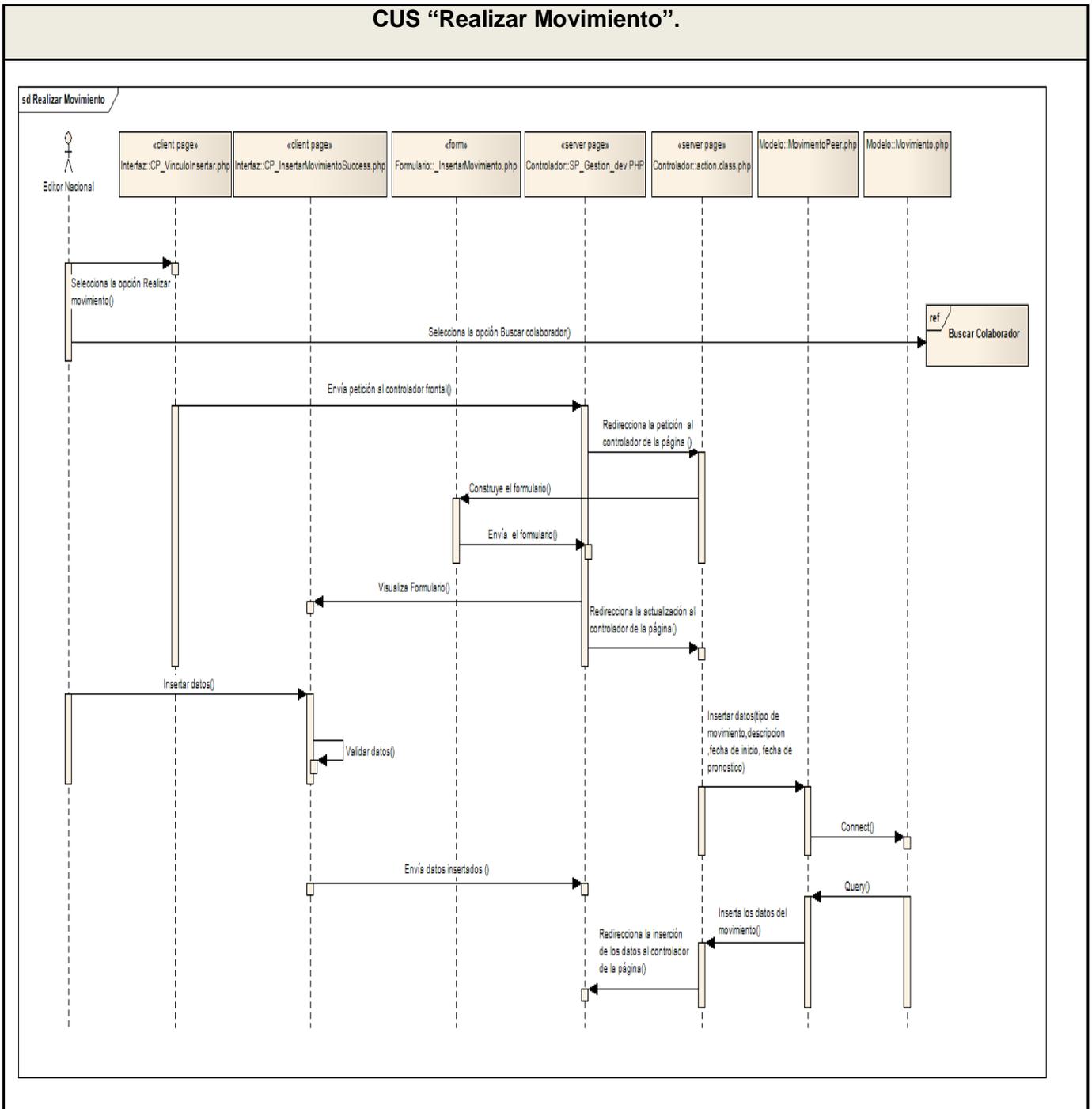


Fig. 18. Diagrama de Secuencia: CU Insertar Movimiento.

**3.5.4 Descripción de las clases del diseño.**

**CU Autenticar Usuario**

|   |             |
|---|-------------|
| <b>Nombre: CP_AutenticarSuccess.php</b> |             |
| <b>Tipo de clase :</b> Interfaz         |             |
| <b>Atributo</b>                         | <b>Tipo</b> |
| Usuario                                 | TextField   |
| Contraseña                              | TextField   |
| <b>Para cada responsabilidad:</b>       |             |
| Nombre:                                 |             |
| Descripción:                            |             |

Tabla #12. Descripción de clases del diseño CU: Autenticar Usuario.

|                                    |  |
|------------------------------------|--|
| <b>Nombre: actions.class.php</b>   |  |
| <b>Tipo de clase:</b> controladora |  |
| <b>Atributo</b>                    | <b>Tipo</b>  |
|                                    |  |
| <b>Para cada responsabilidad:</b>  |  |
| Nombre:                            | executeAutenticar(usuario, contraseña)               |
| Descripción:                       | Permite el acceso a las funcionalidades del sistema. |

Tabla #13. Descripción de clases del diseño CU: Autenticar Usuario.

**CU Buscar colaborador.**

|  |             |
|--|-------------|
| <b>Nombre: CP_BuscarColaboradorSuccess.php</b> |             |
| <b>Tipo de clase :</b> Interfaz                |             |
| <b>Atributo</b>                                | <b>Tipo</b> |
| CI   | TextField   |
| Nombre   | TextField   |
| 1er Apellido                                   | TextField   |
| 2do Apellido                                   | TextField   |
| Tipo de profesional                            | ComboBox    |

|                                   |              |
|-----------------------------------|--------------|
| Buscar                            | Button       |
| Misión Anterior                   | Radio Button |
| Provincia                         | ComboBox     |
| Municipio                         | ComboBox     |
| Ministerio                        | ComboBox     |
| <b>Para cada responsabilidad:</b> |              |
| Nombre:                           |              |
| Descripción:                      |              |

Tabla #14. Descripción de clases del diseño CU: Buscar Colaborador.

|                                    |   |
|------------------------------------|---|
| <b>Nombre: actions.class.php</b>   |   |
| <b>Tipo de clase: controladora</b> |   |
| <b>Atributo</b>                    | <b>Tipo</b>   |
|                                    |   |
| <b>Para cada responsabilidad:</b>  |   |
| 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.   |

Tabla #15. Descripción de clases del diseño CU: Buscar Colaborador.

**Crear expediente**

|  |             |
|--|-------------|
| <b>Nombre: CP_CrearExpedienteSuccess.php</b> |             |
| <b>Tipo de clase : Interfaz</b>              |             |
| <b>Atributo</b>                              | <b>Tipo</b> |
| CI   | TextField   |
| Nombre                                       | TextField   |
| 1er Apellido                                 | TextField   |
| 2do Apellido                                 | TextField   |
| talla  | TextField   |
| Sexo   | ComboBox    |
| Teléfono                                     | TextField   |

|                                   |           |
|-----------------------------------|-----------|
| Correo                            | TextField |
| Estado civil                      | ComboBox  |
| Organizaciones políticas          | ComboBox  |
| Tipo profesional                  | ComboBox  |
| Nivel técnico                     | ComboBox  |
| Fecha de entrada                  | ComboBox  |
| Entregado por                     | TextField |
| <b>Para cada responsabilidad:</b> |           |
| Nombre:                           |           |
| Descripción:                      |           |

Tabla #16. Descripción de clases del diseño CU: Crear Expediente.

|                                    |   |
|------------------------------------|---|
| <b>Nombre: actions.class.php</b>   |   |
| <b>Tipo de clase: controladora</b> |   |
| <b>Atributo</b>                    | <b>Tipo</b>   |
|                                    |   |
| <b>Para cada responsabilidad:</b>  |   |
| Nombre:                            | executeCrearExpediente()  |
| Descripción:                       | Permite crear el expediente de un colaborador determinado para recoger todos los datos del mismo. |

Tabla #17. Descripción de clases del diseño CU: Crear Expediente.

**Asignar Misión**

|  |             |
|--|-------------|
| <b>Nombre: CP_AsignarMisiónSuccess.php</b> |             |
| <b>Tipo de clase : Interfaz</b>            |             |
| <b>Atributo</b>                            | <b>Tipo</b> |
| CI   | TextField   |
| Nombre                                     | TextField   |
| 1er Apellido                               | TextField   |
| 2do Apellido                               | TextField   |
| N. de registro                             | TextField   |

|                                   |          |
|-----------------------------------|----------|
| Centro de procedencia             | ComboBox |
| Ocupación                         | ComboBox |
| País                              | ComboBox |
| Dedicación                        | ComboBox |
| Relevo                            | ComboBox |
| Tipo de misión                    | ComboBox |
| Tiempo de estancia                | ComboBox |
| Fecha de salida                   | ComboBox |
| Especialidad                      | ComboBox |
| <b>Para cada responsabilidad:</b> |          |
| Nombre:                           |          |
| Descripción:                      |          |

Tabla #18. Descripción de clases del diseño CU: Asignar misión.

|                                    |   |
|------------------------------------|---|
| <b>Nombre: actions.class.php</b>   |   |
| <b>Tipo de clase: controladora</b> |   |
| <b>Atributo</b>                    | <b>Tipo</b>   |
|                                    |   |
| <b>Para cada responsabilidad:</b>  |   |
| Nombre:                            | executeAsignarMisión()                                  |
| Descripción:                       | Permite asignar una misión a un colaborador determinado |

Tabla #19. Descripción de clases del diseño CU: Asignar misión.

**Realizar Movimiento**

|   |             |
|---|-------------|
| <b>Nombre: CP_RealizarMovimientoSuccess.php</b> |             |
| <b>Tipo de clase : Interfaz</b>                 |             |
| <b>Atributo</b>                                 | <b>Tipo</b> |
| CI  | TextField   |
| Nombre  | TextField   |
| 1er Apellido                                    | TextField   |
| 2do Apellido                                    | TextField   |

|                                   |           |
|-----------------------------------|-----------|
| N. de registro                    | TextField |
| Relevo                            | ComboBox  |
| Tipo de misión                    | ComboBox  |
| Tiempo de estancia                | ComboBox  |
| Fecha de salida                   | ComboBox  |
| Especialidad                      | ComboBox  |
| Tipo de movimiento                | ComboBox  |
| <b>Para cada responsabilidad:</b> |           |
| Nombre:                           |           |
| Descripción:                      |           |

Tabla #20. Descripción de clases del diseño CU: Realizar Movimiento.

|                                    |   |
|------------------------------------|---|
| <b>Nombre: actions.class.php</b>   |   |
| <b>Tipo de clase: controladora</b> |   |
| <b>Atributo</b>                    | <b>Tipo</b>   |
|                                    |   |
| <b>Para cada responsabilidad:</b>  |   |
| Nombre:                            | executeAsignarMisión()                                  |
| Descripción:                       | Permite asignar una misión a un colaborador determinado |

Tabla #21. Descripción de clases del diseño CU: Realizar Movimiento.

### **Conclusiones**

En el presenta capítulo, después de realizar los diagramas de análisis, diseño y secuencia, se obtuvo la estructura del funcionamiento del sistema internamente. Con la generación de estos artefactos el equipo de desarrollo de la aplicación posee una idea común y una línea base a seguir para llevar a cabo la implementación exitosa del sistema.

## Capítulo 4: Implementación

El trabajo en este capítulo parte del resultado del análisis y diseño obtenidos en el capítulo anterior y se procederá a implementar el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. Además, se describirá cómo se organiza el sistema de acuerdo con los nodos específicos en el modelo de despliegue.

### 4.2 Modelo de datos

A continuación se presenta el modelo de datos, su objetivo es dar a conocer una mejor comprensión del funcionamiento del sistema, permite describir los elementos de la realidad que intervienen en el problema dado y la forma en que se relacionan esos elementos entre sí. (38)

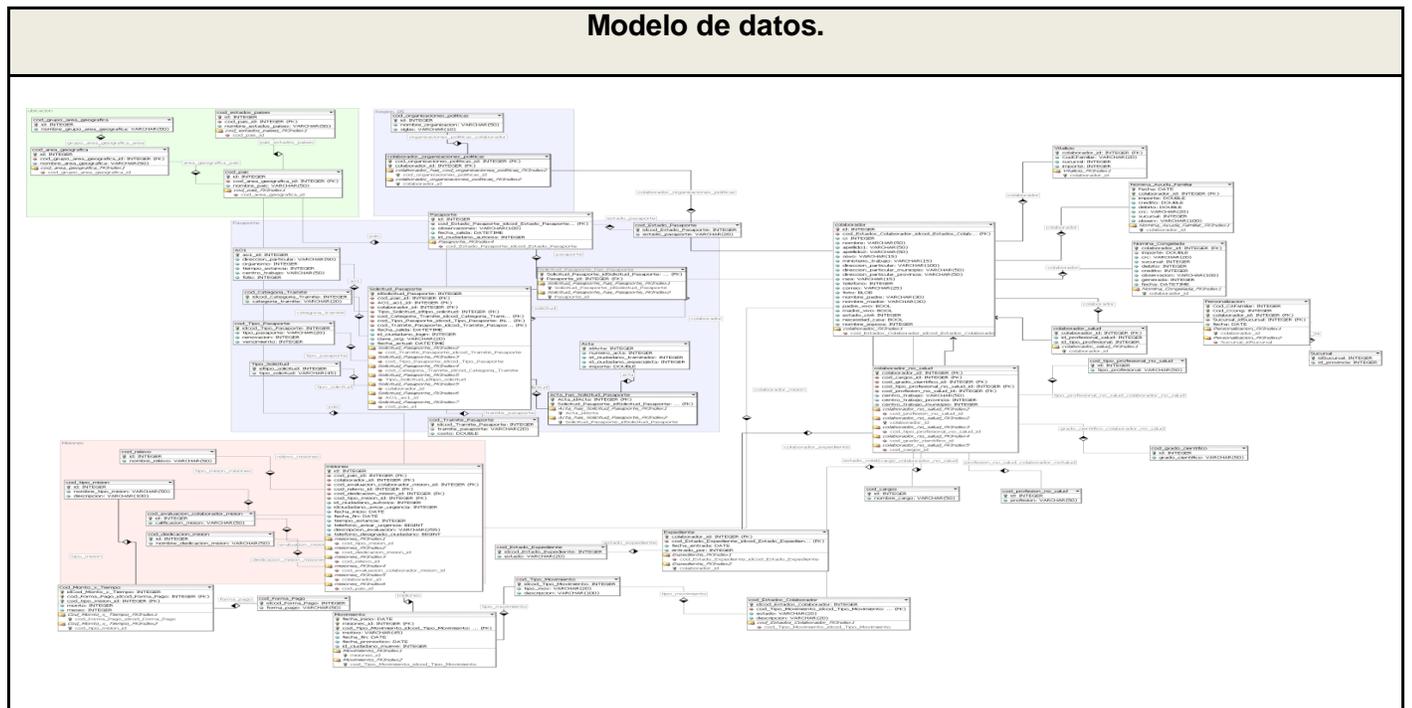


Fig. 19. Modelo de datos

### 4.3 Diagrama de despliegue

Un diagrama de despliegue muestra la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. A continuación se muestra una propuesta de despliegue que es como se distribuirá el sistema.

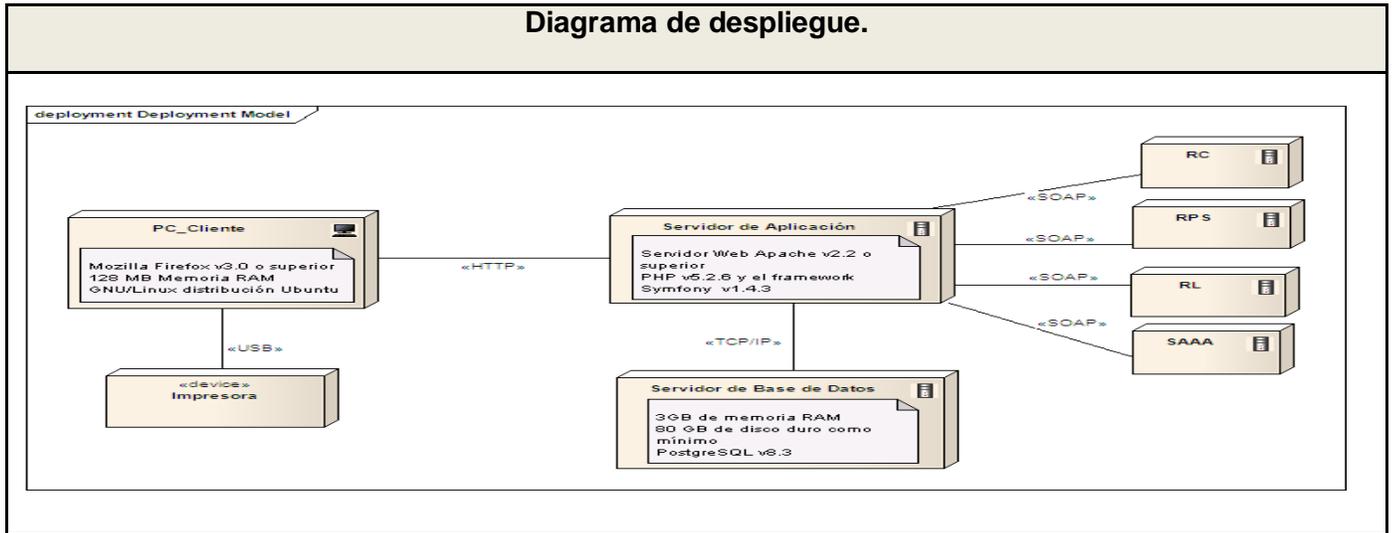


Fig. 20. Diagrama de despliegue del sistema.

#### 4.4 Diagrama de componentes

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. Modelan la vista estática de un sistema. Muestran la organización y las dependencias entre un conjunto de componentes. Cada diagrama describe un apartado del sistema.

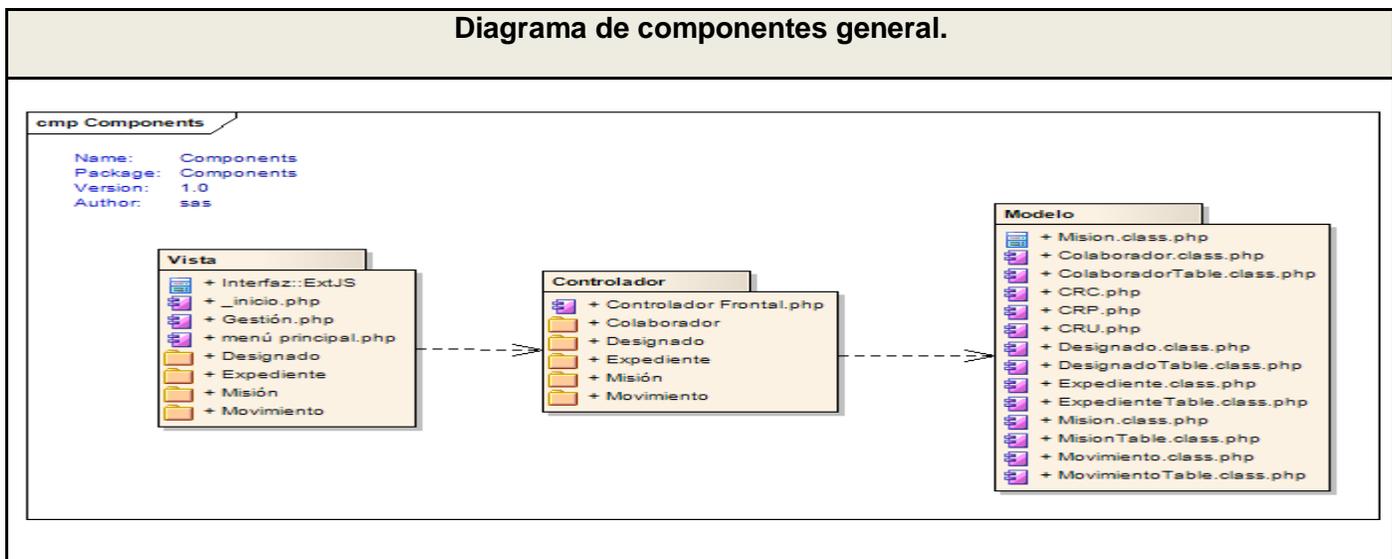


Fig.21. Diagrama de componentes.

### **4.5 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 minimizar estos errores uno de los métodos utilizados fue el trabajo con el lenguaje de programación JavaScript, 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, es decir, que no llene todos los campos, que introduzca los datos erróneamente.

### **4.6 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, garantizan así la protección de la misma.

Para lograr la seguridad total del sistema propuesto se realiza la autenticación del mismo mediante un sistema de administración que nos brinda en proyecto denominado SAAA, de esta manera, el usuario obtendrá los permisos necesarios para acceder al sistema. También el sistema gestor de base datos debe mantenerse en un lugar restringido y asegurado, además de realizarle salvallas continuas para evitar las pérdidas de datos que hayan sido insertados en el sistema.

### **4.7 Estrategias de codificación. Estándares y estilos utilizados**

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.

**4.8 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 cual 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, Insertar Expediente y Buscar Aspirante se muestran a continuación:

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

| Label | # Muestras | Media | Mediana | Linea de 9... | Mín | Máx  | % Error | Rendimiento | Kb/sec |
|-------|------------|-------|---------|---------------|-----|------|---------|-------------|--------|
| TOTAL | 3400       | 59    | 15      | 56            | 1   | 6756 | 0,00%   | 147,4/sec   | 583,67 |

Fig.21. Resultados de las pruebas.

**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 | 1300       | 739   | 789     | 948          | 8   | 1406 | 0,00%   | 13,3/sec    | 329,59 |

Fig.21. Resultados de las pruebas.

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 Gestión de la información de los Colaboradores del Sistema para la Colaboración Médica en la versión 2.0.

**Conclusiones**

En el presente 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, debido a que brinda una idea más clara de cómo desarrollar una aplicación Web confiable. Comprobándose que el tiempo de respuesta disminuyó considerablemente en comparación con la aplicación desarrollada anteriormente.

### **Conclusiones Generales**

La realización del presente trabajo ha cumplido con los objetivos propuestos, contribuyendo a mejorar el proceso de gestión de los colaboradores de la salud, por lo que se puede concluir que:

- Con la implementación del Módulo de Gestión de Información de los Colaboradores del Sistema para la Colaboración Médica v2.0, se obtuvo una arquitectura más robusta y se mejora la trazabilidad y auditoría en la gestión de información de los colaboradores de la salud.
- El módulo desarrollado proporciona un menor tiempo de respuesta al usuario, lo que posibilita un buen rendimiento.

## **Beneficios**

- Garantizará la centralización y seguridad de los usuarios que se autenticarán en el sistema.
- Proporcionará el control de la información de los colaboradores de la salud con una mayor rapidez.
- Proporcionará la trazabilidad y la auditoría, lo que garantizará una mayor seguridad de la información que se maneja.
- Permitirá la obtención de reportes mediante la recuperación de datos históricos almacenados en el sistema.

### **Recomendaciones**

Por las experiencias alcanzadas durante la realización de este trabajo y con vista a enriquecer la solución propuesta, se recomienda:

- Integrar la versión del módulo desarrollado con el reporteador dinámico de la UCI para aumentar las funcionalidades del mismo.
- Brindar servicios Web de la información almacenada de los colaboradores de la salud a todas las entidades involucradas que tengan acceso a la información que se brindará.

### Referencias Bibliográficas

1. Curación global, empoderamiento local: la cooperación médica. [En línea] [Citado el: 20 de 01 de 10.] [http://www.connergorry.com/article\\_pdfs/Pagesfromcaminos\\_42.pdf](http://www.connergorry.com/article_pdfs/Pagesfromcaminos_42.pdf).
2. Igual a (1)
3. Igual a (1)
4. SLD095-El Registro Informatizado de Salud (RIS). Solución ... [En línea] [Citado el: 4 de 3 de 10.] <http://www.informatica2007.sld.cu/Members/arieldr/el-registro-informatizado-de-salud-ris-solucion-informatica-integral-para-el-sistema-nacional-de-salud/>.
5. Empleados MSD Multiusuario. [En línea] [Citado el: 20 de 01 de 10.] <http://www.msdssoft.com/esp/products/empleados/info.htm>.
6. Wikimetal RRHH v3.0. [En línea] [Citado el: 15 de 02 de 10.] <http://www.wikimetal.net/>.
7. @GesRRHH v2.0. [En línea] [Citado el: 15 de 02 de 10.] <http://www.e-rem.net/>.
8. Infomed, Portal de Salud de Cuba. [En línea] [Citado el: 25 de 2 de 10.] <http://www.sld.cu/>.
9. Sistemas Fastos. [En línea] [Citado el: 15 de 02 de 10.] <http://www.desoft.cu/.../SistemasFastos/.../Default.aspx?>.
10. Igual a (1)
11. Penichet Ulibarri, Juan Manuel. Universidad de las Américas Puebla2004. [En línea] [Citado el: 20 de 02 de 10.] [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/ulibarri\\_p\\_jm/capitulo4.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/ulibarri_p_jm/capitulo4.pdf).
12. Comex grupo ibérica. [En línea] [Citado el: 11 de 03 de 10.] <http://www.grupocomex.com/reqm-gestion-de-requisitos.aspx..>
13. Entorno Virtual de Aprendizaje. [En línea] [Citado el: 10 de 01 de 2010.] [http://eva.uci.cu/mod/resource/view.php?id=2241&subdir=/Conferencias\\_IS1\\_2007-2008..](http://eva.uci.cu/mod/resource/view.php?id=2241&subdir=/Conferencias_IS1_2007-2008..)
14. Qué es RUP. [En línea] [Citado el: 20 de 01 de 2010.] <http://www.scribd.com/doc/12812403/Que-es-RUP>.
15. UML. [En línea] [Citado el: 10 de 05 de 10.] <http://usuarios.multimania.es/oopere/uml.htm>.
16. Enterprise Architect- Herramienta de diseño UML. [En línea] [Citado el: 01 de 01 de 10.] <http://www.sparxsystems.com.ar/products/ea.html..>
17. Bienvenido a NetBeans y [www.netbeans.org](http://www.netbeans.org), Portal del IDE Java de... [En línea] [Citado el: 21 de 01 de 10.] [En línea] [Citado el: 21 de 01 de 2010.] [http://netbeans.org/index\\_es.html](http://netbeans.org/index_es.html).

18. Qué es PHP. [En línea] [Citado el: 22 de 01 de 10.] <http://www.desarrolloweb.com/articulos/392.php>.
19. ¿Qué es Javascript? [En línea] [Citado el: 22 de 01 de 10.] <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.
20. Garcia, Víctor Martín. - » Entendiendo AJAX. [En línea] [Citado el: 22 de 01 de 10.] <http://www.victormartingarcia.com/2007/04/02/entendiendo-ajax/>.
21. Patrón "Modelo-Vista-Controlador". [En línea] [Citado el: 22 de 01 de 10.] <http://www.proactiva-calidad.com/java/patrones/mvc.html>.
22. Definición arquitectura cliente servidor. [En línea] [Citado el: 22 de 01 de 10.] <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml#element>.
23. Linux Para Todos Linux Para Todos - Un buen sitio donde empezar. [En línea] [Citado el: 22 de 01 de 10.] <http://www.linuxparatodos.net/portal/>.
24. Manual del usuario de PostgreSQL Sistemas Gestores de Bases de Datos. [En línea] [Citado el: 22 de 01 de 10.] <http://www.scribd.com/doc/5703210/Manual-del-usuario-de-PostgreSQL>.
25. ¿Qué es un servidor? - Definición y explicación del término servidor. [En línea] [Citado el: 22 de 01 de 10.] <http://www.masadelante.com/faqs/servidor>.
26. Mozilla Firefox, el navegador web del momento. [En línea] [Citado el: 22 de 01 de 10.] <http://www.maestrosdelweb.com/editorial/firefox/>.
27. Definición de Servicio web - ¿qué es Servicio web? [En línea] [Citado el: 10 de 4 de 10.] <http://www.alegsa.com.ar/Dic/servicio%20web.php>.
28. Manual de CodeIgniter en español. [En línea] [Citado el: 22 de 01 de 10.] <http://techtastico.com/post/manual-codeigniter-castellano/>.
29. Tutoriales muy útiles de CakePHP. [En línea] [Citado el: 5 de 12 de 10.] <http://techtastico.com/post/tutoriales-muy-utiles-de-cakephp/>.
30. Symfony, la guía definitiva | LibrosWeb.es. [En línea] [Citado el: 22 de 01 de 10.] <http://www.librosweb.es/symfony>.
31. ExtJS lo bueno, lo malo y lo feo | Desarrollo en Web. [En línea] [Citado el: 22 de 01 de 10.] <http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>.
32. El Mundo Informático. [En línea] [Citado el: 2 de 5 de 10.] <http://jorgesaavedra.wordpress.com/category/patrones-grasp/>.

33. EVA UCI. [En línea] 10 de 02 de 2010. <http://eva.uci.cu/mod/resource/view.php?id=22668..>
34. EVA UCI. [En línea] [Citado el: 03 de 03 de 10.] <http://eva.uci.cu/mod/resource/view.php?id=14069..>
35. Igual a (32).
36. Igual a (32).
37. Igual a (32).
38. Pozo Alfonso, Yoelvis. Diseño e Implementación de la Base de Datos del Sistema de Colaboración Médica v2.0. Cuba. 2010.

## Bibliografía

1. Bienvenido a NetBeans y [www.netbeans.org](http://www.netbeans.org), Portal del IDE Java de... [En línea] [Citado el: 21 de 01 de 2010.] [http://netbeans.org/index\\_es.html](http://netbeans.org/index_es.html).
2. Comex grupo ibérica. [En línea] 2007. [Citado el: 11 de marzo de 2010.] <http://www.grupocomex.com/reqm-gestion-de-requisitos.aspx>.
3. Curación global, empoderamiento local: la cooperación médica [En línea] [Citado el: 20 de 01 de 2010.] [http://www.connergorry.com/article\\_pdfs/Pagesfromcaminos\\_42.pdf](http://www.connergorry.com/article_pdfs/Pagesfromcaminos_42.pdf).
4. Boloy Daina y otros, (2008) .Modulo para la gestión de la Información de los colaboradores de la Salud v1.0.Cuba.
5. Definición arquitectura cliente servidor - Monografias.com [En línea] [Citado el: 22 de 01 de 2010.] <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml#element>.
6. Empleados MSD Multiusuario v3.20 [En línea] [Citado el: 20 de 01 de 2010.] <http://www.msdssoft.com/esp/products/empleados/info.htm>.
7. Enterprise Architect- Herramienta de diseño UML. [En línea] 01-2010 <http://www.sparxsystems.com.ar/products/ea.html>.
8. EVA UCI.[En línea] [Citado el: 10 de enero de 2010.] [http://eva.uci.cu/mod/resource/view.php?id=2241&subdir=/Conferencias\\_IS1\\_2007-2008](http://eva.uci.cu/mod/resource/view.php?id=2241&subdir=/Conferencias_IS1_2007-2008).
9. EVA UCI. [En línea] 2009. <http://eva.uci.cu/mod/resource/view.php?id=22668>.
10. EVA UCI. [En línea] 2009. <http://eva.uci.cu/mod/resource/view.php?id=14069>.
11. ExtJS lo bueno, lo malo y lo feo | Desarrollo en Web [En línea] [Citado el: 22 de 01 de 2010.] <http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>.
12. García, Victor Martin. - » Entendiendo AJAX [En línea] [Citado el: 22 de 01 de 2010.] <http://www.victormartingarcia.com/2007/04/02/entendiendo-ajax/>.
13. @GesRRHH v2.0 [En línea] [Citado el: 15 de 2 de 2010.] <http://www.e-rem.net/>.
14. Infomed, Portal de Salud de Cuba [En línea] [Citado el: 20 de 01 de 2010.]<http://www.sld.cu/>.
15. Ingenieria de SoftwareUML - Monografias.com [En línea] [Citado el: 21 de 01 de 2010.] <http://www.monografias.com/trabajos5/insof/insof.shtml>.
16. Linux Para Todos - Un buen sitio donde empezar [En línea] [Citado el: 22 de 01 de 2010.] <http://www.linuxparatodos.net/portal/>.
17. Manual del usuario de PostgreSQL Sistemas Gestores de Bases de Datos [En línea] [Citado el: 22 de 01 de 2010.] <http://www.scribd.com/doc/5703210/Manual-del-usuario-de-PostgreSQL>.

18. Mozilla Firefox, el navegador web del momento. [En línea] [Citado el: 22 de 01 de 2010.] <http://www.maestrosdelweb.com/editorial/firefox/>.
19. El Mundo Informático. [En línea] 2007 <http://jorgesaavedra.wordpress.com/category/patrones-grasp/>.
20. Penichet Ulibarri, Juan Manuel. Universidad de las Américas Puebla. [En línea] 17 de mayo de 2004. [Citado el: 20 de febrero de 2010.]
21. Patrón "Modelo-Vista-Controlador" [En línea] [Citado el: 22 de 01 de 2010.] <http://www.proactiva-calidad.com/java/patrones/mvc.html>.
22. Pozo Alfonso, Yoelvis (2010). Diseño e Implementación de la Base de Datos del Sistema de Colaboración Médica v2.0. Cuba.
23. ¿Qué es Javascript? [En línea] [Citado el: 22 de 01 de 2010.] <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>
24. Qué es PHP [En línea][Citado el: 22 de 01 de 2010.] <http://www.desarrolloweb.com/articulos/392.php>
25. Qué es RUP [En línea] [Citado el: 20 de 01 de 2010.] <http://www.scribd.com/doc/12812403/Que-es-RUP>.
26. ¿Qué es un Servidor Web? « Linux, Java y programación [En línea] [Citado el: 22 de 01 de 2010.] <http://www.misrespuestas.com/que-es-un-servidor-web.html>
27. ¿Qué es un servidor? - Definición y explicación del término servidor [En línea] [Citado el: 22 de 01 de 2010.] <http://www.masadelante.com/faqs/servidor>
28. Selsoft [En línea] [Citado el: 15 de 2 de 2010.] [HT://www.desoft.cu/.../Selsoft/.../Default.aspx?http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/ulibbarri\\_p\\_jm/capitulo4.pdf](HT://www.desoft.cu/.../Selsoft/.../Default.aspx?http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/ulibbarri_p_jm/capitulo4.pdf).
29. Sistemas Gestores de Bases de Datos [En línea] [Citado el: 22 de 01 de 2010.] [http://pdf.rincondelvago.com/sistemas-gestores-de-bases-de-datos\\_1.html](http://pdf.rincondelvago.com/sistemas-gestores-de-bases-de-datos_1.html).
30. Sobre Linux | El rincón de Linux [En línea] [Citado el: 22 de 01 de 2010.] [http://www.linux-es.org/sobre\\_linux](http://www.linux-es.org/sobre_linux).
31. Wikimetal RRHH v3.0 [En línea] [Citado el: 15 de 2 de 2010.] <http://www.wikimetal.net/>.

### 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 Free Software Foundation 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 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. Imágenes de propuesta del sistema.

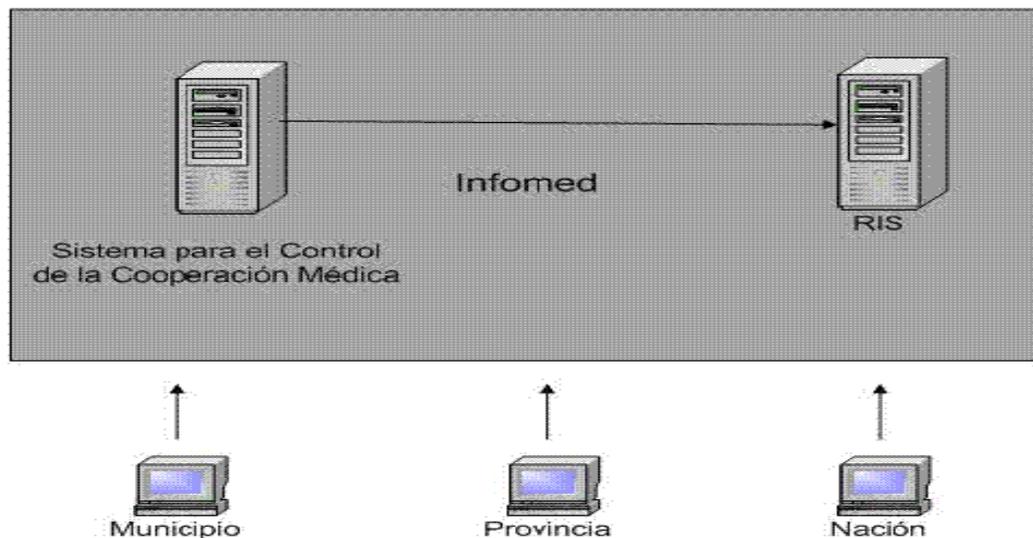


Fig. 4 Propuesta del funcionamiento del sistema.

El sistema contará con una política de asignación de roles, los cuales se encuentran definidos en tres categorías, que se representan a continuación en un árbol de jerarquía para una mejor comprensión.

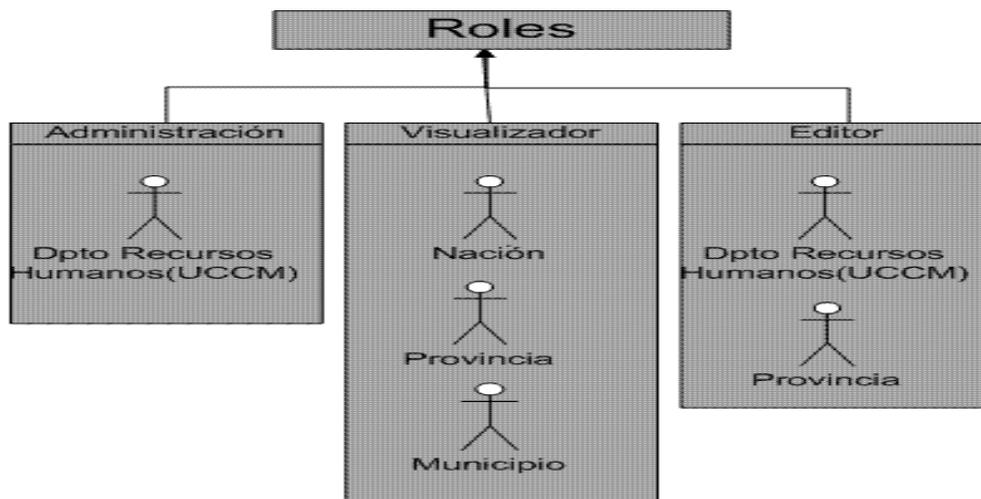


Fig. 5 Asignación de roles.

Anexo 2. Diagramas de actividades.

Diagrama de Actividades (Asignar Misión)

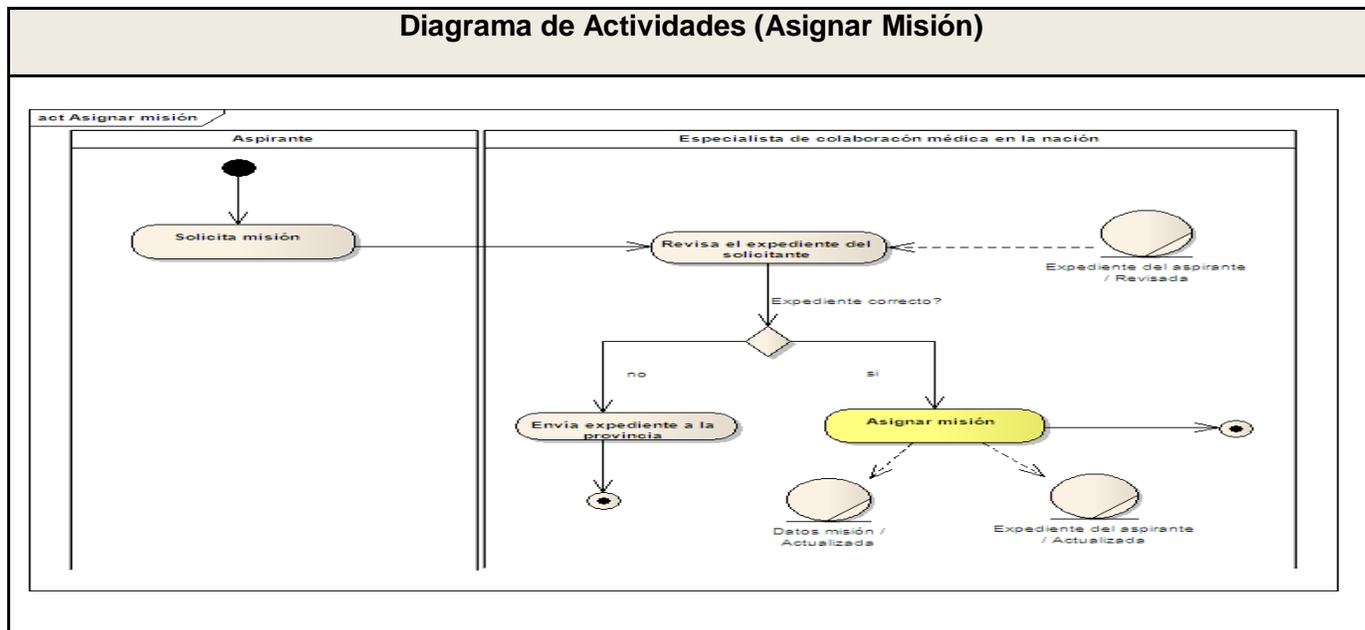


Fig.1 Diagrama de actividades: CU Asignar Misión.

Diagrama de Actividades (Realizar movimiento)

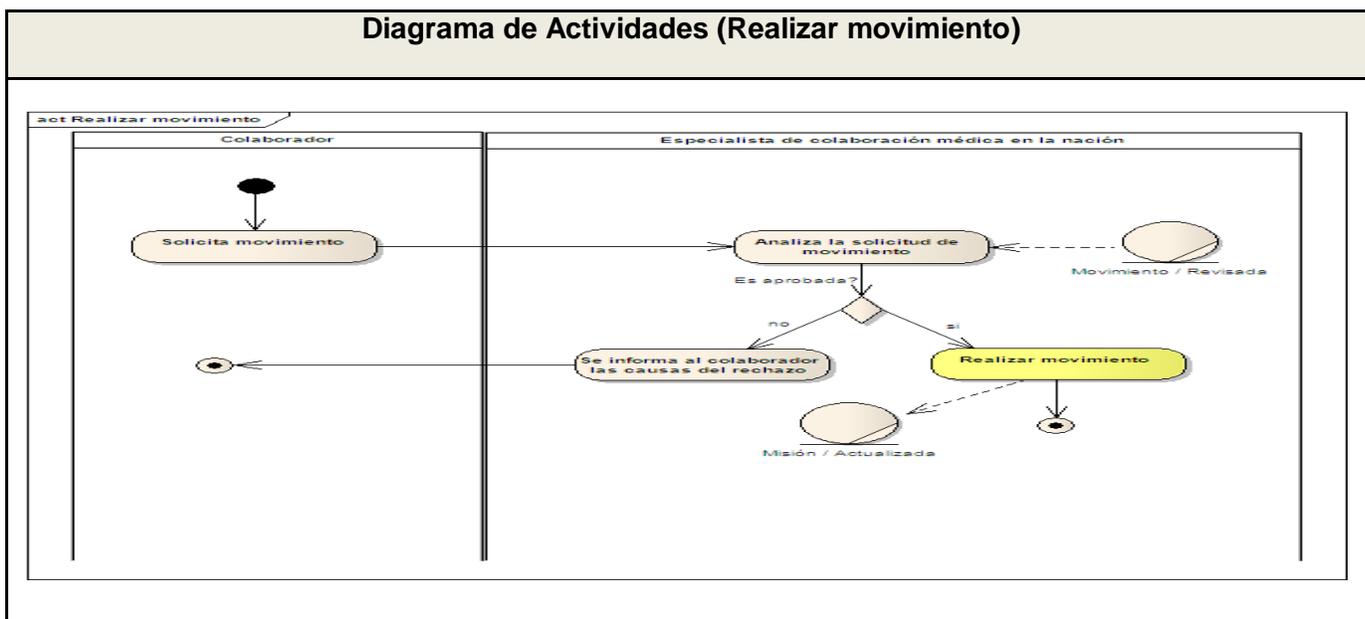


Fig. 3 Diagrama de actividades: CU Realizar movimiento.

Diagrama de Actividades (Confeccionar expediente del aspirante a colaborador)

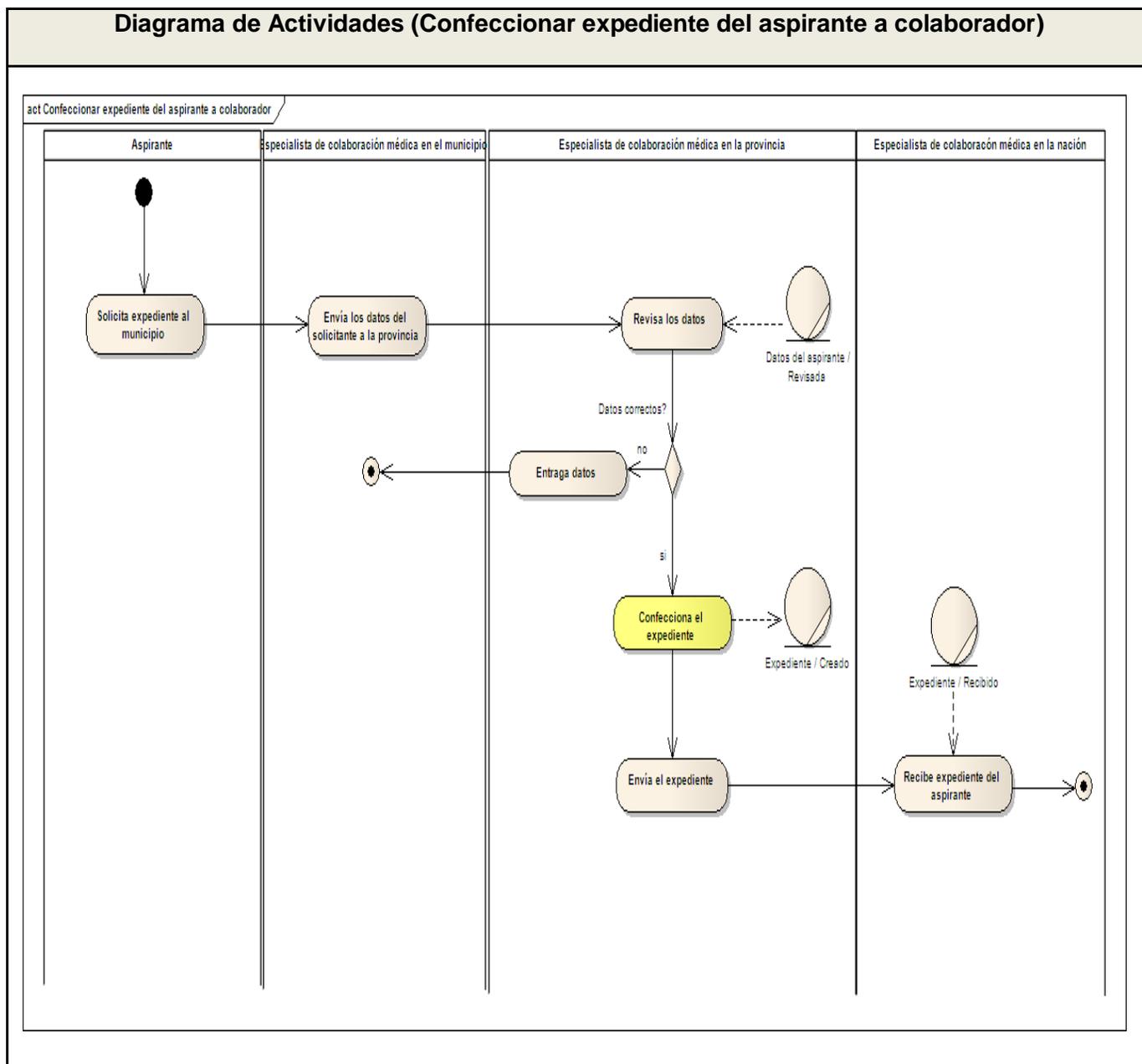


Fig. 2 Diagrama de actividades: CU Confeccionar Expediente del Aspirante a Colaborador.