

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
Facultad 7



TÍTULO: Módulo Registro de Estudiantes v1.1

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN INFORMÁTICA**

AUTORES: Zeydi Valera Lara
Fabio Astray Campos Ramos

TUTORA: Ing. Jany Coto García

CO-TUTORA: Ing. Anabel Martín Aparicio

ASESOR: Ing. William Sóñora Cruz

Ciudad de la Habana, Junio del 2009

“Año del aniversario 50 del triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 25 días del mes de Junio del año 2009.

Zeydi Valera Lara

Autora

Fabio Astray Campos Ramos

Autor

Ing. Jany Coto García

Tutora

Ing. Anabel Martín Aparicio

Co-Tutora

DATOS DE CONTACTO

Ing. Jany Coto García. (jcoto@uci.cu). Graduada de Ingeniera en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en su segunda graduación. Actualmente profesora en adiestramiento en el departamento de Sistemas Digitales de la Facultad 7. Profesora del Área Temática Sistemas de Apoyo a la Salud, perteneciente al Grupo de Implementación de Negocio y jefa del Módulo Registro del proyecto Docencia.

Agradecimientos

A mis padres y familia por darme la estabilidad emocional, económica, sentimental, para poder llegar hasta este logro, que definitivamente no hubiese podido ser realidad sin ustedes. Mi vieja, serás siempre mi inspiración para alcanzar mis metas, por enseñarme que todo se aprende y que todo esfuerzo es al final recompensa.

A mi abuela Adis, sé que me ves y estas orgullosa de mí.

A todos mis amigos pasados y presentes; pasados por ayudarme a crecer y madurar como persona y presentes por estar siempre conmigo apoyándome en todo las circunstancias posibles, también son parte de esta alegría.

A mi novia en quien se ven cristalizados todos mis sueños y su familia.

A mi compañera de tesis por su abnegado trabajo, a nuestro asesor-tutor por su apoyo incondicional, a Pompa, gracias hermano por tanta paciencia con todos nosotros.

A todos los profesores que me han ayudado a ser el ser humano y profesional que soy hoy, muchas gracias.

A todas aquellas personas quienes buscan usar la ciencia en pos de ayudar a los demás.

Muchas gracias a todos de corazón, Fabio.

Agradecimientos

Quisiera agradecer:

A mi abuela, mi mamá, mi Yusma, a mi hermanito lindo y a mi novio William, porque son la parte más importante de mi vida, son mi fuerza, mi apoyo, son la familia que siempre quise tener.

A mi abu Mima, a mis tíos: Papilla, Millita e Ileana, a mis primos que son todos maravillosos, en especial mis primotas lindas Saylí, Anabel, Dianita, la Guaji, y todos los que siempre están a mi lado.

A mis amigos Osvaldo e Isabel, a Yele, a mis compañeros de todos estos años en la universidad, a mis profesores pues sin sus enseñanzas no hubiese sido posible alcanzar esta meta.

Al mejor compañero de tesis, no solo por su eficiencia en el trabajo sino también por soportar mis perretas y mis locuras.

A todos, porque son la causa de mi presente.

Muchas gracias, Zeydi.

Dedicatoria

A mis padres, este trabajo es de y para Uds.

Fabio

A mi abuelo Papa, por ser el primero en creer en mí.

A Sobeyda, mi abuela del alma, mi apoyo y mi vida.

A Leydis, mi madre y guía de todos mis éxitos.

A Yusma, por ser el mejor padre del mundo.

A Lachy, mi querido hermano.

A la memoria de mi tío Willy,

Y en especial a mi novio, William, por su apoyo, su amor e incondicionalidad.

Zeydi

RESUMEN

A raíz de las limitaciones técnicas presentadas en el Componente Registro de Estudiantes del Sistema de Gestión de la Información en el Proceso de Formación de Recursos Humanos en la Salud; desarrollado en la Universidad de las Ciencias Informáticas en el año 2008 que impidieron su despliegue y explotación, se decidió desarrollar el Módulo Registro de Estudiantes (MRE) con el propósito de automatizar el proceso de matrícula en los centros de formación adscritos al Ministerio de Salud Pública (MINSAP) en Cuba.

El MRE se implementó con tecnologías actuales para el desarrollo de sistemas de gestión sobre entornos Web. Para ello, se utilizó el framework de aplicaciones Web Symfony, basado en el lenguaje PHP y como gestor de bases de datos PostgreSQL, respetando los principios de independencia y soberanía tecnológica que rigen a la mayoría de los desarrollos en la Universidad de las Ciencias Informáticas (UCI) y en Cuba.

Se concibió una aplicación consistente, flexible y fiable, la cual puede ser reutilizada y adaptada a otros proyectos donde sean necesarias actividades relacionadas con los procesos de matrícula en secretarías docentes.

PALABRAS CLAVES:

Proceso de matrícula, sistemas de gestión, entornos Web, Symfony, PostgreSQL

Tabla de Contenidos

Contenidos

INTRODUCCIÓN.....	1
CAPÍTULO 1: Fundamentación Teórica.....	6
1.1 Sistemas relacionados con el proceso de formación de recursos humanos en secretarías docentes.....	6
1.2 Tendencias y tecnologías actuales	9
1.3 Herramientas utilizadas en el desarrollo.....	18
1.4 Framework y Librería Web	19
CAPÍTULO 2: Características del Sistema	22
2.1 Conceptos asociados al dominio del problema	22
2.2 Objeto de Estudio.....	22
2.3 Propuesta del Sistema	23
2.4 Modelo de Dominio.....	25
2.5 Especificación de los Requerimientos de Software.....	26
2.6 Modelo de Casos de Uso del Sistema.....	35
CAPÍTULO 3: Análisis y Diseño del Sistema	57
3.1 Análisis.....	57
3.2 Diseño	60
3.3 Descripción de las principales clases del diseño	68
3.4 Definiciones de diseño.....	76
3.5 Tratamiento de errores	78
3.6 Estándar de codificación	79
CAPÍTULO 4: Implementación.....	81
4.1 Diseño de la Base de Datos.....	81
4.2 Modelo de Despliegue.....	86
4.3 Modelo de Implementación	87
4.4 Integración con otros sistemas	96
CONCLUSIONES	99
RECOMENDACIONES.....	100
REFERENCIAS BIBLIOGRÁFICAS	101
BIBLIOGRAFÍA.....	104
ANEXOS.....	107
GLOSARIO.....	110

INTRODUCCIÓN

El desarrollo acelerado de la industria del software en Cuba ha matizado el mundo empresarial cubano en los albores del siglo XXI. En el ámbito nacional, muy pocos se atreven a negar el papel cada vez más importante que juegan en la economía cubana las aún incipientes producciones de soluciones, productos y servicios relacionadas con las tecnologías de la información de factura criolla.

La industria del software en Cuba constituye un pilar para gran parte de los objetivos económicos, políticos y sociales del país. Se ha convertido en la llave para abrir el camino que eleve al proceso revolucionario socialista cubano a los niveles humanos y de eficiencia que el mundo y las sociedades de hoy aspiran.

En el desarrollo de esta industria, el sector de la salud ha sido priorizado, no solo por constituir una de las bases de la Revolución Cubana, sino porque también representa un mercado potencial importantísimo; en el cual Cuba podría insertar sus producciones de software basadas en el prestigio internacional de la medicina cubana y de la industria biofarmacéutica, así como en el área de influencia ganada por el gobierno revolucionario a partir de las misiones internacionalistas médicas que desde Cuba se extienden por todas las regiones del mundo. Por tanto, representa una oportunidad de aportar grandes beneficios a todos los involucrados e interesados y en especial a toda la nación.

Entre los principales participantes en el desarrollo de soluciones y productos de software y la prestación de servicios informáticos relacionados con la esfera de la salud en Cuba, se encuentran las empresas: DESOFT y SOFTEL ambas adscriptas al Ministerio de la Informática y las Comunicaciones (MIC).

La Universidad de las Ciencias Informáticas, surge al calor de la batalla de ideas, y es una universidad de nuevo tipo que vincula la docencia y la producción. Actualmente tiene el mayor volumen de contratos y ejecuta un gran número de proyectos y servicios tanto para el mercado nacional como para la exportación de productos, soluciones y servicios informáticos relacionados con el sector de la salud. El desarrollo de estos proyectos informáticos está basado en la especialización de cada una de las 13 facultades que la conforman (10 ubicadas en su sede principal en la Ciudad de la Habana y 3 distribuidas en diferentes regiones del país: La Habana, Ciego de Ávila y Granma).

La facultad dedicada a responder directamente por la mayoría de los desarrollos que en materia de salud se necesiten, cualquiera que sea su propósito comercial, es la número siete, la cual posee cuatro años de trabajo y a pesar de su juventud, se ha visto enfrascada en proyectos de gran envergadura. La estructura organizativa de esta facultad está diseñada para agrupar y organizar un conjunto de proyectos que coinciden con el tema de desarrollo e investigación. Esta estructura la encabezan dos grandes polos productivos: el polo de Procesamiento de Imágenes y Señales y el de Sistemas de Información para la Salud. Cada polo está compuesto por varias áreas temáticas las cuales desarrollan los distintos proyectos en los que está involucrada la facultad.

Una de las áreas temáticas es Sistemas de Apoyo a la Salud (SAS), que tiene como propósito el desarrollo y soporte de aquellos productos de software que informaticen los procesos no asistenciales del MINSAP. Actualmente, esta área tiene asociado un grupo de proyectos en desarrollo, dentro de los cuales se encuentran: Sistema para la Planificación y el Balance de Suministros Médicos, Colaboración Médica, Sistema de Informatización Estadístico Complementario de Salud, y el Sistema de Gestión de la Información en el Proceso de Formación de Recursos Humanos en la Salud.

Este último surge como necesidad de agrupar los procesos de gestión de la información en las secretarías docentes de las instituciones clínico docentes cubanas, en un único sistema, que se adapte con facilidad a los cambios y necesidades de las mismas, y que contenga todas las especialidades médicas con las que ha crecido la educación médica superior cubana en los últimos años.

El Componente Registro de Estudiantes (CRE) es uno de los componentes del Sistema de Gestión de la Información en el Proceso de Formación de Recursos Humanos en la Salud, desarrollado en el curso académico 2007-2008. Este proponía automatizar el proceso de matrícula de los estudiantes en los centros de formación adscriptos al MINSAP.

Funcionalidades como la configuración personalizada de las planillas de inscripción en cada uno de los centros de estudios, la designación de los centros que brindarán matrícula en un curso determinado y la asignación de las planillas previamente configuradas a los centros, hacían a este producto lo suficientemente configurable con el

objetivo de mejorar la gestión y el control de la información generada en el proceso de formación y superación del profesional de la salud cubana.

En la actualidad, existen actividades que no han sido automatizadas dentro del proceso que informatiza esta solución, entre las cuales se encuentra: la conformación de las brigadas docentes y la gestión de los movimientos de estudiantes, dígase dar baja, licencia o traslado a un estudiante.

El CRE se desarrolló con tecnologías para la Web utilizando herramientas de software libre, siguiendo así las políticas del MINSAP y del país. Se diseñó bajo una arquitectura acorde con las tendencias contemporáneas, lo cual le permitió ser reusable y adaptable en un buen grado. Pero, su versión actual, aún tiene limitado su desarrollo y soporte a los cambios propios de los requerimientos funcionales y al grado de experticia y de actualización en el conocimiento de las nuevas tendencias tecnológicas de los responsables de su desarrollo y mantenimiento.

Además, no posee una estructura estándar en su codificación, carece de una documentación detallada tanto del código como del funcionamiento de sus partes, lo cual limita el entendimiento y el mantenimiento de la solución. Su evolución se circunscribe al conocimiento que tiene el número restringido de sus desarrolladores por lo que la posibilidad de adaptarse a las nuevas tecnologías queda sujeta al grado en que estos puedan asimilar esas nuevas tendencias y estándares.

A este ambiente se une la necesidad que tiene la red médica nacional de un rápido flujo de datos de los estudiantes, lo cual enmarca la investigación en el CRE, para proponer y desarrollar una solución a las nuevas necesidades.

Dada la situación anteriormente planteada, se define como **problema a resolver**: Limitada escalabilidad de la arquitectura del Componente Registro de Estudiantes del Sistema de Gestión de la Información en la Formación de Recursos Humanos en la Salud.

Se define como **objeto de estudio** el Sistema de Gestión de la Información en la Formación de Recursos Humanos en la Salud del área temática Sistemas de Apoyo a la Salud.

El **campo de acción** se enmarca en el Componente Registro de Estudiantes del Sistema de Gestión de la Información en la Formación de Recursos Humanos en la Salud del área temática Sistemas de Apoyo a la Salud.

Para dar solución al problema expuesto se propone como **objetivo general** desarrollar Módulo Registro de Estudiantes v1.1, siguiendo los lineamientos de arquitectura establecidos por el área temática Sistemas de Apoyo a la Salud.

Las tareas investigativas propuestas para dar cumplimiento al objetivo general se definen de la siguiente manera:

- Asimilar las tecnologías (PHP 5.0, YAML, XML Web Services, metodología AJAX, PostgreSQL, HTML, Propel, JavaScript, framework Symfony) y la arquitectura definida por el área temática Sistemas de Apoyo a la Salud para la implementación de la solución.
- Examinar los servicios públicos de las Unidades de Salud (US), Sistema de Autenticación, Autorización y Auditoría (SAAA), Registro de Ubicación (RU) del Sistema de Información para la Salud, para establecer la integración requerida con los mismos.
- Proponer un diseño flexible para el sistema basado en el patrón arquitectónico Modelo Vista Controlador.
- Desarrollar los artefactos correspondientes a los flujos de trabajo del Proceso Unificado de Desarrollo de Software: “Modelado de Negocio”, “Análisis y Diseño” e “Implementación”.
- Implementar el MRE utilizando el framework Symfony.

Esta investigación cuenta con cuatro capítulos donde se realiza el análisis y fundamentación de la solución que se propone, abarca las tecnologías y herramientas que son usadas para el desarrollo del módulo y que incluye el modelado del negocio, el análisis, diseño e implementación. Seguidamente, se hace mención al contenido tratado en cada uno de los capítulos:

Capítulo 1 “**Fundamentación Teórica**”: en este capítulo se realiza un estudio del estado del arte del tema tratado y de las tecnologías actuales que facilitarán el desarrollo de la investigación, describiendo las herramientas y tecnologías a utilizar.

Capítulo 2 “**Características del Sistema**”: se desarrolla el objeto de estudio, los objetivos y se realiza una descripción del modelo del negocio, los requisitos y los casos de uso del sistema.

Capítulo 3 “**Análisis y Diseño del Sistema**”: en este capítulo se realiza el análisis y diseño del sistema donde se muestran los diagramas generados en este flujo.

Capítulo 4 “**Implementación**”: se realiza la implementación en términos de componentes y se describe cómo se organiza el sistema especificado en el diagrama de despliegue. También se muestra el modelo de datos, donde se realiza la descripción de sus tablas y campos.

Capítulo 1: Fundamentación Teórica

CAPÍTULO 1: Fundamentación Teórica

El objetivo de este capítulo es la realización de un análisis del estado del arte y la situación actual de los principales productos, soluciones y servicios informáticos relacionados con el proceso de formación de recursos humanos, en instituciones docentes del MINSAP en Cuba y el mundo. Se expone el estudio de las tecnologías, metodologías y herramientas a utilizar en la presente investigación.

1.1 Sistemas relacionados con el proceso de formación de recursos humanos en secretarías docentes.

El sector de la salud en Cuba se ha convertido en una importante fuente de ingresos para la economía, además de una bandera insignia del marcado carácter internacionalista. El país invierte innumerables esfuerzos en mantener un alto nivel científico en los profesionales de esa rama que egresan de las facultades de ciencias médicas pertenecientes al MINSAP; por lo cual, constantemente se encuentra perfeccionando los planes de estudio de las carreras universitarias que allí se estudian. El principal reto en la formación de recursos humanos en este sector está dado por la dinámica en la formación de personal médico calificado; que abarca un carácter universal trayendo a las universidades médicas cubanas estudiantes de diversos países haciendo que los procesos de gestión de la información en esta área sean más complejos. (1)

A través de los años han habido varios intentos de informatizar estos procesos en el sector de la salud en Cuba, debido a la importancia que tiene mantener un estricto control y manejo eficiente de este capital humano. Pero las soluciones desarrolladas hasta el momento son desarrollos a la medida, por lo general soluciones locales que resuelven los problemas particulares de cada centro de estudios, soportados por varias plataformas tecnológicas y con un deficiente mantenimiento y actualización de sus funcionalidades.

Ejemplos de aplicaciones que han sido desarrolladas con la finalidad de gestionar la información referente a la gestión de recursos humanos en centros de estudios superiores se tienen:

Ámbito Internacional

Capítulo 1: Fundamentación Teórica

UNIVERSITAS XXI – ACADÉMICO: sistema que automatiza todos los procesos que tienen relación con el alumnado, con la planificación y con el seguimiento de los recursos docentes. Su primera versión fue desarrollada por el equipo de desarrollo de la Oficina de Cooperación Universitaria (OCU), con sede en España, en el año 2004.

Las funcionalidades de la aplicación abarcan desde la organización de las pruebas de acceso y preinscripción de los estudiantes a la tramitación de los títulos pasando por la matrícula, la calificación de las actas, el control del expediente, estadísticas, gestión de becas nacionales e internacionales, organización de prácticas en empresas, así como lo referente a la gestión económica de la actividad académica de la universidad. La definición de los planes de estudios y la planificación de la docencia son los pilares básicos de la aplicación y garantizan la integración y coherencia de toda la información registrada. UNIVERSITAS XXI - ACADÉMICO está desarrollado con tecnologías Oracle.
(2)

Sistema Integrado de Registros Estudiantiles, ULA-SIRE: es un conjunto integrado de subsistemas informáticos que brindan apoyo a las Oficinas de Registros Estudiantiles (ORE) de la Universidad de Los Andes. Su implantación apunta a la automatización de los procesos, mejor integración de la información y a la toma de decisiones de forma más ágil y oportuna. Dispone de una serie de módulos, dentro de los cuales se encuentran: El Sistema de Apoyo a la Coordinación del Proceso de Inscripción (ACPI), el Sistema de Registro y Control de Calificaciones y el Sistema de Inscripción para estudiantes, entre otros. (3)

XesCampus: es un sistema de información de gestión académica construido en torno a una base de datos relacional a la que se proporciona acceso mediante tecnologías estándar de Internet. Desarrollada bajo tecnología .Net, la aplicación está dividida en varios módulos que abordan procesos característicos de gestión académica como planificación, matrícula, expediente, títulos, entre otros.

El enfoque del sistema es integral e integrado, pretende ser una solución de gestión y de análisis que agrupe toda la información relacionada con la labor académica y que, por tanto, incluya todos los procesos de la institución. Al mismo tiempo, todos estos procesos están integrados con el resto de sistemas de información de la organización, es decir, con la gestión de recursos humanos y la económica. (4)

Capítulo 1: Fundamentación Teórica

Ámbito Nacional

AKADEMOS: sistema de gestión académica desarrollado y utilizado en la Universidad de las Ciencias Informáticas, el cual brinda una interfaz común para todos sus usuarios, además ofrece la información actualizada en tiempo real de las diez facultades de la universidad. Está implementado sobre la filosofía Web.

Sistema de Gestión de la Nueva Universidad, SIGENU: desarrollado en el Instituto Superior Politécnico “José Antonio Echeverría”, se utiliza en la mayoría de los centros de educación superior y contempla las sedes universitarias municipales. Cuenta con varios módulos entre ellos, Matrícula, Plan de Estudio, Codificadores, Control Docente, Réplica y Seguridad. Está desarrollado bajo plataforma Java. (5)

Sistema de Gestión de Información en el Proceso de Formación de Recursos Humanos en Salud: desarrollado en la Universidad de las Ciencias Informáticas en el año 2007, sobre tecnologías Web, corriendo sobre un servidor LAMP (Linux, Apache, MySQL, PHP). Surge como necesidad de agrupar los procesos de gestión de la información de las secretarías docentes en las instituciones clínico docentes cubanas, en un único sistema. Fácil de adaptar a los cambios y necesidades de las mismas; y que contenga todas las especialidades médicas con las que ha crecido la educación médica superior cubana en los últimos años. (6)

Durante el curso académico 2007-2008, se desarrolla un componente para automatizar el proceso de matrícula de los estudiantes de los centros adscriptos al MINSAP. El mismo, presentó algunas limitaciones debido a la arquitectura con la que fue definido.

Después de este análisis realizado sobre los sistemas relacionados con el objeto de estudio y el campo de acción de esta investigación, se decidió no utilizar ninguno de estos y las razones se explican a continuación: los sistemas internacionales mencionados no son la mejor opción. La mayoría están desarrollados con software bajo licencia propietario, lo cual podría traer alguna repercusión económica y legal en el futuro, si no se garantiza el pago de los mismos para su explotación en el país. Además no se ajustan a las necesidades reales de captura de información para la enseñanza superior en los centros de estudios de las ciencias médicas en Cuba.

Capítulo 1: Fundamentación Teórica

En el caso de los sistemas nacionales, como AKADEMOS, el CRE del Sistema de Gestión de Información en el Proceso de Formación de Recursos Humanos en Salud y SIGENU, tampoco resuelven la problemática actual. La captura de información que realizan no se ajusta a las particularidades de la información de los estudiantes que se registran en los centros de estudios superiores de las ciencias médicas y tanto AKADEMOS como el CRE son proyectos en desarrollo, aunque el primero con un grado de terminación mucho mayor. Por otra parte, el CRE, tiene una arquitectura poco escalable, lo cual limita su desarrollo y soporte ante los cambios propios de las condiciones o capacidades que el sistema debe cumplir para satisfacer las necesidades actuales del cliente.

Es por ello, que se hace necesario asumir el desarrollo del MRE, donde se tengan en cuenta todas las particularidades de la enseñanza de la medicina en Cuba y se obtenga una solución con un alto nivel de prestaciones, flexible y adaptable al entorno y finalmente con un tiempo de explotación útil y suficiente que justifique el costo de la inversión.

1.2 Tendencias y tecnologías actuales

1.2.1 Internet. Su funcionamiento.

Internet es una gran red de redes y es el resultado de la interconexión de miles de computadoras de todo el mundo. Estas computadoras comparten los protocolos de comunicación, es decir que todos hablan el mismo lenguaje para ponerse en contacto unas con otras y funcionando según una estructura de clientes y servidor. (7)

Para lograr la eficiencia de la comunicación entre las diferentes redes deben existir de manera física caminos que las unan, como: fibra óptica, enlaces satelitales, cable coaxial, teléfonos y demás tecnologías; y que las mismas utilicen técnicamente la misma familia de protocolos, TCP/IP. (8)

1.2.2 La información a través de Internet. La Web.

La Web permite una comunicación rica y variada gracias a que puede presentar texto, gráficos, animación, fotos, sonido y vídeo, por las facilidades que brinda la Internet.

Capítulo 1: Fundamentación Teórica

Las ventajas de desarrollar aplicaciones sobre la Web con respecto a las tradicionales aplicaciones de escritorio dejan claro el potencial de estas en el mundo contemporáneo asociado a la era de la información. Entre los principales beneficios que aportan están: (9)

- Desarrollo barato, sencillo y rápido de soluciones de software.
- Acceso ubicuo, sin necesidad de distribución e idealmente con pocos requerimientos técnicos.
- Datos centralizados y fácil integración de datos de múltiples fuentes.
- Desarrollo de comunidades que dan valor a las aplicaciones.

Aunque también puede presentar algunos inconvenientes, como puede ser la necesidad de una conexión permanente y rápida, lo cual podría ser la principal limitante en el desarrollo de soluciones empresariales sobre la Web.

1.2.3 Estilos Arquitectónicos

“Los estilos expresan la arquitectura en el sentido más formal y teórico, describen una clase de arquitectura, o piezas identificables de las arquitecturas empíricamente dadas.” (10)

De manera general, los estilos permiten sintetizar estas estructuras de soluciones que luego serán refinadas en el diseño y definir los posibles patrones para una aplicación, evitando así errores arquitectónicos y permitiendo evaluar las arquitecturas alternativas.

1.2.3.1 Arquitectura Orientada a Servicios (SOA)

La Arquitectura Orientada a Servicios se basa en la definición de pequeñas porciones de funcionalidad auto-contenidas e independientes entre sí, que son los denominados servicios. Cada servicio expone una tarea bien definida y se comunicará con otros servicios para así llevar a cabo dicha tarea. La aplicación final simplemente se encargará de orquestar un conjunto de servicios, añadir su lógica de programa particular y presentar la interfaz al usuario final.

Las arquitecturas SOA comienzan a imponerse debido a las numerosas ventajas que proporcionan, como son la reducción del tiempo de desarrollo de nuevas soluciones de software y de los costos de desarrollo y el mantenimiento de estas. Además permite el

Capítulo 1: Fundamentación Teórica

desarrollo de sistemas más flexibles, ágiles y escalables, así como la interoperabilidad de las aplicaciones. (11)

1.2.3.1.1 Servicios Web. (Web Services)

Un Servicio Web es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios para intercambiar información en redes de ordenadores. La interoperabilidad de los mismos se consigue mediante la adopción de estándares abiertos.

Una de las razones principales por lo cual los servicios Web son muy prácticos y útiles es la gran independencia que pueden aportar entre la aplicación que usa el servicio Web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro. Esta flexibilidad será cada vez más importante, dado que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más utilizada. (12)

1.2.4 Patrones de Arquitectura y Diseño

“Los patrones expresan esquemas de organización estructural fundamentales para los sistemas de software. Estos se ocupan de cuestiones que están más cerca del diseño, la práctica, la implementación, el proceso, el refinamiento y el código.” (13)

Su objetivo principal es crear un lenguaje común para comunicar experiencia sobre los problemas y sus soluciones. Existen varios tipos de patrones que dependen del nivel de abstracción, del contexto particular o de la etapa del proceso de desarrollo en la que se encuentre. Seguidamente, se explica el patrón Modelo Vista Controlador (MVC) que se implementó en el desarrollo de la aplicación.

1.2.4.1 Modelo Vista Controlador (MVC)

La lógica de una interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si se realiza un diseño ofuscado, es decir, una mezcla de los componentes de interfaz y de negocio, entonces la consecuencia será que cuando se necesite cambiar la interfaz, se tendrán que modificar trabajosamente los componentes de negocio.

Capítulo 1: Fundamentación Teórica

El MVC propone un diseño que desacopla la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos. Los elementos de este patrón son: el modelo, la vista y el controlador. (Ver Anexo 1)

Este modelo aporta determinadas ventajas, dentro de las que están: la posibilidad de tener diferentes vistas para un mismo modelo. Además crea independencia del funcionamiento y facilita el mantenimiento en caso de errores. Permite el escalamiento de la aplicación en caso de ser requerido y proporciona un mecanismo de configuración a componentes más complejos.

1.2.4.2 Alta Cohesión y Bajo Acoplamiento

Alta Cohesión y Bajo Acoplamiento son dos patrones que pueden verse separados aunque están íntimamente ligados. De hecho si se aumenta mucho la cohesión del sistema, provoca una reducción del acoplamiento, y por el contrario si se aumenta mucho el acoplamiento, se verá disminuida la cohesión.

Alta Cohesión es la información que se gestiona en un servicio determinado, es decir, que la información que almacena una clase debe ser coherente y estar en mayor medida relacionada con la clase.

El **Bajo Acoplamiento** hace referencia a las relaciones que tienen las clases entre sí dentro de un sistema. De tal forma, que para que no se vean afectadas necesariamente al producirse una modificación en alguna de ellas, se tengan las mínimas dependencias posibles. Así se pueden realizar modificaciones en partes del programa sin necesidad de modificar todo el sistema.

1.2.5 Protocolos y Estándares

Debido a que cada servicio Web puede estar implementado en una tecnología heterogénea, es necesario cumplir una serie de estándares y de utilizar protocolos para hacer posible la comunicación entre ellos. Los más utilizados son: XML, SOAP, WSDL, HTTP.

1.2.5.1 XML (Extensible Markup Language)

Capítulo 1: Fundamentación Teórica

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con posibilidades mucho mayores. Permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. Se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. (14)

1.2.5.2 SOAP (Simple Object Access Protocol)

SOAP es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Es un protocolo liviano para el intercambio de información en forma descentralizada. Consta de tres partes: define un marco para describir lo que hay en un mensaje y la forma de proceso; un conjunto de reglas de codificación para expresar los casos de solicitud de datos definidos; y un convenio para la representación de llamadas a procedimientos remotos y respuestas. (15)

1.2.5.3 WSDL (Web Services Description Language)

WSDL describe la interfaz pública de los servicios Web. Está basado en XML y describe los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se ligan al protocolo concreto de red y al formato del mensaje. (16)

1.2.5.4 HTTP (Hypertext Transfer Protocol)

HTTP es el protocolo usado en cada transacción de la Web. Define la sintaxis y la semántica que utilizan los elementos software de la arquitectura Web para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. La información transmitida se llama recurso y se identifica mediante una URL. (17)

1.2.6 Lenguajes de programación

Definir qué es y qué no es un lenguaje de programación es algo complicado debido a la variedad de conceptos que muchos autores han emitido sobre el tema. Pero en informática, puede definirse como aquel elemento que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que se ponen a disposición

Capítulo 1: Fundamentación Teórica

del programador para que este pueda comunicarse con los dispositivos hardware y software existentes.

Actualmente existen diferentes lenguajes de programación para desarrollar en la Web. Estos han ido surgiendo debido a las tendencias y necesidades de las distintas plataformas existentes y se clasifican en dos grandes grupos: los lenguajes del lado del cliente y los lenguajes del lado del servidor.

Considerados como los más utilizados a nivel mundial, dentro del primer grupo están: HTML, JavaScript, Visual Basic Script, CSS, entre otros; y en el segundo grupo, se pueden mencionar: PHP, JAVA, ASP, JSP, Python, Ruby y Perl. (18)

1.2.6.1 Lenguajes del lado del Cliente

➤ **JavaScript:**

Lenguaje interpretado que no requiere compilación y que es utilizado principalmente en páginas Web. Se integra dentro del código HTML de estas y se ejecuta en el navegador al mismo tiempo que las sentencias van descargándose junto con el código HTML. Brinda rapidez a la aplicación Web en el momento de las validaciones de los formularios. La mayoría de los navegadores en sus últimas versiones interpretan código JavaScript, por lo que es el lenguaje de programación del lado del cliente más utilizado en el mundo. (19)

➤ **HTML:**

HTML es diseñado para estructurar textos, y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web. Este indica al navegador donde colocar cada texto, imagen o video, y la forma que tendrán estos en una página. Es un lenguaje muy sencillo y sumamente flexible. (20)

1.2.6.2 Lenguaje del lado del Servidor

➤ **PHP:**

PHP es un lenguaje script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. No necesita ser compilado para ejecutarse y para su funcionamiento necesita tener instalado

Capítulo 1: Fundamentación Teórica

Apache con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. (21)

Resumiendo, alguna de las cualidades que lo hacen uno de los lenguajes más populares del lado del servidor son:

- Es multiplataforma, es decir, puede ser utilizado en cualquiera de los principales sistemas operativos del mercado actual y es soportado por la mayoría de los servidores Web.
- Es software libre, lo que implica menos costos y servidores más baratos, por lo que se puede utilizar en proyectos sin tener que pagar por su licencia.
- El tiempo de aprendizaje de PHP es muy corto gracias a su simplicidad.
- Es muy rápido y permite su integración con la mayoría de los gestores de base de datos como son: PostgreSQL, MySQL, Oracle, entre otros.

Por último, PHP tiene una de las comunidades de desarrollo más grande de Internet, por lo que es fácil encontrar ayuda, documentación, artículos y demás recursos.

1.2.6.3 Lenguaje de Modelado

- **UML 2.0**

UML 2.0 (Unified Modeling Language), construido bajo el estándar altamente exitoso UML 1.x, el cual se ha convertido en un estándar de la industria para el modelado, diseño y construcción de sistemas de software.

El modelo conceptual ha sido reestructurado completamente y nuevos diagramas han sido incorporados. Los diagramas tradicionales también han sido mejorados, proporcionando un mejor aprovechamiento de los modelos y como consecuencia generar una mayor cantidad de código reduciendo significativamente el ciclo de desarrollo de la aplicación. (22)

1.2.7 Otras Tecnologías

- **AJAX:**

Capítulo 1: Fundamentación Teórica

JavaScript Asíncrono y XML: es una técnica de desarrollo Web para crear aplicaciones interactivas. Éstas se ejecutan en el lado del cliente, o sea, en el navegador de los usuarios, y mantiene comunicación asíncrona con el servidor en segundo plano. De manera que es posible hacer cambios en la misma página sin necesidad de que se recargue, lo que significa aumentar la interactividad, velocidad y usabilidad de la misma.

AJAX es la unión de varias tecnologías que juntas pueden lograr cosas realmente impresionantes. Incorpora: (23)

- Presentación basada en estándares usando XHTML y CSS.
- Exhibición e interacción dinámicas usando el Document Object Model (DOM).
- Intercambio y manipulación de datos usando XML y XSLT.
- Recuperación de datos asíncrona usando XMLHttpRequest y JavaScript poniendo todo junto.

1.2.8 Metodología de Desarrollo

- **RUP**

RUP (Rational Unified Process), es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software. Empleado en diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y tamaño de proyectos. RUP está dirigido por casos de uso, se centra en los aspectos estáticos y dinámicos más significativos del sistema y es un proceso iterativo e incremental. (24)

1.2.9 Sistema Gestor de Base de Datos

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos, asegurando su integridad, confidencialidad y seguridad. Algunos de los sistemas más conocidos son: PostgreSQL, MySQL, Microsoft Access, Microsoft SQL Server, Oracle, entre otros.

Estos sistemas ayudan a realizar una serie de acciones, como son: (25)

- Definición de los datos.
- Mantenimiento de la integridad de los datos dentro de la base de datos.

Capítulo 1: Fundamentación Teórica

- Control de la seguridad y privacidad de los datos.
- Manipulación de los datos.

Dentro de los sistemas gestores de bases de datos anteriormente mencionados, para el desarrollo de la solución propuesta, se decidió utilizar PostgreSQL.

1.2.9.1 ¿Por qué PostgreSQL?

PostgreSQL es el sistema de gestión de base de datos objeto-relacional de código abierto más avanzado hoy en día. Publicado bajo la licencia BSD, la cual permite hacer lo que desee el cliente con el código fuente, siempre que sea reconocido su trabajo. Desarrollado originalmente en el Departamento de Ciencias de Computación de UC Berkeley, fue pionero en muchos de los conceptos de objetos y relacionales que ahora están apareciendo en algunas bases de datos comerciales.

Entre sus principales características cabe mencionar el control de concurrencia multiversión que permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Esta estrategia es superior al uso de bloqueos por tabla o por filas, común en otras bases de datos, eliminando la necesidad del uso de bloqueos explícitos. Además soporta casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con muchos lenguajes de programación. (26)

1.2.10 Servidor Web

Un servidor Web es un programa que sirve para atender y dar respuesta a las distintas peticiones de los navegadores. Brinda los recursos que soliciten a través del protocolo HTTP o el HTTPS. (27)

Se puede destacar que la palabra servidor se le llama tanto al software como a la máquina donde se ejecuta.

1.2.10.1 Servidor Web Apache

Apache es uno de los servidores Web más utilizados en la actualidad y está publicado bajo la licencia BSD. Este servidor, corre en una gran cantidad de sistemas operativos, lo que lo hace prácticamente universal. Es altamente configurable por lo que es muy sencillo

Capítulo 1: Fundamentación Teórica

ampliar sus capacidades, además trabaja con PHP y otros lenguajes script. También permite personalizar la respuesta ante los posibles errores que se pueden presentar en el servidor. Hoy, es considerado una de las herramientas más potentes y configurables unido a su costo gratuito, su robustez, su gran estabilidad y gran extensibilidad. (28)

1.3 Herramientas utilizadas en el desarrollo

Son muchas las herramientas que se utilizan durante la realización de un proyecto de software. Estas son empleadas tanto en la etapa de diseño como en la etapa de construcción de la aplicación y la selección de las mismas está determinada por las tecnologías en que se desarrolle.

1.3.1 Dreamweaver 8

Dreamweaver 8, es un software profesional para la creación de sitios y aplicaciones Web. Las funciones de edición visual del mismo permiten agregar rápidamente diseño y funcionalidad a las páginas. Se pueden crear tablas, editar marcos, trabajar con capas e insertar comportamientos JavaScript, de una forma muy sencilla y visual. Además incluye un software de cliente FTP completo, lo que permite, entre otras cosas, trabajar con mapas visuales de los sitios Web, actualizando el mismo en el servidor sin salir del programa. (29)

1.3.2 Zend Studio 5.5.2

Excelente y robusta plataforma de desarrollo PHP. Asegura el desarrollo de software mediante la combinación del IDE líder para PHP con un entorno de prueba que agiliza la seguridad de la calidad y la integración. Brinda todo lo que se necesita para construir, probar y entregar aplicaciones PHP de alto rendimiento. El mismo se ha diseñado para una amplia gama de programadores y proporciona una facilidad de uso inigualable, escalabilidad y fiabilidad. (30)

1.3.3 EMS Manager 2007 para PostgreSQL

Es una herramienta de alto rendimiento para la administración del servidor de base de datos PostgreSQL. Funciona con cualquier versión de PostgreSQL hasta la más reciente. SQL Manager para PostgreSQL ofrece muchas de las poderosas herramientas de base de datos, como Designer para crear base de datos en pocos clics, Visual Query Builder

Capítulo 1: Fundamentación Teórica

para crear consultas complicadas, potente editor BLOB y muchas más características útiles para una eficiente administración de PostgreSQL. (31)

1.3.4 Embarcadero ER/Studio 6.0

Es una herramienta de modelado de datos, fácil de usar y multinivel para el diseño y construcción de bases de datos a nivel físico y lógico. Direcciona las necesidades diarias de los administradores de bases de datos, desarrolladores y arquitectos de datos que construyen y mantienen aplicaciones grandes y complejas. (32)

1.3.5 Enterprise Architect 7.0

Enterprise Architect es una herramienta CASE (Computer Aided Software Engineering) para el diseño y construcción de sistemas de software. Soporta la especificación UML 2.0, que describe un lenguaje visual por el cual se pueden definir mapas o modelos de un proyecto. Es una herramienta progresiva que cubre todos los aspectos del ciclo de desarrollo del software, proporcionando una trazabilidad completa desde la fase inicial del diseño al despliegue y mantenimiento. También provee soporte para pruebas, mantenimiento y control de cambio. (33)

1.4 Framework y Librería Web

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

Simplifican el desarrollo de las aplicaciones mediante la automatización de muchas de las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al programador a crear código más legible y fácil de mantener.

Librería: conjunto de procedimientos y funciones agrupadas en un archivo con el fin de que puedan aprovecharlas otros programas.

Las librerías Web consisten en una serie de utilidades JavaScript para poder realzar las aplicaciones con componentes visuales tales como animaciones, cuadros de mensajes a medida, ventanas emergentes sin necesidad de abrir otra instancia de navegador, posibilidad de crear cuadros de autocompletar, calendarios, menús, entre otras.

1.4.1 YUI 2.5.0

La Yahoo UI Library, conocida como YUI, se puede traducir como "Librería de componentes de interfaz de usuario de Yahoo". Es un conjunto de utilidades y controles escritos en JavaScript y CSS de código abierto bajo la licencia BSD. La librería completa está dividida en módulos y componentes relacionados con CSS, DOM, eventos, AJAX y otros. Tiene capacidad para generar código ordenado para el programador y brinda la posibilidad de reutilizar controladores y personalizar sus componentes. (34)

1.4.2 Symfony 1.2

Symfony es un framework elegante, estable, productivo y muy bien documentado que simplifica al máximo el desarrollo de aplicaciones web profesionales con PHP 5, utilizando las mejores prácticas y los patrones de diseño más importantes. (35)

Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. 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 los gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft.

1.4.2.1 Mapeo de Objetos a Bases de Datos

El mapeo objeto-relacional (Object-Relational Mapping, o sus siglas ORM) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional. En la práctica esto crea una base de datos orientada a objetos virtual, sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos, básicamente herencia y polimorfismo. (36)

1.4.2.2 Propel

Propel es el ORM clásico de Symfony. Su principal ventaja es que está completamente integrado con Symfony y que decenas de plugins sólo funcionan para Propel. Facilita la

Capítulo 1: Fundamentación Teórica

labor de desarrollo de aplicaciones Web, gracias a la capa que transforma el tratamiento de la base de datos mediante objetos, con la que se puede recuperar, insertar y modificar datos. No es necesario preocuparse por las conexiones de la base de datos y escribir SQL, ni escapar datos o realizar casting, tan solo es necesario definir la base de datos u obtener la definición desde una base de datos ya existente. (37)

1.4.2.3 YAML

Symfony es uno de los framework que ha prescindido del formato XML para sus archivos de configuración y se ha pasado a YAML, un lenguaje mucho más compacto, fácil de leer y escribir. La sintaxis es relativamente sencilla y fue diseñada teniendo en cuenta que fuera muy legible pero que a la vez fuese fácilmente mapeable a los tipos de datos más comunes en la mayoría de los lenguajes de alto nivel. Además, YAML utiliza una notación basada en la indentación y un conjunto de caracteres distintos de los que se usan en XML, haciendo que sea fácil componer ambos lenguajes. (38)

Después del análisis y estudio realizado, se muestra una panorámica lo más actual posible de las tecnologías y tendencias actuales en el mundo del desarrollo Web y aunque se conoce que existe mucha más información de la que se ha podido reflejar aquí de forma muy resumida, se define que las herramientas tratadas en el capítulo son las indicadas para el desarrollo rápido y eficaz del MRE, siguiendo la política del país de desarrollar sistemas informáticos con soberanía tecnológica.

Capítulo 2: Características del Sistema

CAPÍTULO 2: Características del Sistema

En este capítulo se describe la propuesta de solución para ser utilizada en las secretarías docentes del MINSAP. Debido a que existe una incipiente estructura de procesos de negocio, se determina desarrollar la modelación del dominio de la aplicación, para poder comprender mejor el contexto en el cual se enmarca el sistema; lo que describe las relaciones entre sus definiciones.

2.1 Conceptos asociados al dominio del problema

Algunos de los conceptos que se identifican en el modelo de dominio son:

Centro Docente: Es la entidad que se dedica a la enseñanza o comunicación de conocimientos, habilidades, ideas o experiencias a personas que no las tienen con la intención de que las aprendan.

Brigada Docente: agrupa a los estudiantes que ingresan al centro docente.

Secretaria(o) Docente: Es el responsable de ingresar nuevos estudiantes al centro docente. Gestiona las brigadas docentes que se crean en el centro de estudios y los movimientos de estudiantes (bajas, licencias o traslados).

Matrícula: Materializa la incorporación de un aspirante en una lista o registro dentro del centro docente.

2.2 Objeto de Estudio

2.2.1 Situación Problemática

Existen actividades que aún no han sido automatizadas dentro del proceso que informatiza el CRE, entre las cuales se encuentran: la conformación de las brigadas docentes y la gestión de los movimientos de estudiantes, dígame dar baja, licencia o traslado a un estudiante. Pero además, el CRE, pese a su diseño y arquitectura acorde con las tendencias contemporáneas, tiene limitado su desarrollo y soporte a los cambios propios de los requerimientos funcionales y al grado de experticia de los responsables de su desarrollo y mantenimiento.

Capítulo 2: Características del Sistema

No posee una estructura disciplinada en su codificación, carece de una documentación detallada tanto del código como del funcionamiento de sus partes en detalle, lo cual limita el entendimiento y el mantenimiento de la solución. Además su evolución se circunscribe al conocimiento que tiene el número restringido de sus desarrolladores, por lo que la posibilidad de adaptarse a las nuevas tecnologías queda sujeta al grado en que estos puedan asimilarlas. A este ambiente se une la necesidad que tiene la red médica nacional de un rápido flujo de datos de los estudiantes.

2.3 Propuesta del Sistema

El sistema que se propone se encontrará brindando servicio sobre la red de salud cubana INFOMED. A través de ella todos los nodos (secretarías docentes, puestos de trabajos relacionados con el sistema) accederán al sistema, que contará con un módulo de configuración que brindará flexibilidad ante los aspectos relacionados con la configuración del área de trabajo. El mismo dará la posibilidad de configurar las planillas de matrícula que serán destinadas luego a cada unidad de salud designada para impartir clases en un curso determinado, permitiendo también la gestión de las planillas.

El módulo de Registro, brindará al usuario la posibilidad de gestionar la matrícula del estudiante, la gestión y conformación de las brigadas docentes y los movimientos de estudiantes: licencias, bajas y traslados.

En la figura 2.1 se muestra cómo las estaciones clientes o nodos estarán conectadas al servidor central de la red médica INFOMED, donde estará desplegada la aplicación.

Capítulo 2: Características del Sistema

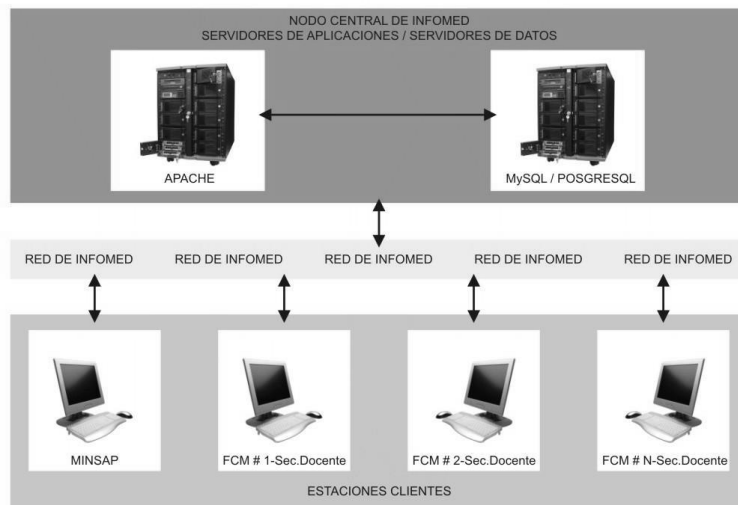


Fig. 2.1 Propuesta del funcionamiento de la aplicación

Pero, existe la posibilidad de que la conectividad no este disponible en todo momento y como consecuencia la utilización de la aplicación se verá afectada para los usuarios finales. Esta es una problemática real en la red telemática de INFOMED, por lo que la propuesta anterior no es la más óptima.

Para solucionar esta situación se propone como alternativa; construir la aplicación que resuelva las necesidades del usuario final y que además posibilite su utilización aún cuando la conectividad no se encuentre disponible. Para ello, se deben establecer en cada una de las unidades docentes del MINSAP, servidores locales donde se despliegue la aplicación con todas sus funcionalidades. Los servidores centrales quedarían para la réplica de datos con el objetivo de garantizar la integridad y seguridad de la información. De esta manera, aunque aumentan los costos asociados al despliegue del sistema, la solución no depende de la conectividad para su funcionamiento garantizando la fiabilidad y eficiencia. Ver figura 2.2.

Capítulo 2: Características del Sistema

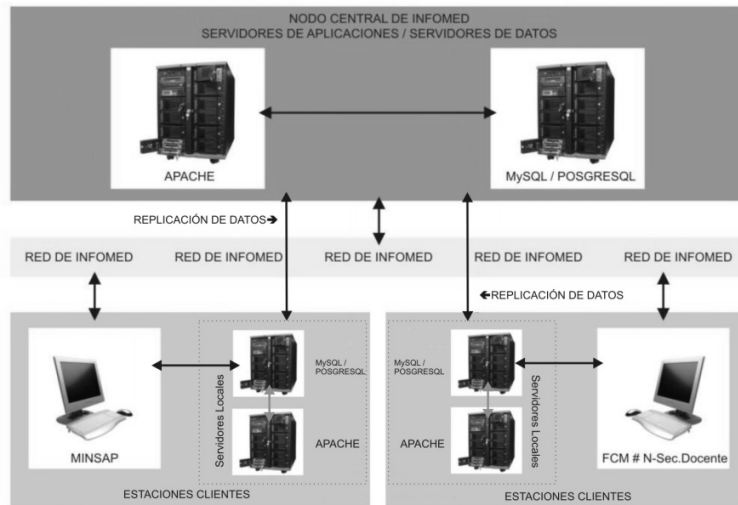


Fig. 2.2 Propuesta final del funcionamiento de la aplicación

2.4 Modelo de Dominio

Los principales tipos de objetos del contexto del sistema y las actividades o eventos que surgen en el entorno del mismo, son capturados en el modelo de dominio y mostrados en los diagramas correspondientes.

En la figura 2.4, aparecen representadas las principales clases que se manejan en el paquete Registro.

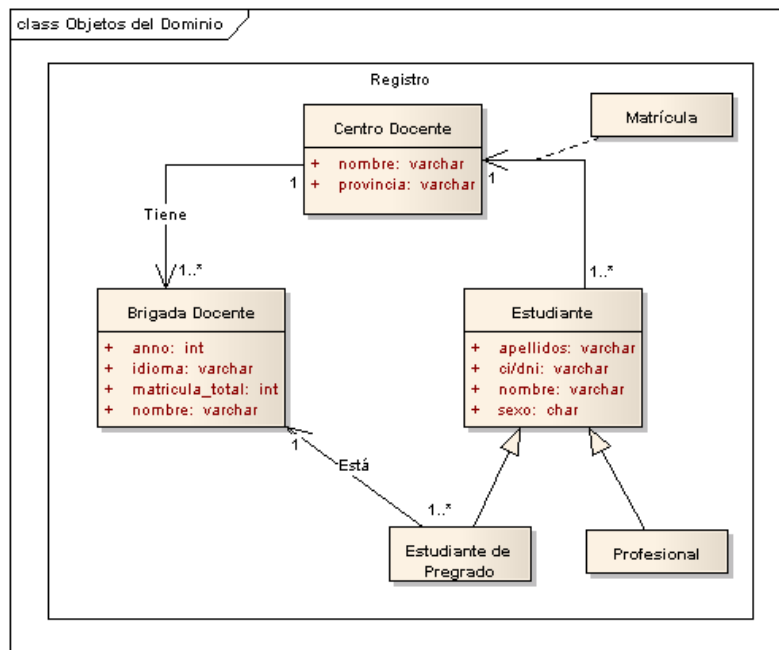


Fig. 2.4 Modelo del dominio del paquete "Registro"

Capítulo 2: Características del Sistema

El centro docente está compuesto por brigadas y la matrícula consiste en la inserción de los estudiantes en una brigada de un centro. En este caso, los estudiantes están divididos en dos categorías, estudiante de pregrado y el profesional, que no es más que aquel graduado universitario que cursa estudios de posgrado. Es por ello, que en un centro docente donde se brinde matrícula tanto de pregrado como de posgrado, hay que ver estos procesos por separados, ya que el profesional no se agrupa por brigadas docentes y esto lo hace diferente en el contexto que se analiza.

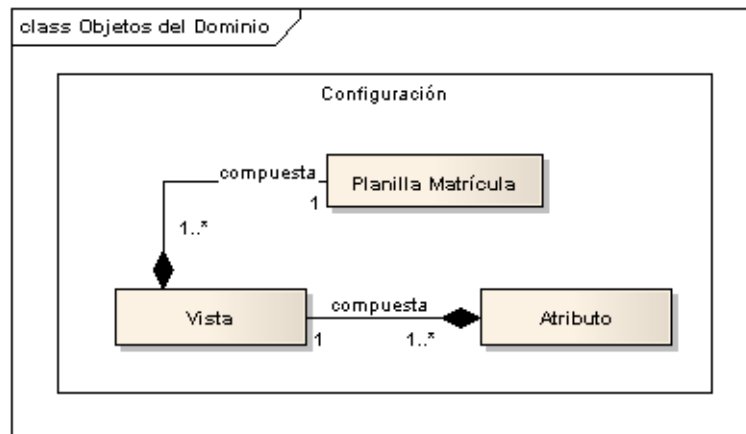


Fig. 2.5 Modelo del dominio del paquete "Configuración"

La imagen anterior, muestra las clases que se manejan en el paquete de Configuración, como son: planilla matrícula, vista y atributo. La planilla matrícula está compuesta por varias vistas, que son definidas previamente en dependencia de las necesidades de captura de información del centro al cual será asignada la misma, y las vistas a su vez están compuestas por los datos que se desean capturar del estudiante, los cuales se consideran como atributos.

2.5 Especificación de los Requerimientos de Software

Mediante una descripción detallada de los requisitos del sistema se logra una mayor comunicación y entendimiento con el cliente y con los desarrolladores sobre lo que el sistema debe o no hacer. Los requisitos son capacidades o propiedades que debe cumplir o tener un sistema para satisfacer un contrato entre los interesados, con un final denominado producto.

2.5.1 Requerimientos Funcionales

Capítulo 2: Características del Sistema

Las acciones que el sistema debe permitir realizar para cumplir con las metas trazadas inicialmente, se definen a continuación y están organizadas en los paquetes Registro y Configuración.

Paquete Registro

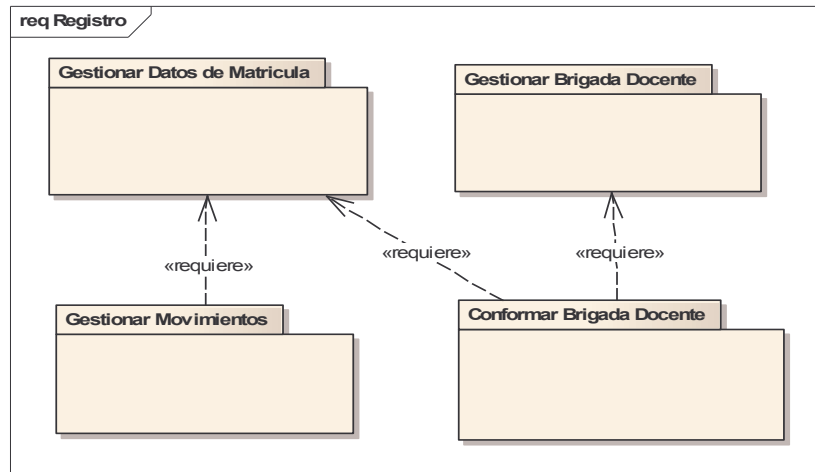


Fig. 2.6 Requerimientos funcionales del paquete "Registro"

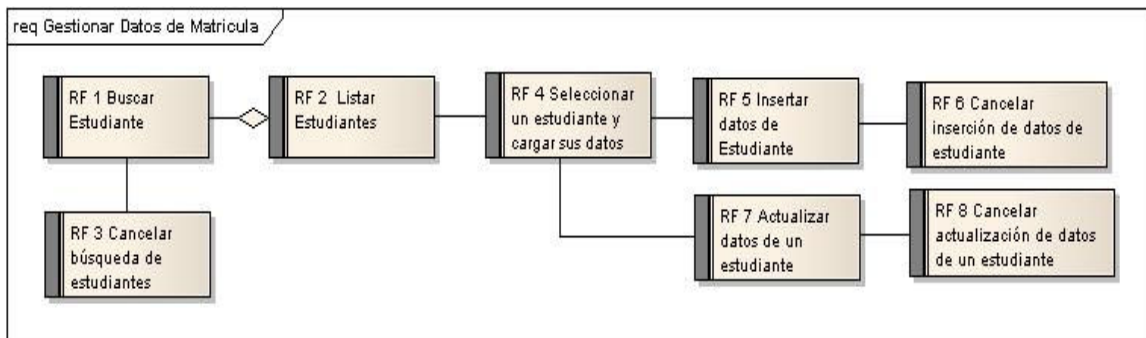


Fig. 2.7 Requerimientos funcionales del paquete "Gestionar Datos de Matrícula"

Capítulo 2: Características del Sistema

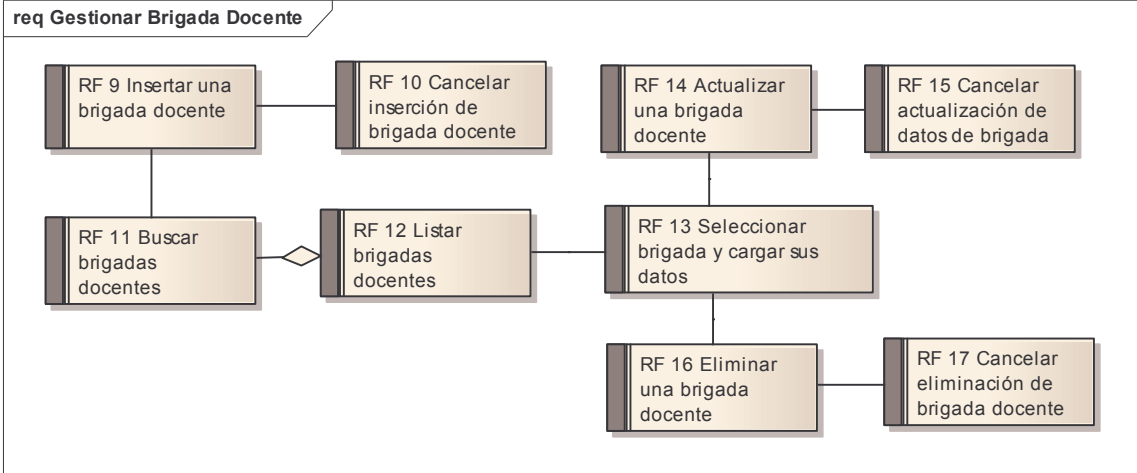


Fig. 2.8 Requerimientos funcionales del paquete "Gestionar Brigada Docente"

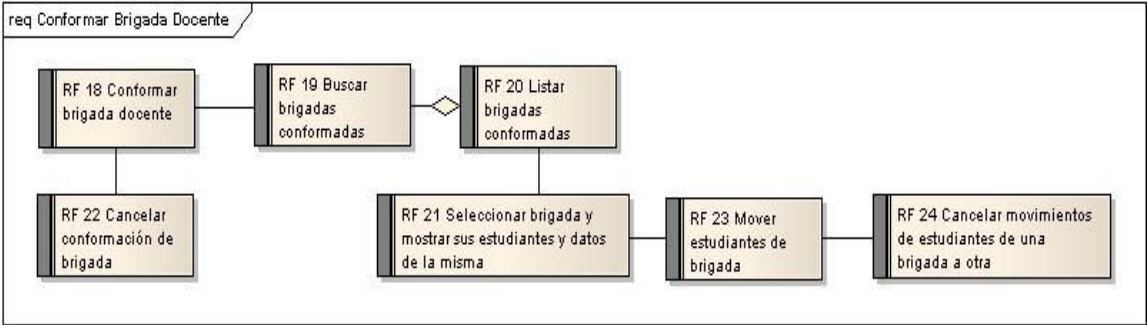


Fig. 2.9 Requerimientos funcionales del paquete "Conformar Brigada Docente"

Capítulo 2: Características del Sistema

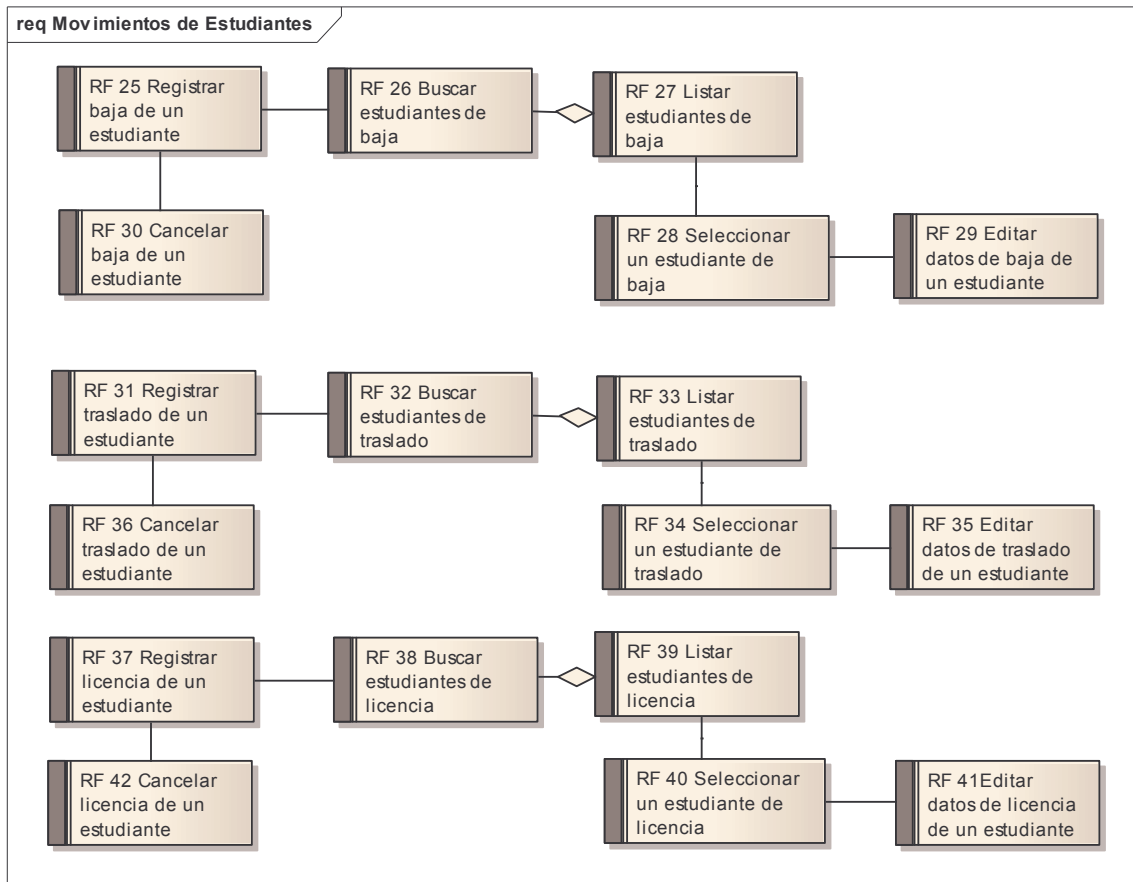


Fig. 2.10 Requerimientos funcionales del paquete "Gestionar Movimientos"

Paquete de Configuración

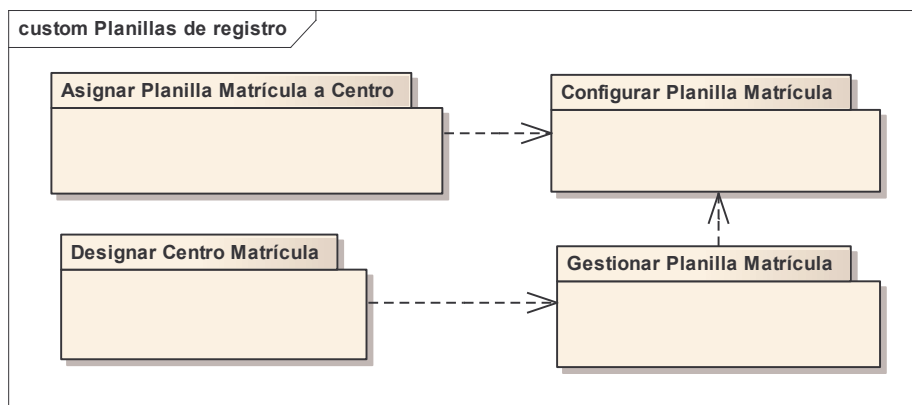


Fig. 2.11 Requerimientos funcionales del paquete "Configuración"

Capítulo 2: Características del Sistema

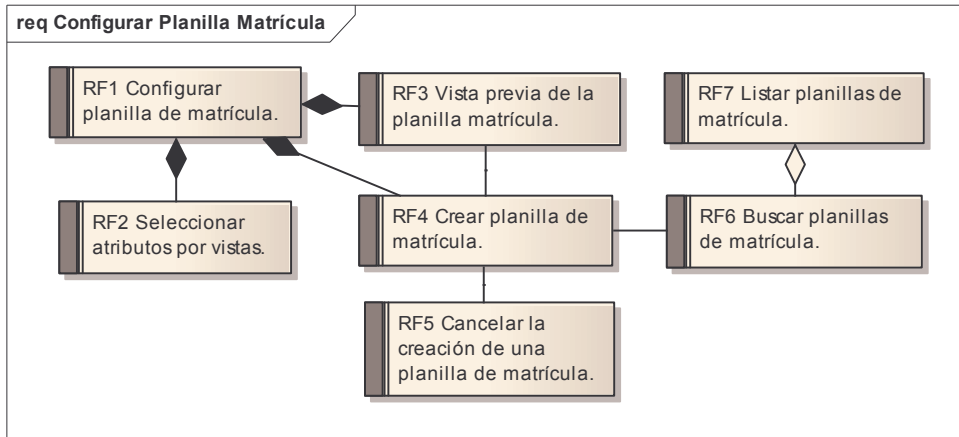


Fig. 2.12 Requerimientos funcionales del paquete "Configurar Planilla Matrícula"

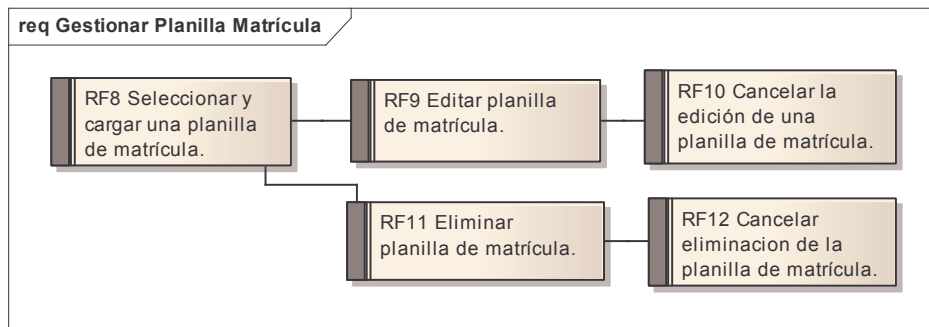


Fig. 2.13 Requerimientos funcionales del paquete "Gestionar Planilla Matrícula"

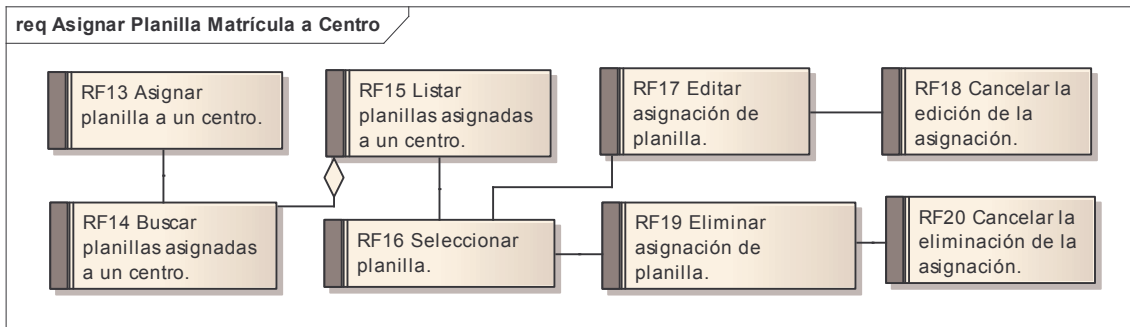


Fig. 2.14 Requerimientos funcionales del paquete "Asignar Planilla Matrícula a Centro"

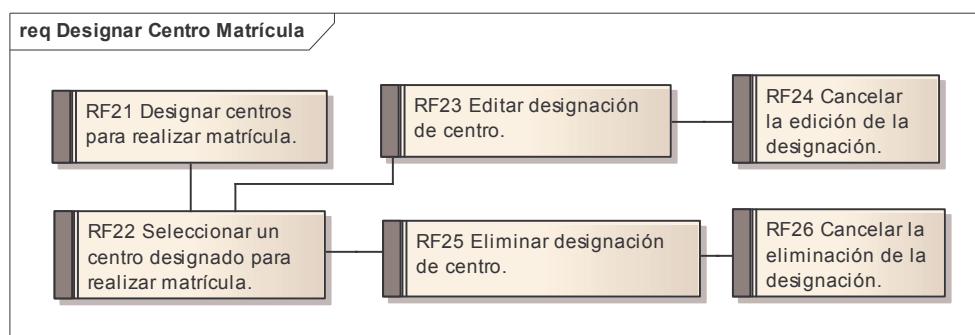


Fig. 2.15 Requerimientos funcionales del paquete “Designar Centro Matrícula”

2.5.2 Requerimientos No Funcionales

Para hacer más atractivo al producto y que sea usado por su calidad, usabilidad, seguridad y fácil entendimiento, el sistema debe tener una serie de cualidades, que son presentadas a continuación.

2.5.2.1 Requisitos de Usabilidad

RNF1 El sistema sólo podrá ser accedido por los usuarios definidos. Cada vez que se autenticuen tendrán los derechos necesarios de acceso según los privilegios del rol que desempeñen.

RNF2 El sistema debe garantizar un acceso fácil y rápido, podrá ser usado por usuarios con conocimientos informáticos básicos (manejo de la computadora, y específicamente de un navegador Web).

RNF3 Las funcionalidades estarán visibles en la parte superior del sistema haciendo más fácil la distribución de las mismas y la localización de estas por el usuario.

2.5.2.2 Requisitos de Fiabilidad

RNF4 Se garantiza el acceso controlado a la información. Esto influye en la forma en que se presentarán las interfaces a cada usuario en dependencia del nivel de acceso que le será conferido.

2.5.2.3 Requisitos de Confidencialidad

Capítulo 2: Características del Sistema

RNF5 La información estará protegida contra accesos no autorizados utilizando mecanismos de autenticación y autorización, lográndose con la utilización de servicios Web que brinda el Sistema de Autenticación, Autorización y Auditoría.

RNF6 La autenticación será la primera acción del usuario en el sistema y consistirá en suministrar un nombre de usuario único y una contraseña que debe ser de conocimiento exclusivo de la persona que se autentica. Si el usuario autenticado no se encuentra registrado se debe reportar un error de acceso.

2.5.2.4 Requisitos de Integridad

RNF7 La información podrá ser modificada solo por personal autorizado.

RNF8 La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, mediante la validación de los datos de entrada.

RNF9 Se implementarán políticas de resguardo de información, así como la realización de copias periódicas de seguridad, que puedan restaurar el sistema en caso de fallo crítico o pérdida total de la información.

RNF10 Se harán validaciones de la información tanto en el cliente como en el servidor contra ataques de inyección HTML o SQL.

2.5.2.5 Requisitos de Disponibilidad

RNF11 La seguridad no implicará lentitud o retraso en la respuesta dada por el sistema, por lo que se debe minimizar y reducir el tiempo de respuesta, así como optimizar el código.

2.5.2.6 Requisitos de Eficiencia

RNF12 El sistema deberá ser rápido ante las solicitudes de los usuarios en el procesamiento de la información, el tiempo de respuesta deberá ser el mínimo posible.

2.5.2.7 Requisitos de Soporte

RNF13 Para una mayor organización del trabajo y facilitar el soporte y mantenimiento del sistema, se establecerán estándares para la codificación, pautas para la arquitectura de

Capítulo 2: Características del Sistema

información de la interfaz de usuario, así como para la documentación generada como parte del proceso de desarrollo.

RNF14 Una vez terminado el sistema, se realizarán procesos de despliegue, capacitación y mantenimiento de software.

RNF15 La capacitación se realizará a través de cursos al personal designado como usuarios finales, según los tipos de usuarios definidos para el sistema.

RNF16 El servicio de soporte técnico comenzará una vez instalada la solución. Este servicio proporcionará atención, seguimiento y solución a requerimientos técnicos reportados por los clientes, relacionados con el funcionamiento del sistema informático.

2.5.2.8 Restricciones de Diseño

RNF17 Se definen pautas para la arquitectura de información, de manera que el diseño del sistema sea homogéneo en todas sus interfaces de usuario.

RNF18 Se definen además, estándares de diseño y codificación por los cuales debe regirse tanto la implementación como los modelos generados como parte del proceso de desarrollo.

2.5.2.9 Requisitos Documentación y Ayuda

RNF19 El sistema facilitará ayuda en línea sobre el funcionamiento del mismo, manuales de usuario y tutoriales dinámicos que además podrán ser utilizados para las actividades de capacitación.

RNF20 En la ejecución de las funcionalidades del sistema, se emitirán mensajes que pueden ser informativos, de error o de advertencia. Los informativos se refieren a la respuesta del sistema al realizar alguna operación; las advertencias se mostrarán antes de realizar alguna acción que afecte otros procesos o suponga una modificación de la información registrada. Por otro lado los mensajes de error se emitirán cuando no se introduzca información obligatoria, o esta sea incorrecta.

RNF21 Se pautarán los mensajes del sistema, de manera que se garantice una presentación homogénea de sus textos.

Capítulo 2: Características del Sistema

RNF22 Se dispondrá de la documentación del desarrollo del sistema, realizada siguiendo la metodología de desarrollo RUP.

2.5.2.10 Requisitos Interfaz de Usuario

RNF23 El sistema debe tener una interfaz sencilla, agradable, legible y de fácil uso para el usuario. Su funcionamiento deberá ser intuitivo, y requerir de información mínima.

RNF24 Debe tener claridad y buena organización de la información, permitiendo la interpretación correcta e inequívoca de la misma.

RNF25 Debe permitir la ejecución de acciones de manera rápida.

RNF26 Se debe hacer uso de elementos visuales para la selección de información siempre que sea posible para minimizar los posibles errores.

RNF27 Todos los textos y mensajes en pantalla aparecerán en idioma español.

RNF28 Los errores serán visibles al usuario e incluirán sugerencias de las posibles soluciones.

RNF29 Se definen pautas, y aplicación de normas de diseño que garanticen armonía visual entre los elementos de la interfaz.

2.5.2.11 Requisitos de Interfaz de Hardware

Para nodo procesador servidor de aplicación y servidor de base de datos:

RNF30 Tipo de procesador: Intel Pentium IV o superior.

RNF31 Velocidad del procesador: 3.00 GHz.

RNF32 Memoria RAM: 1 GB.

RNF33 Disco Duro: 80 GB para servidor de aplicación y 160 GB para el de base de datos.

RNF34 Se requiere tarjeta de red.

Para nodo procesador cliente o estaciones de trabajo:

Capítulo 2: Características del Sistema

RNF35 Tipo de procesador: Intel Pentium III o superior.

RNF36 Velocidad del procesador: 512 MHz o superior.

RNF37 Memoria RAM: 256 MB o superior.

RNF38 Disco Duro: 1 GB.

RNF39 Se requiere tarjeta de red.

2.5.2.12 Requisitos de Interfaz de Software

Para nodo procesador cliente o estaciones de trabajo:

RNF40 Navegador: Internet Explorer 6.0 o superior y Mozilla Firefox 2.0 o superior, con los componentes JavaScript habilitados.

RNF41 Sistema Operativo: Windows XP Service Pack 2 o GNU/Linux distribución Ubuntu 8.04.

Para nodo procesador servidor de aplicación y servidor de base de datos:

RNF42 Sistema Operativo: GNU/Linux Distribución Debian 4 Etch.

RNF43 Servidor web: Apache 2.2.2.

RNF44 Framework Symfony 1.2 en el servidor de aplicación.

RNF45 PHP 5.0 en el servidor de aplicación.

RNF46 Sistema gestor de base de datos: PostgreSQL 8.3 en el servidor de base de datos.

2.6 Modelo de Casos de Uso del Sistema

El modelo de casos de uso del sistema es un artefacto que describe, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Este modelo está formado por actores, casos de usos y las relaciones que se establecen entre estos y constituye una entrada para el análisis, el diseño y las pruebas.

Capítulo 2: Características del Sistema

2.6.1 Definición de los Actores del Sistema

Los actores representan terceros fuera del sistema que colaboran con el mismo. A continuación se describen las definiciones de los actores del sistema propuesto:

ACTORES	JUSTIFICACIÓN
Editor Sede.	Sólo tiene acceso a la información de su centro. Es el responsable de ingresar nuevos estudiantes al sistema. Gestiona las brigadas docentes que se crean en el centro de estudios y la gestión de los movimientos (bajas, licencias y traslados).
Usuario Avanzado	Es el encargado de la configuración de la planilla de matrícula. Designa los centros que realizarán matrícula y además asigna las planillas a los distintos centros.

Tabla 2.1 Definición de los actores del sistema

2.6.2 Definición de los Casos de Uso del Sistema

DEFINICIÓN DE CASOS DE USO	
CUS-1 Buscar un estudiante.	CUS-5 Editar brigada docente.
CUS-2 Insertar Datos de estudiante.	CUS-6 Eliminar brigada docente.
CUS-3 Actualizar Datos de estudiante.	CUS-7 Conformar brigada docente.
CUS-4 Crear brigada docente.	CUS-8 Mover estudiante de brigada docente.
	CUS-9 Gestionar Movimientos.

Tabla 2.2 Definición de casos de uso del paquete "Registro"

DEFINICIÓN DE CASOS DE USO	
CUS-1 Configurar Planilla Matrícula.	CUS-5 Eliminar Centro Matrícula.
CUS-2 Editar Planilla Matrícula.	CUS-6 Asignar Planilla Matrícula a Centro.
CUS-3 Eliminar Planilla Matrícula.	CUS-7 Actualizar Planilla Matrícula a Centro.
CUS-4 Designar Centro Matrícula.	CUS-8 Eliminar Planilla Matrícula a Centro.

Tabla 2.3 Definición de casos de uso del paquete "Configuración"

2.6.3 Diagramas de Casos de Uso del Sistema

Capítulo 2: Características del Sistema

Un diagrama de casos de uso del sistema es una representación gráfica de los actores y casos de uso del sistema, incluyendo sus interacciones. Se muestran, por tanto, los distintos requerimientos funcionales que se esperan del sistema y cómo se relacionan con su entorno.

El siguiente diagrama muestra los casos de uso del MRE divididos en paquetes: Registro y Configuración.

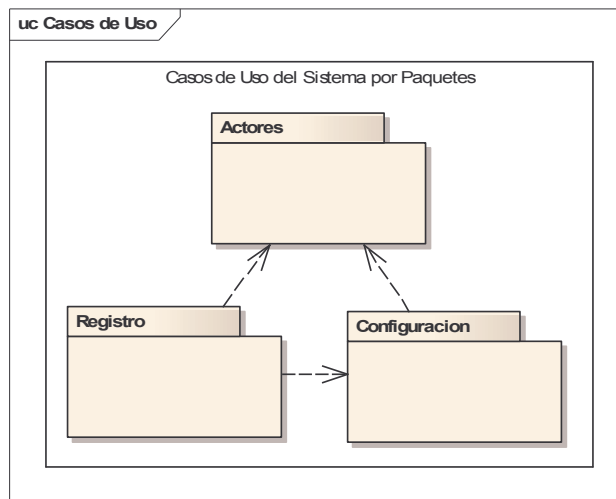


Fig. 2.6 Diagrama de casos de uso del sistema por paquetes.

El Registro se concibe igual tanto para la matrícula de Pregrado como para Posgrado aunque en este último no se concibe la gestión de las brigadas docentes ni la conformación de las mismas, como se puede apreciar en el siguiente diagrama.

Capítulo 2: Características del Sistema

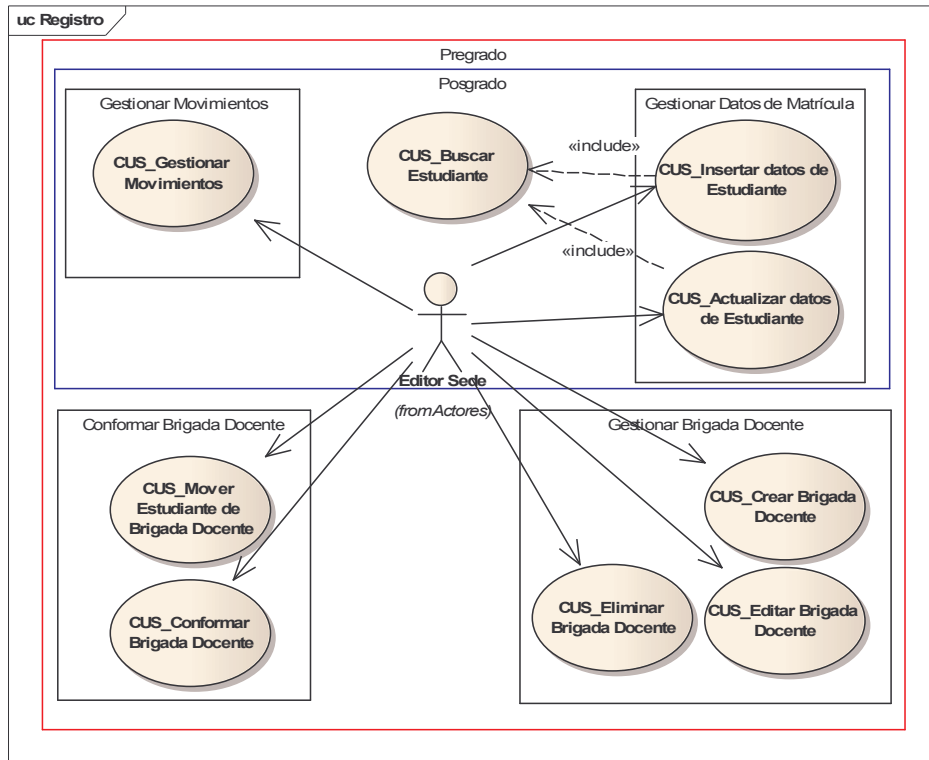


Fig. 2.7 Diagrama de CUS del paquete "Registro"

La Configuración, también se concibe igual tanto para la matrícula de Pregrado como para Posgrado, solo se debe especificar el tipo de matrícula cuando se vayan a designar los centros, se configuren las planillas y se asignen estas a cada uno de los centros designados.

Capítulo 2: Características del Sistema

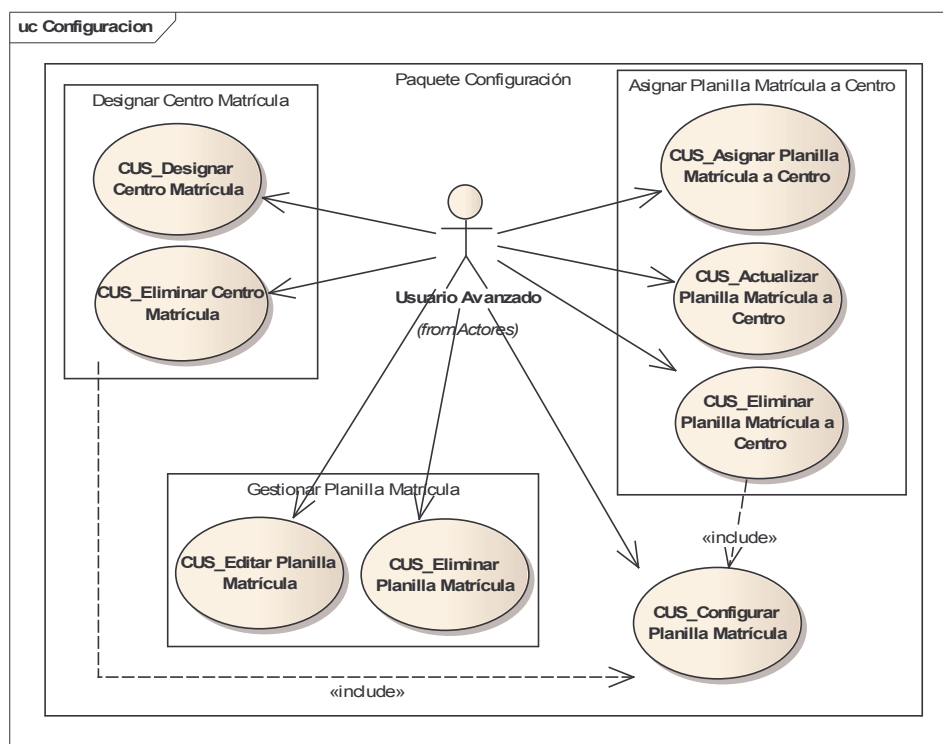


Fig. 2.8 Diagrama de CUS del Paquete Configuración

2.6.4 Descripción textual de los Casos de Uso del Sistema

La descripción de los casos de uso identificados, mediante los cuales se satisfacen los requerimientos funcionales del sistema en desarrollo se muestran en las siguientes tablas. Primeramente se representan los casos de usos del sistema del paquete Registro seguidos de los casos de usos del sistema del paquete Configuración:

CASO DE USO:	Buscar Estudiante
Resumen:	El caso de uso inicia cuando el actor accede a la opción Buscar Estudiante para poder insertar, actualizar o eliminar los datos de un Estudiante. El sistema brinda la posibilidad de introducir criterios de búsqueda para localizar al Estudiante, el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra los Estudiantes que cumplen con los criterios de búsqueda, el caso de uso termina.
Complejidad:	Alta
Prioridad:	Crítico
Precondiciones:	Editor Sede autenticado satisfactoriamente por el sistema.
REFERENCIAS	
Actores:	Editor Sede
Requisitos:	1
Entidades:	Estudiante

Capítulo 2: Características del Sistema

Casos de Uso:	Insertar Datos de un Estudiante Actualizar Datos de un Estudiante
FLUJO NORMAL DE EVENTOS	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor accede a la opción Buscar Estudiante	
	2. Brinda la posibilidad de introducir los criterios elementales de búsqueda: <ul style="list-style-type: none"> • Nombre • Apellidos • CI/DNI • Sexo y permite: <ul style="list-style-type: none"> • Buscar Estudiante dado criterios. • Cancelar operación. Ver Alternativa 2: “Cancelar operación.”
3. Introduce los datos que considera como criterios para realizar una búsqueda y selecciona la opción de Buscar Estudiante dado criterios.	
	4. Busca los datos de Estudiante que cumplen con los criterios de búsqueda.
	5. Si no se encuentra ningún Estudiante que cumpla con los criterios de búsqueda. Ver Alternativa 1: “No se encuentra información que cumpla con los criterios de búsqueda.”

Capítulo 2: Características del Sistema

	<p>6. Muestra un listado de Estudiantes que cumplen con los criterios de búsqueda, mostrando los siguientes atributos:</p> <ul style="list-style-type: none"> • Nombre • Apellidos • CI/DNI • Sexo <p>Ordenados ascendentemente por el Nombre, mostrando la cantidad de elementos configurados para mostrar por página, permitiendo navegar por el resultado.</p>
	7. El caso de uso termina.
Prototipo de Interfaz (Ver anexo 2)	
FLUJOS ALTERNOS	
Alternativa 1: “No se encuentra información que cumpla con los criterios de búsqueda.”	
Acción del Actor	Respuesta del Sistema
	1. Muestra el mensaje de información “No se encontró información que cumpla con los criterios de búsqueda.”
	2. Regresa al paso 2 del Flujo Normal de Eventos.
Alternativa 2. “Cancelar operación.”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción de Cancelar operación.	
	2. Regresa a la vista anterior.
	3. El caso de uso termina.
Poscondiciones	Se buscó el Estudiante dado criterios.

Tabla 2.4 Descripción del CU “Buscar Estudiante”

CASO DE USO:	Insertar datos de Estudiante
Resumen:	El caso de uso inicia cuando el actor accede a la opción Insertar Estudiante, el sistema brinda la posibilidad de introducir los datos para insertar el Estudiante, el actor introduce los datos del Estudiante, el sistema inserta el Estudiante, el caso de uso termina.
Complejidad:	Alta

Capítulo 2: Características del Sistema

Prioridad:	Crítico
Precondiciones:	Editor Sede autenticado satisfactoriamente por el sistema.
REFERENCIAS	
Actores:	Editor Sede
Requisitos:	1,2,3,4,5,6
Entidades:	Estudiante
Casos de Uso:	Buscar Estudiante
FLUJO NORMAL DE EVENTOS	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor accede a la opción Insertar Estudiante.	
	<p>2. Muestra los datos predeterminados:</p> <ul style="list-style-type: none"> • Nombre • Apellidos • CI/DNI • Sexo <p>Brinda la posibilidad de introducir los restantes datos de Estudiante en dependencia de la configuración de la Planilla Matrícula, y permite:</p> <ul style="list-style-type: none"> • Aceptar insertar Estudiante. • Cancelar operación. Ver Alternativa 1: "Cancelar operación."
<p>3. Introduce los datos de Estudiante:</p> <ul style="list-style-type: none"> • Nombre • Apellidos • CI/DNI • Sexo <p>Introduce los restantes datos de Estudiante en dependencia de la configuración de la Planilla Matrícula.</p>	
4. Selecciona la opción de aceptar insertar Estudiante.	

Capítulo 2: Características del Sistema

	5. Valida los datos. Si hay datos incompletos, ver Alternativa 2 : “Existen datos incompletos.”. Si hay datos incorrectos, ver Alternativa 3 : “Existen datos incorrectos.”
	6. <i>Inserta Estudiante.</i>
	7. Muestra el mensaje de información “Se ha creado un nuevo Estudiante.” Y permite: <ul style="list-style-type: none"> • Crear un nuevo Estudiante. Ver Alternativa 4. “Crear un nuevo Estudiante.” • Cancelar operación. Ver Alternativa 1: “Cancelar operación.”
	8. El caso de uso termina.
Prototipo de Interfaz (Ver anexo 3)	
FLUJOS ALTERNOS	
Alternativa 1. “Cancelar operación.”	
Acción del Actor	Respuesta del Sistema
4. Selecciona la opción de Cancelar operación.	
	5. Regresa a la vista anterior.
	6. El caso de uso termina.
Alternativa 2. “Existen datos incompletos.”	
Acción del Actor	Respuesta del Sistema
	1. Muestra el mensaje de error “Existen campos vacíos que son obligatorios, por favor, complete estos datos.”
	2. Muestra un indicador sobre los campos vacíos.
Alternativa 3. “Existen datos incorrectos.”	
Acción del Actor	Respuesta del Sistema

Capítulo 2: Características del Sistema

	1. Muestra el mensaje de error “Existen campos escritos incorrectamente, por favor, rectifique estos datos.”
	2. Muestra un indicador sobre los campos incorrectos.
Alternativa 4. “Insertar un nuevo Estudiante.”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción de Insertar un nuevo Estudiante.	
	2. Regresa al paso 2 del Flujo Normal de Eventos.
Poscondiciones	Se insertó un Estudiante.

Tabla 2.5 Descripción del CU “Insertar datos de Estudiante”

CASO DE USO:	Gestionar Movimientos
Resumen:	El caso de uso inicia cuando el actor accede a la opción Movimientos, el sistema brinda la posibilidad de seleccionar los datos para asignar baja, licencia o traslado al Estudiante, el actor introduce los datos del Estudiante, el sistema actualiza la Brigada Docente, el caso de uso termina.
Complejidad:	Alta
Prioridad:	Crítico
Precondiciones:	Editor Sede autenticado satisfactoriamente por el sistema.
REFERENCIAS	
Actores:	Editor Sede
Requisitos:	25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42
Entidades:	Estudiante, Brigada Docente
Casos de Uso:	Buscar Estudiante
FLUJO NORMAL DE EVENTOS	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor accede a la opción Movimientos.	
	2. Se ejecuta el CU “Buscar Estudiante”.
3. Selecciona el Estudiante.	

Capítulo 2: Características del Sistema

	<p>4. Muestra el Nombre y Apellidos del Estudiante seleccionado y brinda la posibilidad de elegir el tipo de Movimiento:</p> <ul style="list-style-type: none"> • Baja • Licencia • Traslado <p>y permite:</p> <ul style="list-style-type: none"> • Aceptar Baja de Estudiante. Ver Sección 1:“Baja de Estudiante.” • Aceptar Licencia de Estudiante. Ver Sección 2:“Licencia de Estudiante.” • Aceptar Traslado de Estudiante. Ver Sección 3:“Traslado de Estudiante.” • Cancelar operación. Ver Alternativa 1: “Cancelar operación.”
Sección 1. “Baja de Estudiante.”	
Acción del Actor	Respuesta del Sistema
5. Selecciona la opción de aceptar baja de Estudiante.	
	<p>6. Muestra los datos predeterminados de la Baja de Estudiante:</p> <ul style="list-style-type: none"> • Fecha • Causas • Observaciones
7. Introduce los datos predeterminados: <ul style="list-style-type: none"> • Fecha • Causas • Observaciones 	
Sección 2. “Licencia de Estudiante.”	
Acción del Actor	Respuesta del Sistema
8. Selecciona la opción de aceptar licencia de Estudiante.	

Capítulo 2: Características del Sistema

	<p>9. Muestra los datos predeterminados de la Licencia de Estudiante:</p> <ul style="list-style-type: none"> • Tiempo • Causas • Observaciones • Fecha probable de reincorporación • Fecha real reincorporación
<p>10. Introduce los datos predeterminados:</p> <ul style="list-style-type: none"> • Tiempo • Causas • Observaciones • Fecha probable de reincorporación • Fecha real reincorporación 	
Sección 3. "Traslado de Estudiante."	
Acción del Actor	Respuesta del Sistema
<p>11. Selecciona la opción de aceptar traslado de Estudiante.</p>	
	<p>12. Muestra los datos predeterminados de la Traslado de Estudiante:</p> <ul style="list-style-type: none"> • Provincia • Centro • Causas • Observaciones
<p>13. Introduce los datos predeterminados:</p> <ul style="list-style-type: none"> • Provincia • Centro • Causas • Observaciones 	
	<p>14. Valida los datos. Si hay datos incompletos, ver Alternativa 2: "Existen datos incompletos.". Si hay datos incorrectos, ver Alternativa 3: "Existen datos incorrectos."</p>
	<p>15. <i>Gestionar Movimientos.</i></p>

Capítulo 2: Características del Sistema

	<p>16. Muestra el mensaje de información “Se le ha asignado la Baja al Estudiante.”</p> <p>17. Muestra el mensaje de información “Se le ha asignado la Licencia al Estudiante.”</p> <p>18. Muestra el mensaje de información “Se le ha asignado el Traslado al Estudiante.”</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Nuevo movimiento de Estudiante. Ver Alternativa 4. “Nuevo movimiento de Estudiante.” • Cancelar operación. Ver Alternativa 1: “Cancelar operación.”
	19. El caso de uso termina.
Prototipo de Interfaz (Ver anexo 4)	
FLUJOS ALTERNOS	
Alternativa 1. “Cancelar operación.”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción de Cancelar operación.	
	2. Regresa a la vista anterior.
	3. El caso de uso termina.
Alternativa 2. “Existen datos incompletos.”	
Acción del Actor	Respuesta del Sistema
	3. Muestra el mensaje de error “Existen campos vacíos que son obligatorios, por favor, complete estos datos.”
	4. Muestra un indicador sobre los campos vacíos.
Alternativa 3. “Existen datos incorrectos.”	
Acción del Actor	Respuesta del Sistema
	3. Muestra el mensaje de error “Existen campos escritos incorrectamente, por favor, rectifique estos datos.”

Capítulo 2: Características del Sistema

	4. Muestra un indicador sobre los campos incorrectos.
Alternativa 4. “Nuevo movimiento de Estudiante.”	
Acción del Actor	Respuesta del Sistema
3. Selecciona la opción de Movimientos de Estudiante.	
	4. Regresa al paso 2 del Flujo Normal de Eventos.
Poscondiciones	Se le asignó baja, licencia o traslado a un Estudiante.

Tabla 2.6 Descripción del CU “Gestionar Movimientos”

A continuación se describen los casos de usos del paquete Configuración.

CASO DE USO:	Configurar Planilla Matrícula
Resumen:	El caso de uso inicia cuando el actor accede a la opción Configurar Planilla Matrícula, el sistema brinda la posibilidad de introducir los datos para crear la Planilla Matrícula, el actor introduce los datos de la Planilla Matrícula, el sistema crea la Planilla Matrícula, el caso de uso termina.
Complejidad:	Alta
Prioridad:	Crítico
Precondiciones:	El usuario avanzado tiene que estar autenticado y los atributos necesarios para la configuración deben estar creados.
REFERENCIAS	
Actores:	Usuario Avanzado
Requisitos:	1,2,3,4,5,6,7
Entidades:	Planilla Matrícula
Casos de Uso:	Asignar Planilla Matrícula a Centro Actualizar Planilla Matrícula a Centro Eliminar Planilla Matrícula a Centro Designar Centro Matrícula Actualizar Centro Matrícula Eliminar Centro Matrícula
FLUJO NORMAL DE EVENTOS	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor accede a la opción Configurar Planilla Matrícula.	

Capítulo 2: Características del Sistema

	<p>2. Muestra las vistas predeterminadas en dependencia del centro.</p> <p>Brinda la posibilidad de introducir los datos de la Planilla Matrícula también en dependencia de las vistas ya establecidas y de las necesidades del centro. Y permite:</p> <ul style="list-style-type: none"> • Aceptar configurar Planilla Matrícula. • Cancelar operación. Ver Alternativa 1: “Cancelar operación.”
3. Introduce los datos predeterminados en la configuración de la Planilla Matrícula.	
4. Selecciona la opción de aceptar configurar Planilla Matrícula.	
	<p>5. Valida los datos. Si hay datos incompletos, ver Alternativa 2: “Existen datos incompletos.”. Si hay datos incorrectos, ver Alternativa 3: “Existen datos incorrectos.”</p>
	6. <i>Configurar Planilla Matrícula</i>
	<p>7. Muestra el mensaje de información “Se ha creado la Planilla Matrícula.”</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Configurar una nueva Planilla Matrícula. Ver Alternativa 4. “Configurar una nueva Planilla Matrícula.” • Cancelar operación. Ver Alternativa 1: “Cancelar operación.”
	8. El caso de uso termina.
Prototipo de Interfaz (Ver Anexo 5)	
FLUJOS ALTERNOS	
Alternativa 1. “Cancelar operación.”	
Acción del Actor	Respuesta del Sistema

Capítulo 2: Características del Sistema

7. Selecciona la opción de Cancelar operación.	
	8. Regresa a la vista anterior.
	9. El caso de uso termina.
Alternativa 2. "Existen datos incompletos."	
Acción del Actor	Respuesta del Sistema
	5. Muestra el mensaje de error "Existen campos vacíos que son obligatorios, por favor, complete estos datos."
	6. Muestra un indicador sobre los campos vacíos.
Alternativa 3. "Existen datos incorrectos."	
Acción del Actor	Respuesta del Sistema
	5. Muestra el mensaje de error "Existen campos escritos incorrectamente, por favor, rectifique estos datos."
	6. Muestra un indicador sobre los campos incorrectos.
Alternativa 4. "Configurar una nueva Planilla Matrícula."	
Acción del Actor	Respuesta del Sistema
5. Selecciona la opción de Configurar una nueva Planilla Matrícula.	
	6. Regresa al paso 2 del Flujo Normal de Eventos.
Poscondiciones	Se creó una Planilla Matrícula.

Tabla 2.7 Descripción del CU "Configurar Planilla Matrícula"

CASO DE USO:	Designar Centro Matrícula
Resumen:	El caso de uso inicia cuando el actor accede a la opción Designar Centro Matrícula, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar el Centro Docente deseado, el actor introduce los datos que considera como criterios para buscar el Centro Docente, el sistema busca y muestra los Centros Docentes que cumplen con los criterios de búsqueda, el actor selecciona el Centro Docente deseado, el sistema carga en la vista anterior cada uno de los Centros Docentes seleccionados, el caso de uso termina.
Complejidad:	Alta
Prioridad:	Crítico

Capítulo 2: Características del Sistema

Precondiciones:	El usuario avanzado debe autenticarse satisfactoriamente en el sistema.
REFERENCIAS	
Actores:	Usuario Avanzado
Requisitos:	20,21,22,23,24,25
Entidades:	Centro Docente
Casos de Uso:	Configurar Planilla Matrícula
FLUJO NORMAL DE EVENTOS	
Acción del Actor	Respuesta del Sistema
20. El caso de uso inicia cuando el actor accede a la opción Designar Centro Matrícula.	
	<p>21. Brinda la posibilidad de introducir los criterios elementales de búsqueda:</p> <ul style="list-style-type: none"> • Provincia • Municipio • Curso Docente <p>y permite:</p> <ul style="list-style-type: none"> • Buscar Centros Docentes dado criterios. • Cancelar operación. Ver Alternativa 2: “Cancelar operación.”
22. Introduce los datos que considera como criterios para realizar una búsqueda de los Centros Docentes.	
	23. <i>Busca los datos de Centro Docente que cumplen con los criterios de búsqueda.</i>
	24. Si no se encuentra ningún Centro Docente que cumpla con los criterios de búsqueda. Ver Alternativa 1: “No se encuentra información que cumpla con los criterios de búsqueda.”

Capítulo 2: Características del Sistema

	<p>25. Muestra un listado de Centro Docente que cumplen con los criterios de búsqueda, mostrando los siguientes atributos:</p> <ul style="list-style-type: none"> • Nombre del Centro • Curso Docente • Provincia <p>Ordenados ascendentemente por el Nombre del Centro, mostrando la cantidad de elementos configurados para mostrar por página, permitiendo navegar por el resultado.</p>
	<p>26. Permite:</p> <ul style="list-style-type: none"> • Seleccionar uno o varios Centros Docentes.
27. Selecciona los Centros Docentes deseados.	
	<p>28. De cada una de los Centros Docentes seleccionados, carga en la vista anterior los datos:</p> <ul style="list-style-type: none"> • Nombre del Centro • Curso Docente • Provincia
	29. El caso de uso termina.
Prototipo de Interfaz (Ver Anexo 6)	
FLUJOS ALTERNOS	
Alternativa 1: “No se encuentra información que cumpla con los criterios de búsqueda.”	
Acción del Actor	Respuesta del Sistema
	3. Muestra el mensaje de información “No se encontró información que cumpla con los criterios de búsqueda.”
	4. Regresa al paso 2 del Flujo Normal de Eventos.
Alternativa 2. “Cancelar operación.”	
Acción del Actor	Respuesta del Sistema

Capítulo 2: Características del Sistema

1. Selecciona la opción de Cancelar operación.	
	2. Regresa a la vista anterior.
	3. El caso de uso termina.
Poscondiciones	Se designó un Centro Docente dado criterios.

Tabla 2.8 Descripción del CU “Designar Centro Matrícula”

CASO DE USO:	Asignar Planilla Matrícula a Centro
Resumen:	El caso de uso inicia cuando el actor accede a la opción Asignar Planilla Matrícula a Centro, el sistema brinda la posibilidad de seleccionar el Centro Docente y la Planilla Matrícula deseados, el actor selecciona el Centro Docente deseado, el sistema carga las Planillas Matrículas previamente creadas y el actor selecciona la Planilla Matrícula del Centro Docente, el caso de uso termina.
Complejidad:	Alta
Prioridad:	Crítico
Precondiciones:	El usuario avanzado debe autenticarse satisfactoriamente en el sistema. Configurada la Planilla Matrícula.
REFERENCIAS	
Actores:	Usuario Avanzado
Requisitos:	13,14,15,16,17,18,19
Entidades:	Centro Docente, Planilla Matrícula
Casos de Uso:	Configurar Planilla Matrícula
FLUJO NORMAL DE EVENTOS	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor accede a la opción Asignar Planilla Matrícula a Centro.	
	2. Brinda la posibilidad de seleccionar los criterios: <ul style="list-style-type: none"> • Planilla Matrícula • Centro Docente y permite: <ul style="list-style-type: none"> • Aceptar la asignación. • Cancelar operación. Ver Alternativa 3: “Cancelar operación.”

Capítulo 2: Características del Sistema

<p>3. Selecciona los datos:</p> <ul style="list-style-type: none"> • Planilla Matrícula • Centro Docente 	
<p>4. Selecciona la opción de aceptar asignar Planilla Matrícula a Centro.</p>	
	<p>5. Valida los datos. Si hay datos incompletos, ver Alternativa 1: “Existen datos incompletos.”. Si hay datos incorrectos, ver Alternativa 2: “Existen datos incorrectos.”</p>
	<p>6. Muestra un listado de Centro Docente con Planilla Matrícula asignada, mostrando los siguientes atributos:</p> <ul style="list-style-type: none"> • Nombre del Centro • Planilla Matrícula <p>Ordenados ascendentemente por el Nombre del Centro, mostrando la cantidad de elementos configurados para mostrar por página, permitiendo navegar por el resultado.</p>
	<p>7. <i>Asignar Planilla Matrícula a Centro.</i></p>
	<p>8. Muestra el mensaje de información “Se ha asignado la Planilla Matrícula al Centro Docente.”</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Asignar una nueva Planilla Matrícula a Centro. Ver Alternativa 4: “Asignar una nueva Planilla Matrícula a Centro.” • Cancelar operación. Ver Alternativa 3: “Cancelar operación.”
	<p>9. El caso de uso termina.</p>
<p>Prototipo de Interfaz (Ver Anexo 7)</p>	
<p>FLUJOS ALTERNOS</p>	

Capítulo 2: Características del Sistema

Alternativa 1. “Existen datos incompletos.”	
Acción del Actor	Respuesta del Sistema
	1. Muestra el mensaje de error “Existen campos vacíos que son obligatorios, por favor, complete estos datos.”
	2. Muestra un indicador sobre los campos vacíos.
Alternativa 2. “Existen datos incorrectos.”	
Acción del Actor	Respuesta del Sistema
	1. Muestra el mensaje de error “Existen campos escritos incorrectamente, por favor, rectifique estos datos.”
	2. Muestra un indicador sobre los campos incorrectos.
Alternativa 3. “Cancelar operación.”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción de Cancelar operación.	
	2. Regresa a la vista anterior.
	3. El caso de uso termina.
Alternativa 4. “Asignar una nueva Planilla Matrícula a Centro.”	
Acción del Actor	Respuesta del Sistema
7. Selecciona la opción de Asignar una nueva Planilla Matrícula a Centro.	
	8. Regresa al paso 2 del Flujo Normal de Eventos.
Poscondiciones	Se asignó una Planilla Matrícula a un Centro Docente.

Tabla 2.9 Descripción del CU “Asignar Planilla Matrícula a Centro”.

En este capítulo se presentaron las características que debe cumplir la solución de software para satisfacer las necesidades que dieron lugar a esta investigación. La realización de esta etapa en el proceso de desarrollo de un producto de software ha posibilitado un entendimiento aceptable de las necesidades de información y automatización en el proceso de matrícula en las secretarías docentes del MINSAP.

Capítulo 2: Características del Sistema

Además permitió sentar las bases para enfrentar las fases y flujos posteriores, previstos en la metodología de desarrollo utilizada, con la calidad requerida y esperada.

CAPÍTULO 3: Análisis y Diseño del Sistema

Este capítulo está dedicado a la etapa de desarrollo donde se profundiza en el análisis y diseño del sistema. Tras las definiciones y descripciones abordadas en el anterior capítulo, se hace necesario mostrar cómo se desarrolló el sistema.

Para ello, se definen diagramas de clases del análisis del sistema. Esto especifica qué clases del análisis toman parte del caso de uso y las relaciones entre ellas. Posteriormente, se muestran los diagramas de clases del modelo de diseño del sistema. También se describen algunas de las principales clases que se encuentran en el diseño.

3.1 Análisis

En esta etapa se refinan y estructuran los requisitos obtenidos con anterioridad, y se profundiza en el dominio de la aplicación. Esto permitirá una mayor comprensión del problema para modelar la solución. Los diagramas representados están organizados por paquetes para una mejor comprensión de los mismos.

3.1.1 Diagramas de clases del análisis

Los diagramas de clases del análisis muestran la definición de una arquitectura candidata basado en la recopilación de experiencias obtenidas en sistemas o dominios similares para restringir y definir la arquitectura del sistema de forma que no se malgasten esfuerzos en el redescubrimiento de la arquitectura.

Las clases mostradas en los siguientes diagramas y las relaciones entre ellas describen la realización de los casos de uso del sistema. La metodología RUP propone clasificar estas clases en: Interfaz, Control y Entidad.

En la tabla 3.1 se muestran los estereotipos que representan cada una de estas clases.




Clases del análisis		
 Interfaz	 Control	 Entidad

Tabla.3.1 Estereotipos de las clases del análisis.

Capítulo 3: Análisis y Diseño del sistema

Las clases interfaz modelan la interacción entre el sistema y sus actores, las clases control coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso y las clases entidad modelan la información que posee larga vida y que es a menudo persistente.

Algunos de los principales diagramas agrupados en los paquetes Registro y Configuración, son mostrados en las figuras siguientes.

3.1.1.1 Paquete de Registro

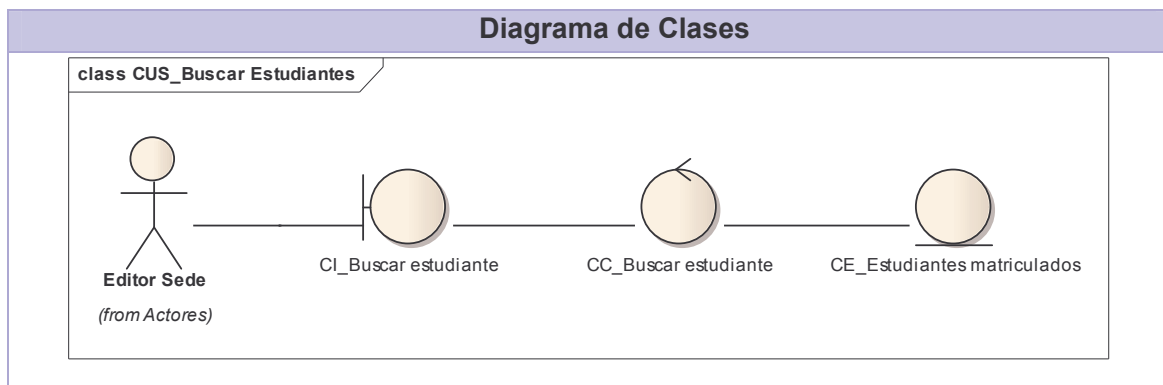


Fig.3.1 Diagrama de clases de análisis "CU Buscar Estudiante"

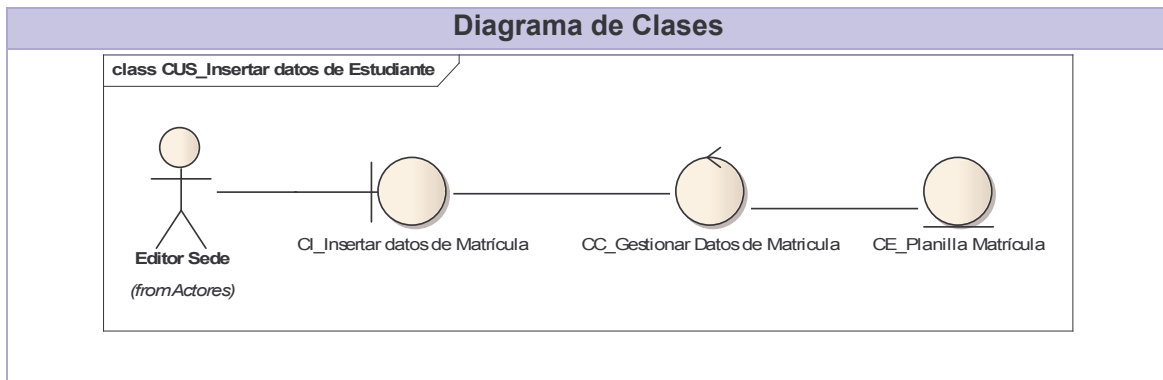


Fig.3.2 Diagrama de clases de análisis "CU Insertar Datos de Matrícula"

Diagrama de Clases

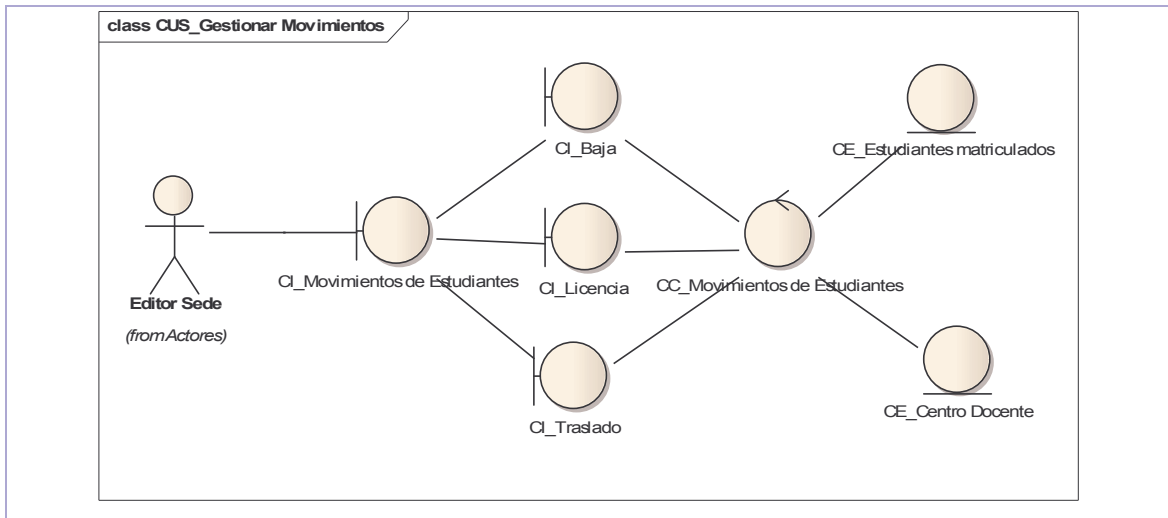


Fig.3.3 Diagrama de clases de análisis “CU Gestionar Movimientos”

3.1.1.2 Paquete de Configuración

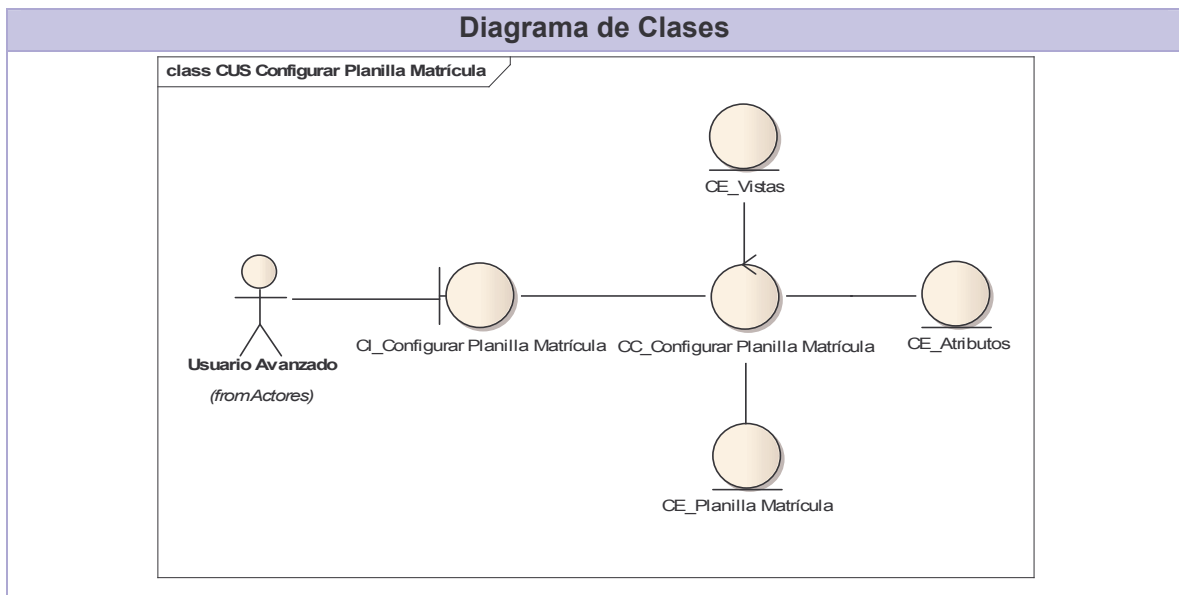


Fig.3.4 Diagrama de clases de análisis “CU Configurar Planilla Matrícula”.

Diagrama de Clases

Capítulo 3: Análisis y Diseño del sistema

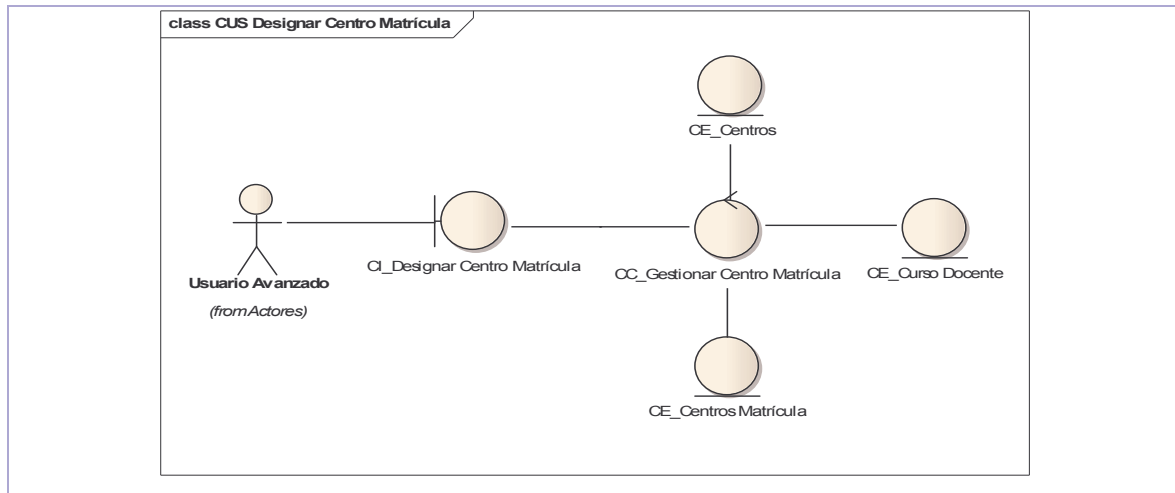


Fig.3.5 Diagrama de clases de análisis "CU Designar Centro Matrícula".

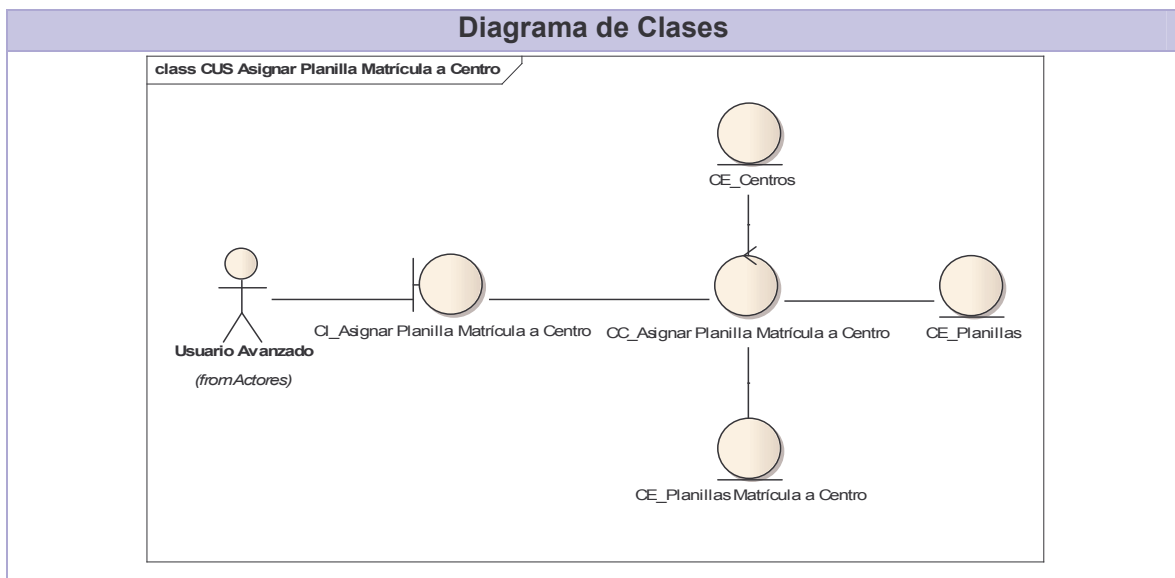


Fig.3.6 Diagrama de clases de análisis "CU Asignar Planilla Matrícula a Centro".

3.2 Diseño

Un Modelo de Diseño es una abstracción del Modelo de Implementación y de su código fuente, el cual fundamentalmente se emplea para representar y documentar su diseño. Es usado como entrada esencial en las actividades relacionadas con la implementación.

Representa los casos de usos en el dominio de la solución. Este modelo puede contener los diagramas, las clases, los paquetes, subsistemas, relaciones, colaboraciones, atributos, las relaciones de los casos de usos, entre otros que se puedan considerar para el desarrollo del sistema en cuestión.

Capítulo 3: Análisis y Diseño del sistema

Symfony está basado en el patrón arquitectónico MVC, el cual permite separar el código del programa en tres capas en dependencia de su naturaleza; la lógica relacionada con los datos se incluyen en el modelo, el código de la presentación en la vista y la lógica de la aplicación en el controlador. El diseño de la aplicación utilizando este patrón arquitectónico se muestra en las figuras que son presentadas seguidamente.

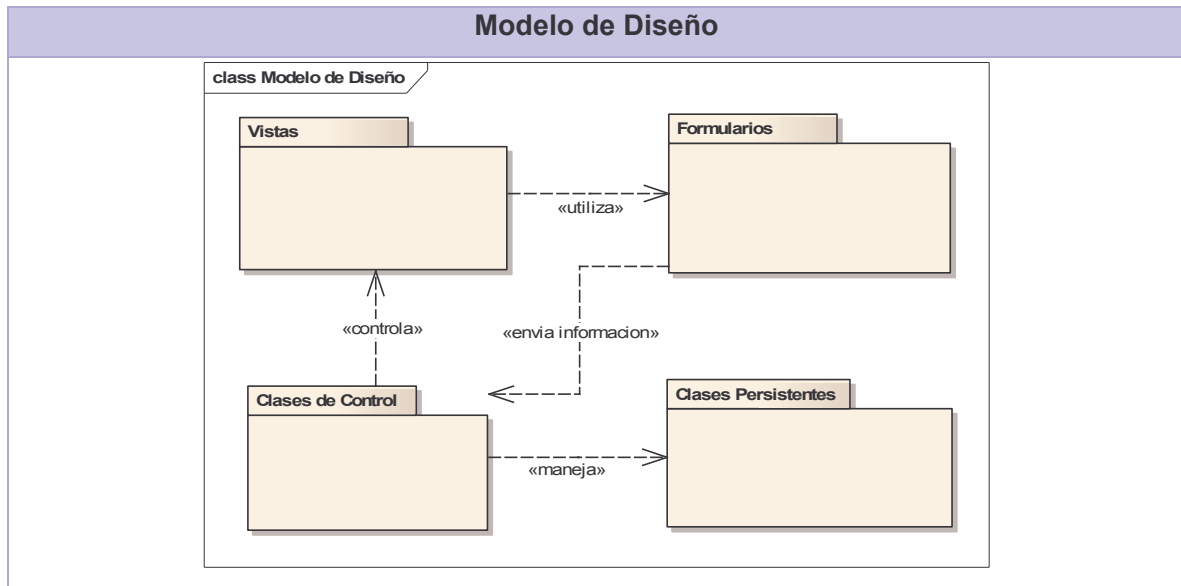


Fig.3.7 Esquema general del diseño.

4.1.1 Diagramas de clases del diseño

Los diagramas de clases muestran el diseño del sistema desde un punto de vista estático, a través de una colección de elementos declarativos, como clases, colaboraciones y sus relaciones.

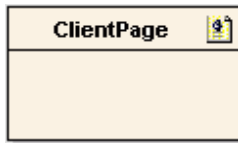
El MRE es una aplicación Web, por lo que se modelará el diseño haciendo uso de estereotipos definidos para este tipo de aplicaciones, como son los estereotipos web, lo cual permite una mejor comprensión de las funcionalidades del sistema.

A continuación se brinda una explicación de cómo son usados estos estereotipos en el diseño de la propuesta del sistema y que representa cada cual:

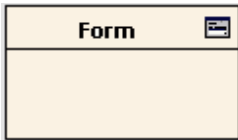


<<Server Page>>: Representa la clase que tiene código que se ejecuta en el servidor, la cual se encarga de construir (<<build>>) el resultado HTML, y realizar peticiones a la capa inferior del modelo.

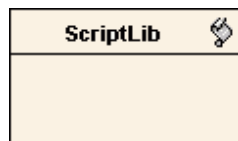
Capítulo 3: Análisis y Diseño del sistema



<<Client Page>>: Es una página web con formato .php, es interpretada por el navegador y cada una, es construida por una sola página servidor.



<<Form>>: Es una colección de elementos de entrada que están contenidos en la página cliente. Sus atributos son los elementos de entrada del formulario como son: textfields boxes, radio buttons, checkboxes, entre otros.



<<Script Library>>: Representan las clases JavaScript que se utilizan para las validaciones y también las librerías que son utilizadas para la creación de los componentes como: datatables, tabview, entre otros.

Relaciones utilizadas entre clases:

	Client Page	Server Page	Form	Script Library
Client Page		<<link>>	<<composite>>	<<include>>
Server Page	<<build>>	<<redirect>>		
Form		<<submit>>		
Script Library				

Tabla. 3.2 Relaciones entre clases del diseño

<<**build**>>: Representa la relación existente entre las páginas servidor y las páginas cliente, que de forma general expresa cómo las páginas que se encuentran en el servidor construyen las páginas en el cliente. Es una relación direccional, donde una página servidor construye una o más páginas cliente.

<<**link**>>: Representa la relación entre la página cliente y la página servidor y expresa la página cliente envía una acción a la página servidor para que esta la ejecute.

<<**redirect**>>: Representa cómo una página servidor puede redireccionar el procesamiento a otra página servidor, o sea, enviar información para que la otra ejecute la acción.

<<**composite**>>: Representa la relación existente entre las páginas cliente y los forms, mostrando cómo las páginas cliente están compuestas por estas páginas. Es una relación

Capítulo 3: Análisis y Diseño del sistema

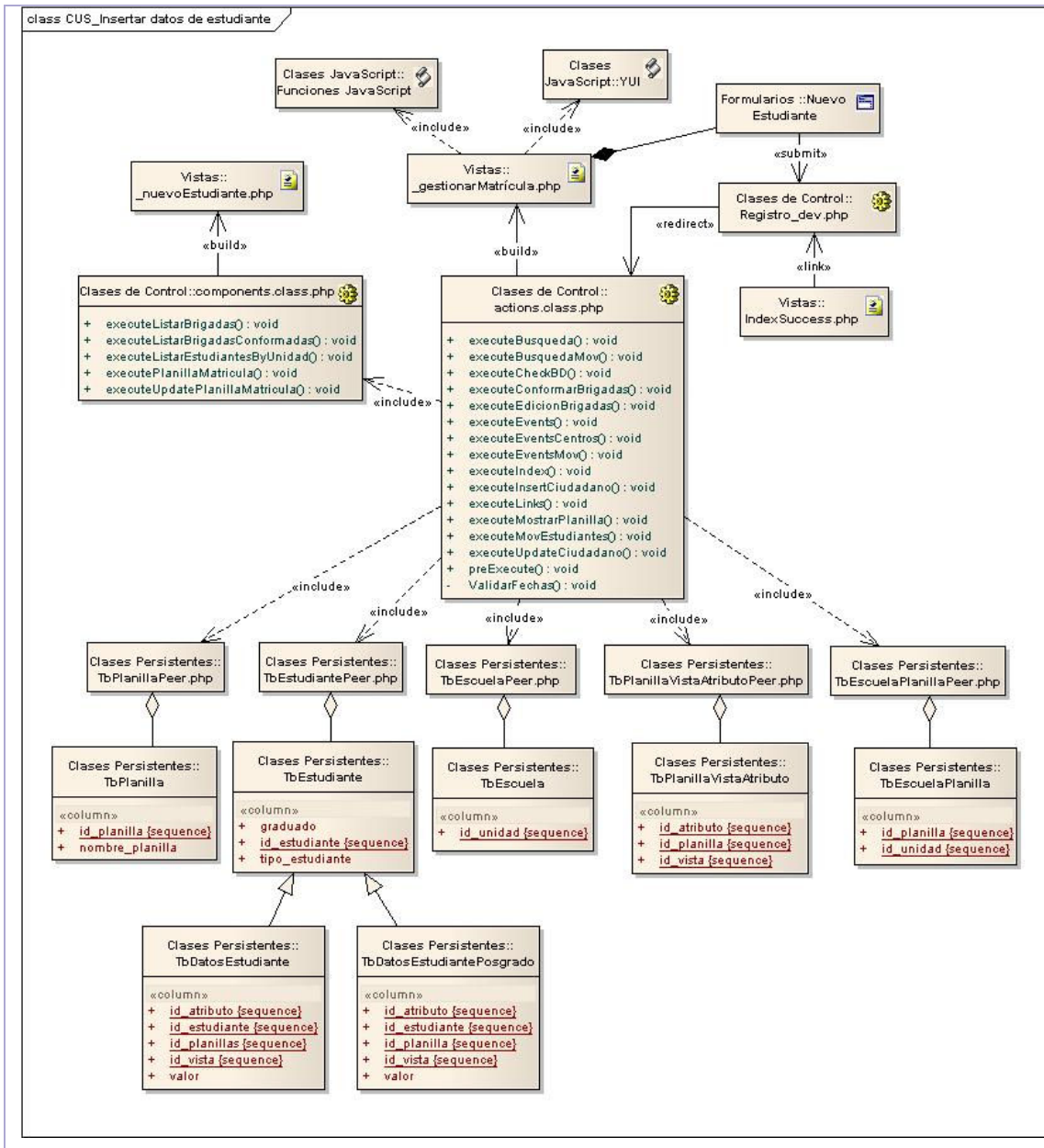


Fig. 3.9 Diagrama de Clases del Diseño “CU Insertar datos de estudiante”

Diagrama de Clases

Capítulo 3: Análisis y Diseño del sistema

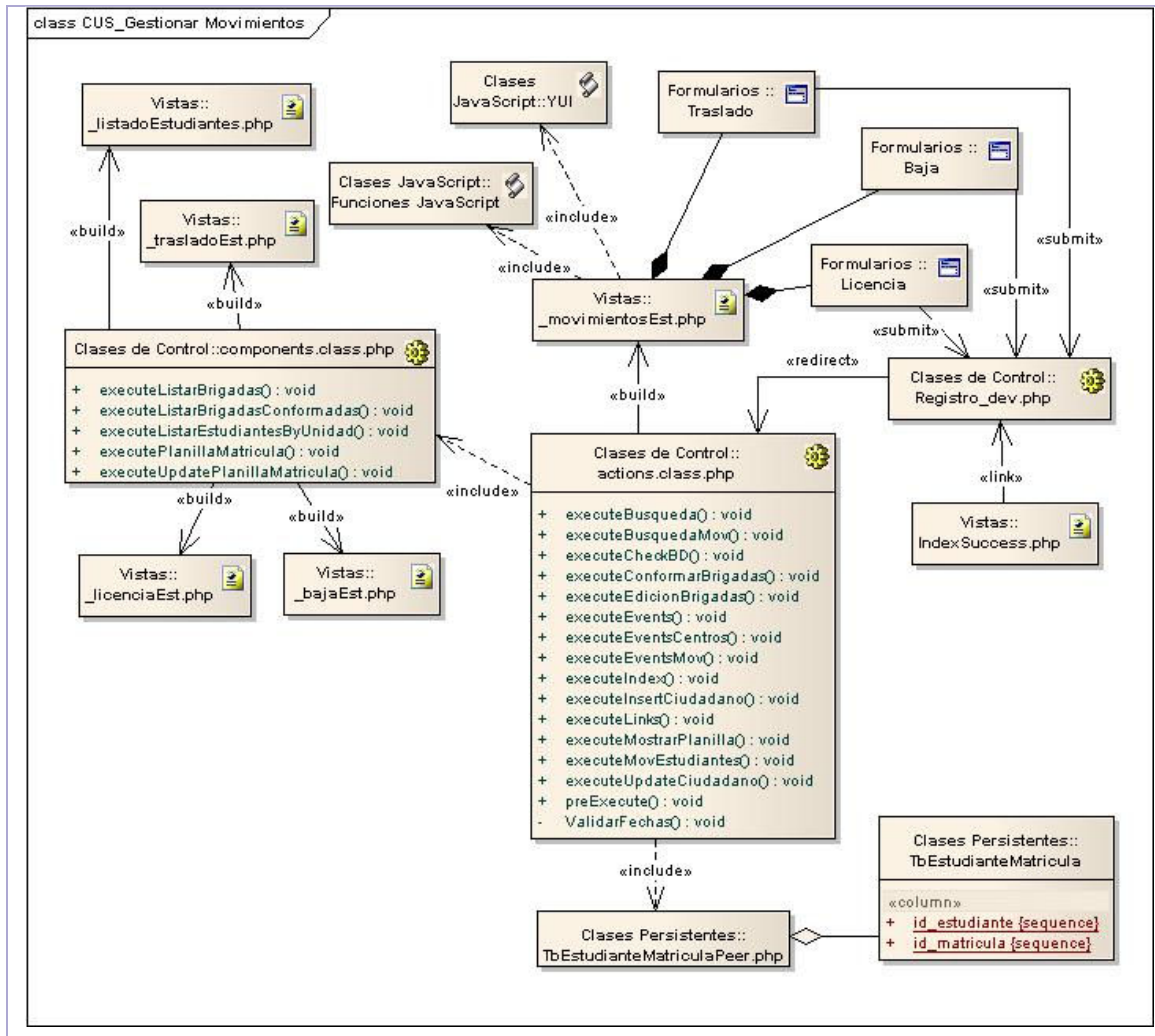


Fig. 3.10 Diagrama de Clases del Diseño “CU Gestionar Movimientos”

3.2.1.2 Paquete de Configuración

Diagrama de Clases

Capítulo 3: Análisis y Diseño del sistema

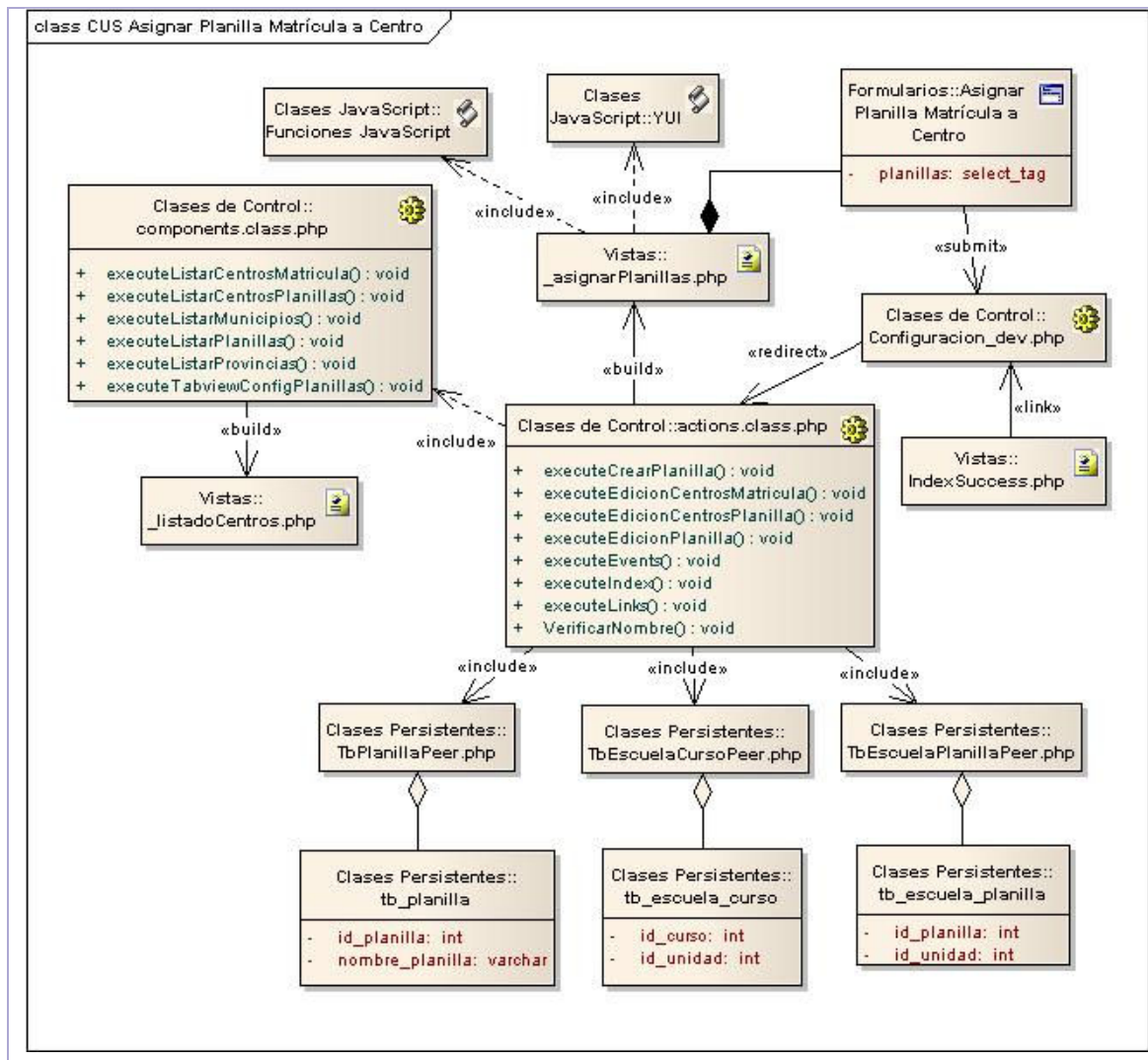


Fig.3.11 Diagrama de clases del diseño “CU Asignar Planilla Matrícula a Centro”.

Diagrama de Clases

Capítulo 3: Análisis y Diseño del sistema

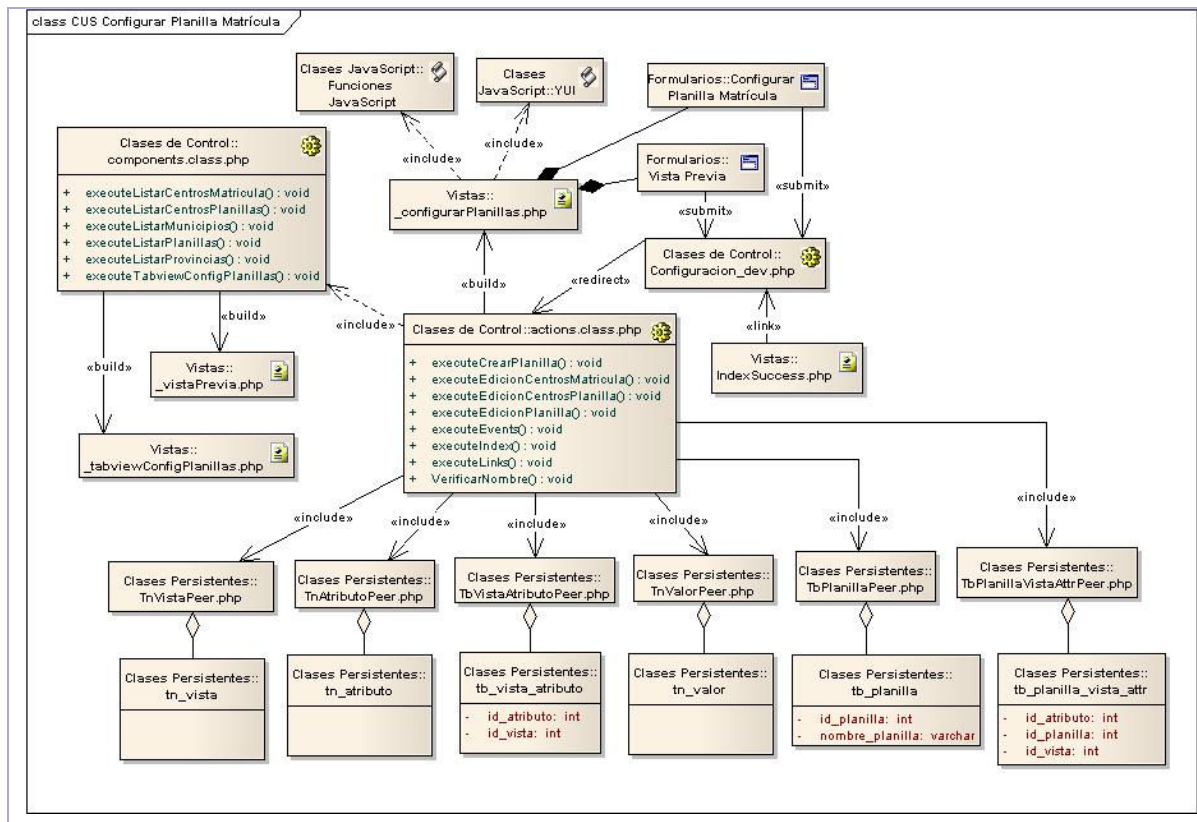


Fig.3.12 Diagrama de clases del diseño “CU Configurar Planilla Matrícula”.

Diagrama de Clases

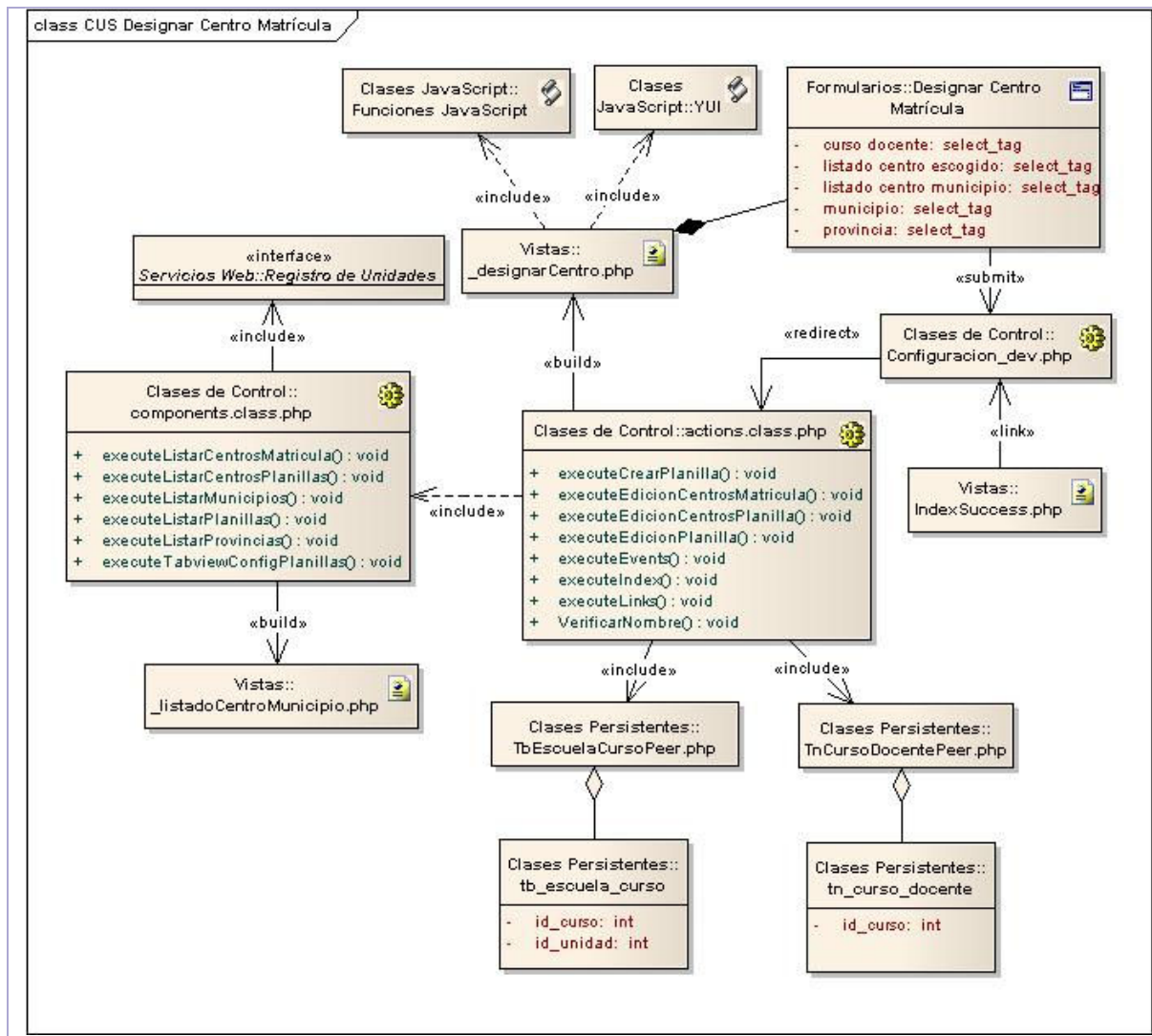


Fig.3.13 Diagrama de clases del diseño “CU Designar Centro Matrícula”

3.3 Descripción de las principales clases del diseño

Una clase es la descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica, definiendo como atributo a las propiedades o información detallada que posee.

En las tablas siguientes se describen algunas de las principales clases del diseño representadas en los diagramas mostrados anteriormente. Estas descripciones están agrupadas en los paquetes Registro y Configuración.

Paquete de Registro

Capítulo 3: Análisis y Diseño del sistema

➤ IndexSuccess.php

Nombre: indexSuccess.php	
Tipo de clase: Client Page	
Atributo	Tipo
Responsabilidad:	
Descripción:	Plantilla generada por defecto en Symfony que complementa la acción executeIndex ().

Tabla 3.3 Descripción de clases del diseño “indexSuccess.php”

➤ Actions.class.php

Nombre: actions.class.php	
Tipo de clase: Server Page	
Atributo	Tipo
Responsabilidad:	
Nombre:	executeIndex()
Descripción:	Acción generada por defecto en Symfony para la plantilla indexSuccess.
Nombre:	executeLinks()
Descripción:	Muestra las pantallas de cada una de las funcionalidades que se encuentran en el menú de navegación, según se seleccionen.
Nombre:	executeBusqueda()
Descripción:	Permite verificar si el estudiante se encuentra ubicado en el Registro de Ciudadanos.
Nombre:	executeCheckBD()
Descripción:	Verifica si el estudiante fue baja, traslado o licencia, además verifica si está matriculado en otro centro docente y permite editar los datos del mismo.

Capítulo 3: Análisis y Diseño del sistema

Nombre:	executeMostrarPlanilla()
Descripción:	Muestra la planilla de matrícula de un estudiante después de verificar si es posible realizar su matrícula o actualizar sus datos.
Nombre:	executeEvents()
Descripción:	Lista los municipios según la provincia seleccionada.
Nombre:	executeInsertCiudadano()
Descripción:	Permite matricular estudiantes de pregrado, posgrado y profesores en el centro docente.
Nombre:	executeUpdateCiudadano()
Descripción:	Permite actualizar los datos de matrícula de un estudiante o los datos de profesores.
Nombre:	executeEdicionBrigadas()
Descripción:	Permite insertar, actualizar o eliminar las brigadas docentes.
Nombre:	executeConformarBrigadas()
Descripción:	Permite insertar los estudiantes de pregrado matriculados en las brigadas docentes.
Nombre:	executeMovEstudiantes()
Descripción:	Permite realizar los movimientos a los estudiantes, dígase baja, licencia o traslado.
Nombre:	ValidarFechas()
Descripción:	Permite validar las fechas incorporación o reincorporación de licencia.
Nombre:	executeBusquedaMov()
Descripción:	Permite realizar una búsqueda de estudiantes matriculados en el centro.

Tabla 3.4 Descripción de clases del diseño "actions.class.php"

➤ **Components.class.php**

Nombre: components.class.php	
Tipo de clase: Server Page	
Atributo	Tipo

Capítulo 3: Análisis y Diseño del sistema

Responsabilidad:	
Nombre:	executePlanillaMatricula()
Descripción:	Permite mostrar la planilla matrícula.
Nombre:	executeUpdatePlanillaMatricula()
Descripción:	Muestra la planilla matrícula correspondiente al tipo de estudiante o profesor para actualizar los datos de estos.
Nombre:	executeListarBrigadas()
Descripción:	Muestra un listado con todas las brigadas del centro docente.
Nombre:	executeListarBrigadasConformadas()
Descripción:	Muestra un listado de las brigadas conformadas en el centro docente.

Tabla 3.5 Descripción de clases del diseño “components.class.php”

➤ Registro_dev.php

Nombre: registro_dev.php	
Tipo de clase: Server Page	
Atributo	Tipo
Responsabilidad:	
Descripción:	Controlador frontal de la aplicación encargado de garantizar que el sistema tenga un único punto de entrada y de salida de datos.

Tabla 3.6 Descripción de clases del diseño “registro_dev.php”

➤ _busquedaEstudiante.php

Nombre: _busquedaEstudiante.php	
Tipo de clase: Client Page	
Atributo	Tipo
Responsabilidad:	

Capítulo 3: Análisis y Diseño del sistema

Descripción:	Permite realizar una búsqueda del estudiante que será matriculado, matricularlo y actualizar los datos de este.
--------------	---

Tabla 3.7 Descripción de clases del diseño “_busquedaEstudiante.php”

➤ **_movimientosEstudiantes.php**

Nombre: _movimientosEstudiantes.php	
Tipo de clase: Client Page	
Atributo	Tipo
Responsabilidad:	
Descripción:	Permite asignarle baja, licencia o traslado a un estudiante de pregrado.

Tabla 3.8 Descripción de clases del diseño “_movimientosEstudiantes.php”

Paquete de Configuración

➤ **IndexSuccess.php**

Nombre: indexSuccess.php	
Tipo de clase: Client Page	
Atributo	Tipo
Responsabilidad:	
Descripción:	Plantilla generada por defecto en Symfony que complementa la acción executeIndex ().

Tabla 3.9 Descripción de clases del diseño “IndexSuccess.php”

➤ **Actions.class.php**

Nombre: actions.class.php	
Tipo de clase: Server Page	

Capítulo 3: Análisis y Diseño del sistema

Atributo	Tipo
Responsabilidad:	
Nombre:	executeIndex()
Descripción:	Acción generada por defecto en Symfony para la plantilla indexSuccess.
Nombre:	executeLinks()
Descripción:	Muestra las pantallas de cada una de las funcionalidades que se encuentran en el menú de navegación, según se seleccionen.
Nombre:	executeEvents()
Descripción:	Lista los municipios según la provincia seleccionada y las unidades de salud del municipio señalado.
Nombre:	executeEdicionCentrosMatricula()
Descripción:	Permite eliminar o asignar un curso a un centro docente.
Nombre:	executeEdicionCentrosPlanilla()
Descripción:	Permite asignar, reasignar o eliminar una planilla a un centro docente.
Nombre:	executeCrearPlanilla()
Descripción:	Permite crear la planilla matrícula y ver una vista previa de esta.
Nombre:	VerificarNombre()
Descripción:	Verifica que no existan nombres de planillas duplicados.
Nombre:	executeEdicionPlanilla()
Descripción:	Permite actualizar o eliminar la planilla matrícula.

Tabla 3.10 Descripción de clases del diseño “actions.class.php”

➤ **Components.class.php**

Nombre: components.class.php	
Tipo de clase: Server Page	
Atributo	Tipo

Capítulo 3: Análisis y Diseño del sistema

Responsabilidad:	
Nombre:	executeListarCentrosMatricula()
Descripción:	Muestra un listado de los centros con cursos asignados.
Nombre:	executeTabviewConfigPlanillas()
Descripción:	Muestra las vistas y atributos que permiten configurar la planilla matrícula.
Nombre:	executeListarProvincias()
Descripción:	Muestra un listado de todas las provincias del país.
Nombre:	executeListarMunicipios()
Descripción:	Muestra un listado de los municipios de la provincia por defecto.
Nombre:	executeListarCentrosPlanillas()
Descripción:	Muestra un listado de los centros con planillas asignadas.
Nombre:	executeListarPlanillas()
Descripción:	Muestra un listado de las planillas creadas.

Tabla 3.11 Descripción de clases del diseño “components.class.php”

➤ Configuración_dev.php

Nombre: configuración_dev.php	
Tipo de clase: Server Page	
Atributo	Tipo
Responsabilidad:	
Descripción:	Controlador frontal de la aplicación encargado de garantizar que el sistema tenga un único punto de entrada y de salida de datos.

Tabla 3.12 Descripción de clases del diseño “configuración_dev.php”

➤ _asignarPlanillas.php

Nombre: _asignarPlanillas.php	
Tipo de clase: Client Page	

Capítulo 3: Análisis y Diseño del sistema

Atributo	Tipo
Responsabilidad:	
Descripción:	Permite asignar la planilla matrícula a un centro docente.

Tabla 3.13 Descripción de clases del diseño “_asignarPlanillas.php”

➤ **_configurarPlanillas.php**

Nombre: _configurarPlanillas.php	
Tipo de clase: Client Page	
Atributo	Tipo
Responsabilidad:	
Descripción:	Permite configurar las vistas y los atributos que conforman la planilla matrícula.

Tabla 3.14 Descripción de clases del diseño “_configurarPlanillas.php”

➤ **_gestionarCentros.php**

Nombre: _gestionarCentros.php	
Tipo de clase: Client Page	
Atributo	Tipo
Responsabilidad:	
Descripción:	Permite designar los centros que brindarán matrícula o eliminar esta designación.

Tabla 3.15 Descripción de clases del diseño “_gestionarCentros.php”

3.4 Definiciones de diseño

El diseño de la interfaz es otro punto fundamental a tratar a la hora de presentar la aplicación; teniendo en cuenta que esa es la capa de presentación al usuario y por lo tanto debe ser lo más amigable y de fácil entendimiento posible. Por esta razón, debe ser sencilla, coherente y necesaria para el usuario.

En este caso, la aplicación está dirigida a personal que labora en el sector de la salud, que en ocasiones tienen un conocimiento mínimo del uso de la computadora. Por lo tanto, en vista de lograr una interfaz clara y fácil de usar, estarán encaminados los esfuerzos del diseño.

Con este principio, en el diseño se tuvieron en cuenta aspectos como: organización de los elementos en la pantalla y estructura de la misma. Para esto todas las páginas tienen un diseño uniforme garantizando que el usuario no se pierda en la navegación. A continuación se explican algunos puntos tenidos en cuenta:

- El equilibrio en la organización de la información, por ejemplo, que todas las páginas que muestran información siempre lo harán en el mismo orden.
- La optimización de la cantidad de elementos en la pantalla, ayudando al fácil manejo y mejor comprensión de la información mostrada.
- La unidad, donde cada elemento de la pantalla se diseñará siguiendo un patrón de tamaño, colores y formas, utilizando para ello hojas de estilos.
- Anticipar lo que el usuario quiere y necesita, brindándole toda la información y herramientas necesarias para cada paso del proceso, es decir los menús y las barras cabeceras estarán disponibles en todas las páginas en que el usuario tenga acceso.
- El diseño no dependerá de las funciones del navegador para ayudar en la navegación.
- La estética nunca deberá sustituir la funcionalidad.

Los elementos que se repitan en las distintas páginas, se situarán en un mismo lugar para mejor manejo de la información. También se trabajará sobre la base de que las páginas no se encuentren muy cargadas. Sólo deberán llevar la información necesaria para mayor

Capítulo 3: Análisis y Diseño del sistema

claridad. A continuación se muestran los diseños de la estructura que deben tener las pantallas del componente siguiendo todas las pautas anteriormente planteadas.

La siguiente figura muestra el diseño del layout, a continuación se señala con números las partes del mismo. Banner [1], Vínculos [2], Cuerpo [3], Pie [4].



Fig.3.14 Divisiones de la estructura de la página

La descripción de los elementos que componen las pantallas son las siguientes:

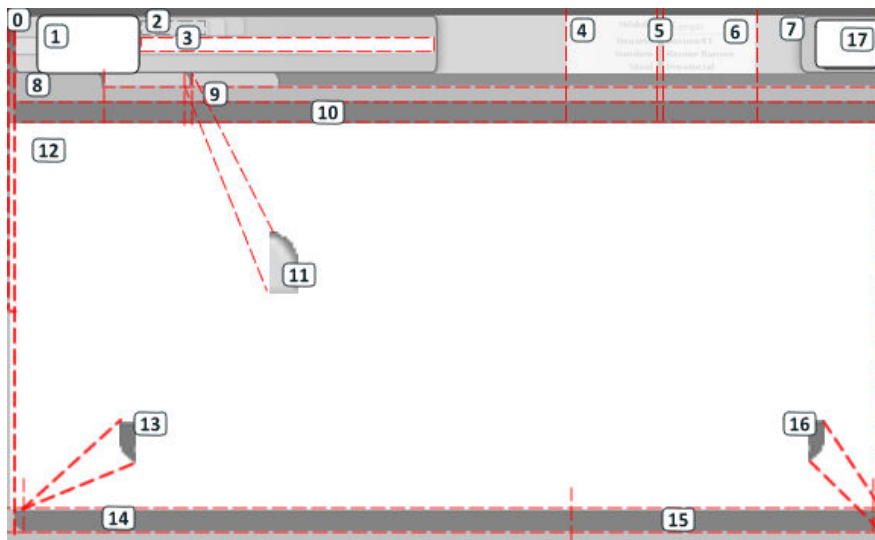


Fig.3.15 Partes del cuerpo

Leyenda	
0. Background o fondo.	8. Tabs o pestañas del menú.
1. Imagen o isotipo representativo del	9. Submenú.

Capítulo 3: Análisis y Diseño del sistema

sistema.	
2. Logotipo del sistema.	10. Funcionalidad seleccionada.
3. Nombre del sistema.	11. Icono representativo de la funcionalidad.
4. Criterios a mostrar del usuario autenticado.	12. Cuerpo.
5. Separador.	13. Imagen izquierda del pie.
6. Datos del usuario autenticado.	14. Parte izquierda del pie. 15. Footer o pie.
7. Superior derecho para colocar una imagen con información.	16. Imagen derecha del pie.

Tabla 3.16 Descripción de las partes del diseño del cuerpo.

3.5 Tratamiento de errores

A menudo en las aplicaciones no se tiene en cuenta durante su desarrollo la importancia del tratamiento de errores en el sistema. Esto puede traer dificultades, y es que a menudo se proporciona al usuario de servicio, más información de la necesaria.

Para contrarrestar estas situaciones se llevan a cabo una serie de acciones como el tratamiento de errores que en la aplicación está presente del lado del cliente. Se utiliza código JavaScript combinando el uso de expresiones regulares que brindan fortaleza y garantizan la ejecución y rapidez del sistema. Del lado del servidor se le da un tratamiento a los errores de forma tal que las operaciones de inserción, eliminación y modificación de registros en la base de datos, se realicen de forma correcta.

En el caso de la inserción se implementarán funciones que validen la entrada de datos y en caso de existir errores, se mostrarán mensajes que aclaren la inserción o modificación errónea de algún dato. Los mensajes emitidos serán fácilmente identificables, siendo estos de tipo informativo, de error o de advertencia. Los informativos se refieren a la respuesta del sistema al realizar alguna operación y las advertencias se mostrarán antes de realizar alguna acción que afecte otros procesos o suponga una modificación de la información registrada. Por otro lado los mensajes de error se emitirán cuando no se introduzca información obligatoria, o esta sea incorrecta.

Capítulo 3: Análisis y Diseño del sistema

A continuación se muestra la iconografía que se utiliza en el sistema para mostrar los diferentes tipos de errores:





Íconos para mensajes de error	
 Mensaje de Error	 Mensaje Correcto
 Mensaje Warning	 Mensaje Informativo

Tabla 3.17 Iconografía de mensajes

Otros errores pueden ser generados por el gestor de base de datos, los cuales se capturan antes de mostrarse al usuario y una vez tratados por el sistema, son mostrados al usuario de una manera entendible para él.

3.6 Estándar de codificación

Es fundamental para desarrollar cualquier software tener reglas o estándares de codificación. Un código bien escrito y documentado y que guarde siempre el mismo estilo, permite el reconocimiento de las funciones y una lectura comprensiva.

Tipo Datos	Prefijo	Ejemplo
Int	i	iCantPacientes
Float	f	fPesoPaciente
double	d	dPesoCarro
Bool	b	bPacienteActivo
String	s	sNombrePaciente
char	c	cLetra
De tipo enum	ev	evSexo
Byte	b	bCantDiasPaciente
Sbyte	sb	sbEdadPaciente
Short	sh	shVariableShort
ushort	us	usVariableUshort
UInt	ui	uiVariableUInt
Long	l	lVariableLong
Ulong	ul	ulVariableUlong
decimal	dc	dcVariableDecimal
Objetos	o	oPacienteHistorico
Objetos de tipo Struct	st	stUnaStruct

Tabla 3.18 Estándar de codificación para variables

Capítulo 3: Análisis y Diseño del sistema

Las normas de codificación para los elementos visuales de los formularios de la interfaz gráfica de usuario son los presentados en la tabla que se muestra a continuación.

Control	Prefijo	Ejemplo
Botón	btn	btnAceptar
Etiqueta	lbl	lblNombre
Lista/Menú	mn	mnPrincipal
Campo de Texto	txt	txtFecha
Botón de Opción	bpt	optSexo
Casilla de Verificación	chx	chxBorrar
Casilla de Selección	cbx	cbxSexo

Tabla 3.19 Estándar de codificación para componentes de interfaz

En este capítulo se muestra el diseño de la solución de software, fruto de atravesar las fases de análisis y diseño en el proceso de ingeniería del software para el desarrollo del módulo. Quedaron plasmados todos los artefactos y diagramas necesarios definidos para esta fase, tanto en la investigación como en la documentación técnica del producto.

Esta etapa permitió pasar a la fase de implementación de la solución de software, aportando a los programadores las características de la solución y la información indispensable para la codificación de la misma.

CAPÍTULO 4: Implementación

En este capítulo se describe el sistema en términos de componentes. Los cuales recogen scripts, ficheros códigos fuentes, ejecutables y similares a partir del resultado obtenido en el capítulo anterior y se representa el diagrama entidad-relación de la base de datos así como las descripciones de las tablas que la conforman. Además se describe cómo se organiza el sistema de acuerdo a los nodos específicos en el modelo de despliegue y algunos servicios Web a consumir. Igualmente se expone la integración del componente a otros sistemas.

4.1 Diseño de la Base de Datos

Un modelo de datos es un sistema formal y abstracto que permite describir los datos de acuerdo con reglas y convenios predefinidos. Los objetos del sistema se manipulan siguiendo reglas perfectamente definidas y utilizando exclusivamente los operadores definidos para el sistema independientemente de lo que estos objetos y operadores puedan significar.

Este modelo proporciona una representación visual y física de los datos persistentes del sistema, obteniéndose el modelo de datos.

A continuación se representa el modelo de datos del MRE.

Modelo de datos

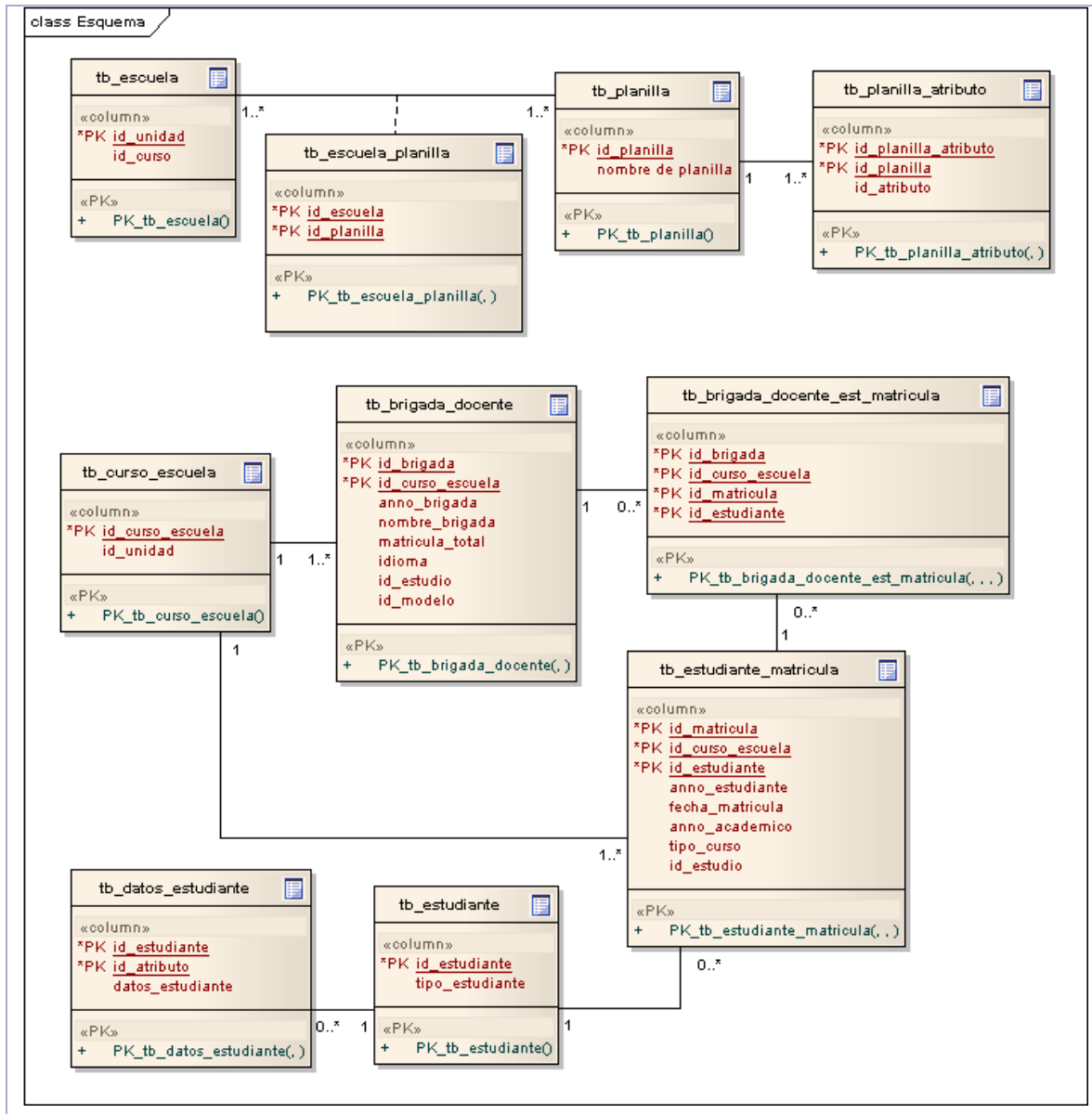


Fig. 4.1 Modelo de datos

4.1.1 Descripción de las tablas de la Base de Datos Registro

Las descripciones de las tablas y los atributos de la base de datos son de gran importancia pues se describen las entidades que forman parte del modelo de datos, y que consistirán en las tablas de la base de datos que se construyó. Estas descripciones se muestran a continuación:

Nombre: tb_planilla
Descripción: Almacena información referente a las planillas de matrícula.

Capítulo 4: Implementación

Atributo	Tipo	Nulo	Descripción
id_planilla	interger	no	Representa el identificador de la planilla de matrícula.
nombre_planilla	varchar (50)	si	Representa el nombre de la planilla de matrícula.

Tabla 4.1 Descripción de la tabla “tb_planilla”.

Nombre: tb_escuela			
Descripción: Almacena los id de los centros docentes que gestionan información.			
Atributo	Tipo	Nulo	Descripción
id_unidad	interger	no	Representa el identificador del centro docente.

Tabla 4.2 Descripción de la tabla “tb_escuela”.

Nombre: tb_escuela_planilla			
Descripción: Almacena la información referente a las planillas de matrícula que le son asignadas a un centro determinado.			
Atributo	Tipo	Nulo	Descripción
id_unidad	interger	no	Representa el identificador del centro docente. Llave foránea absorbida de ta tabla tb_escuela.
id_planilla	interger	no	Representa el identificador de la planilla de matrícula. Llave foránea absorbida de ta tabla tb_planilla.

Tabla 4.3 Descripción de la tabla “tb_escuela_planilla”.

Nombre: tb_planilla_vista_attr			
Descripción: Almacena la información referente a las vistas y atributos que le son asignadas a una planilla de matrícula.			
Atributo	Tipo	Nulo	Descripción
id_planilla	interger	no	Representa el identificador de la planilla de matrícula. Llave foránea absorbida de ta tabla tb_planilla.
id_vista	interger	no	Representa el identificador de la vista. Llave foránea absorbida de ta tabla tb_vista del esquema nomencladores.
id_atributos	interger	no	Representa el identificador de los atributos. Llave foránea absorbida de ta tabla tb_atributo del esquema nomencladores.

Tabla 4.4 Descripción de la tabla “tb_planilla_vista_attr”.

Nombre: tb_estudiante			
Descripción: Almacena la información específica referente a los estudiantes que matriculen en un centro determinado.			
Atributo	Tipo	Nulo	Descripción
id_estudiante	interger	no	Representa el identificador del estudiante. Llave foránea absorbida de ta tabla tb_ciudadano del esquema nomencladores.
tipo_estudiante	interger	no	Representa el tipo de estudiante que se ha

Capítulo 4: Implementación

			registrado.
graduado	boolean	si	Se refiere a si el estudiante es graduado de la carrera o no.

Tabla 4.5 Descripción de la “tabla tb_estudiante”.

Nombre: tb_datos_estudiante			
Descripción: Almacena la información referente a los estudiantes que cursan los estudios superiores en un centro docente determinado.			
Atributo	Tipo	Nulo	Descripción
id_estudiante	integer	no	Representa el identificador del estudiante. Llave foránea absorbida de la tabla tb_estudiante.
id_planilla	integer	no	Representa el identificador de la planilla de matrícula. Llave foránea absorbida de ta tabla tb_planilla_vista_attr.
id_vista	integer	no	Representa el identificador de la vista. Llave foránea absorbida de ta tabla tb_planilla_vista_attr.
id_atributos	integer	no	Representa el identificador de los atributos. Llave foránea absorbida de ta tabla tb_planilla_vista_attr.
Valor	varchar (100)	si	Representa la información que se le recoge a los estudiantes, de estudios superiores durante el registro en un centro determinado.

Tabla 4.6 Descripción de la tabla “tb_datos_estudiante”.

Nombre: tb_datos_est_posgrado			
Descripción: Almacena la información referente a los estudiantes de posgrado en un centro docente determinado.			
Atributo	Tipo	Nulo	Descripción
id_estudiante	integer	no	Representa el identificador del estudiante. Llave foránea absorbida de la tabla tb_estudiante.
id_planilla	integer	no	Representa el identificador de la planilla de registro. Llave foránea absorbida de ta tabla tb_planilla_vista_attr.
id_vista	integer	no	Representa el identificador de la vista. Llave foránea absorbida de ta tabla tb_planilla_vista_attr.
id_atributos	integer	no	Representa el identificador de los atributos. Llave foránea absorbida de ta tabla tb_planilla_vista_attr.
Valor	varchar (100)	si	Representa la información que se le recoge a los estudiantes de posgrado durante el registro en un centro determinado.

Tabla 4.7 Descripción de la tabla “tb_datos_est_posgrado”.

Nombre: tb_estudiante_matricula			
Descripción: Almacena la información referente a la matrícula de los estudiantes en			

Capítulo 4: Implementación

un centro determinado.			
Atributo	Tipo	Nulo	Descripción
id_matricula	interger	no	Define el identificador de la matrícula de un estudiante en un centro determinado.
id_estudiante	interger	no	Representa el identificador del estudiante. Llave foránea absorbida de la tabla tb_estudiante.
anno_estudiante	varchar(50)	si	Representa el año académico en se encuentra el estudiante que cursa estudios superiores.
fecha_matricula	date	si	Define el día en que se realiza la matrícula.
anno_academico	interger	si	Representa el año académico en se encuentra el estudiante que cursa estudios superiores.
tipo_curso	varchar(30)	si	Representa el tipo de curso en que se encuentra el estudiante, diurno, por trabajadores, ect.
id_estudio	interger	si	Representa el estudio que está cursando el estudiante. Llave foránea absorbida de la tabla tb_estudio del esquema nomencladores

Tabla 4.8 Descripción de la tabla "tb_estudiante_matricula".

Nombre: tb_escuela_curso			
Descripción: Almacena la información el curso docente que le es asignado a un determinado centro docente.			
Atributo	Tipo	Nulo	Descripción
id_curso	interger	no	Representa el identificador del curso docente. Llave foránea absorbida de la tabla tb_curso_docente del esquema nomencladores.
id_unidad	interger	no	Representa el identificador del centro docente. Llave foránea absorbida de la tabla tb_escuela.

Tabla 4.9 Descripción la tabla "tb_escuela_curso".

Nombre: tb_grupo_docente			
Descripción: Almacena la información referente a los grupos docentes de un determinado centro.			
Atributo	Tipo	Nulo	Descripción
id_grupo	interger	no	Representa el identificador del grupo docente.
id_curso	interger	no	Representa el identificador del curso docente. Llave foránea absorbida de la tabla tb_escuela_curso.
id_unidad	interger	no	Representa el identificador del centro docente. Llave foránea absorbida de la tabla tb_escuela_curso.
anno_brigada	varchar(50)	si	Representa el año académico que tienen los estudiantes del grupo.

Capítulo 4: Implementación

nombre_brigada	varchar(50)	si	Define el nombre de cada uno de los grupos.
matricula_total	interger	si	Representa la cantidad máxima de estudiantes que va a tener cada grupo.
Idioma	varchar(20)	si	Representa el idioma que hablan los estudiantes del grupo.
id_estudio	interger	si	Define el estudio de los estudiantes del grupo. Llave foránea absorbida de la tabla tb_estudio del esquema nomencladores.
id_modelo	interger	Si	Define el modelo de formación de los estudiantes del grupo. Llave foránea absorbida de la tabla tb_modelo_formacion del esquema nomencladores

Tabla 4.10 Descripción de la tabla “tb_grupo_docente”.

Nombre: tb_grupo_estudiante			
Descripción: Almacena la información referente al grupo docente a que pertenecen los estudiantes en un determinado centro.			
Atributo	Tipo	Nulo	Descripción
id_grupo	interger	no	Representa el identificador del grupo docente. Llave foránea absorbida de la tabla tb_grupo_docente.
id_curso	Interger	no	Representa el identificador del curso docente. Llave foránea absorbida de la tabla tb_grupo_docente.
id_unidad	Interger	no	Representa el identificador del centro docente. Llave foránea absorbida de la tabla tb_grupo_docente.
id_matricula	Interger	no	Representa la matrícula del estudiante en el centro docente. Llave foránea absorbida de la tabla tb_estudiante_matricula.
id_estudiante	Interger	no	Representa el identificador del estudiante. Llave foránea absorbida de la tabla tb_estudiante_matricula.

Tabla 4.11 Descripción de la tabla “tb_grupo_estudiante”.

4.2 Modelo 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. Detalla las capacidades de red, las especificaciones del servidor, los requisitos de hardware y otra información relacionada al despliegue del sistema propuesto.

A continuación se muestra la propuesta de despliegue del sistema. La misma puede variar en dependencia de las posibilidades y recursos con que cuenta la institución cliente, que en este caso es el MINSAP.

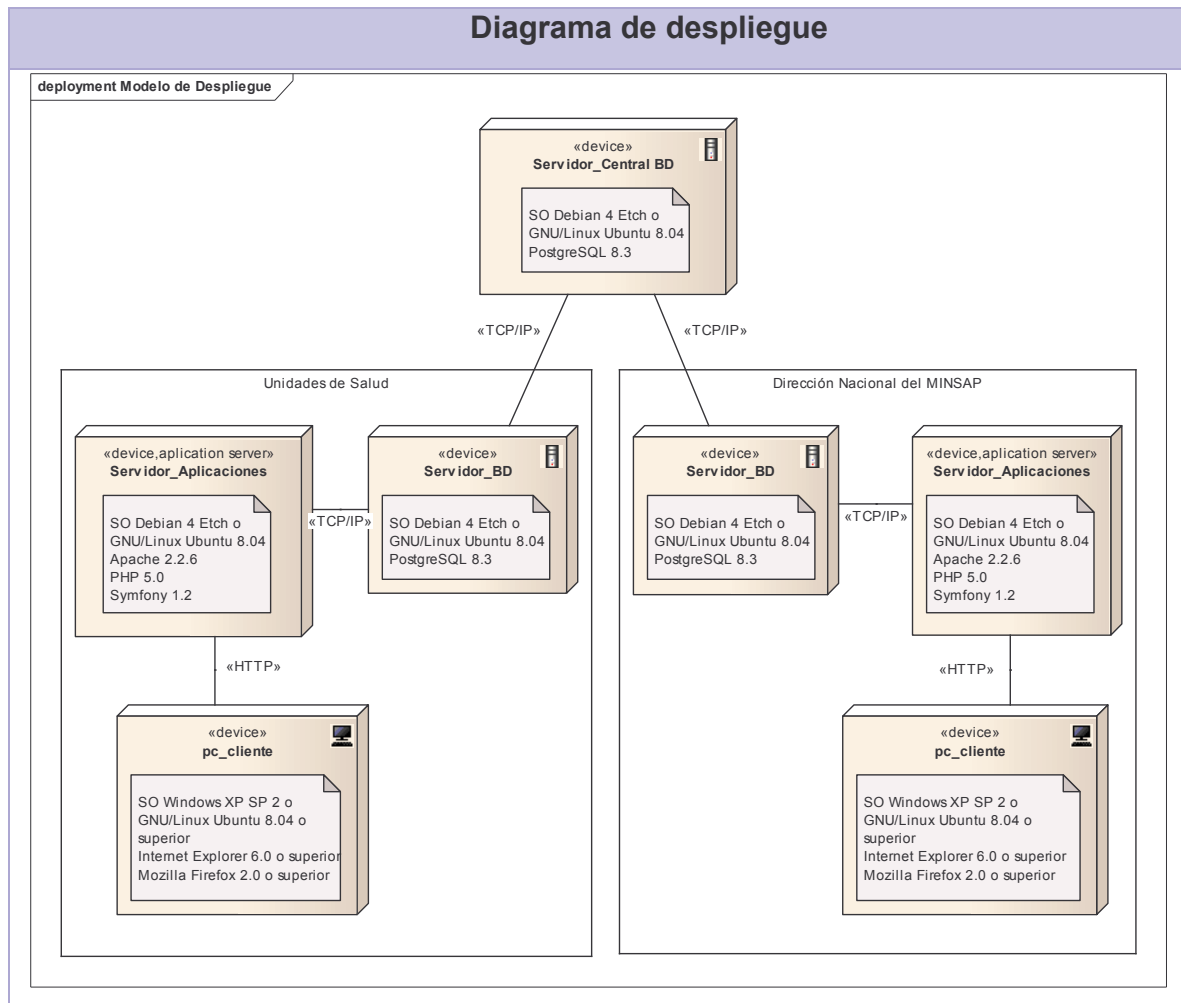


Fig. 4.2 Diagrama de despliegue

El Módulo Registro de Estudiantes estará desplegado en el servidor de aplicaciones Apache. En el servidor de base de datos se encuentra la capa de datos de la aplicación, en la cual se almacena la información que se gestiona a través de la misma. En el dispositivo que representa a la PC_Cliente se encuentra el navegador Web, a través del cual el usuario puede acceder a la aplicación y para el trabajo de impresión en las secretarías docentes, de toda la documentación referente al registro de estudiantes, se utilizará una impresora.

4.3 Modelo de Implementación

El modelo de implementación está compuesto por una colección de componentes y subsistemas. Estos componentes incluyen ficheros ejecutables, de código fuente o cualquier otro tipo necesarios para obtener la versión del sistema y llevarlo al despliegue.

Capítulo 4: Implementación

Se describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración disponibles en el entorno de la implementación.

Un diagrama de componentes contiene componentes, interfaces y las relaciones entre ellos y como todo diagrama puede contener paquetes, utilizados para agrupar elementos del modelo.

Seguidamente, se representan los modelos de componentes de los paquetes Registro y Configuración.

4.3.4 Paquete Registro

El MRE implementa una arquitectura MVC, por ello se decidió agrupar los componentes del modelo de implementación en tres paquetes nombrados de la misma forma, los paquetes Modelo, Vista y Controlador.

La figura 4.3 muestra el modelo de implementación de forma general.

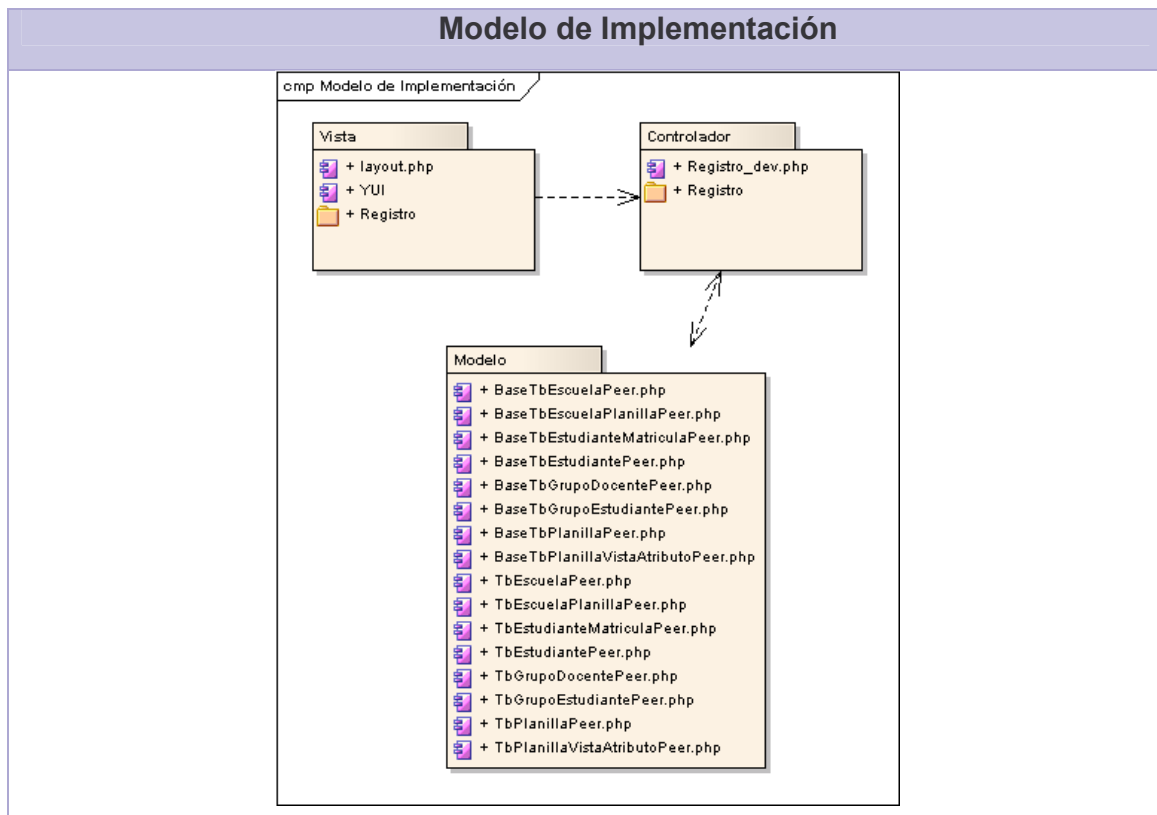


Fig. 4.3 Modelo de implementación general "Registro"

Capítulo 4: Implementación

El paquete Vista está compuesto por los componentes que representan las librerías utilizadas para la implementación de las interfaces mostradas al usuario, las cuales están agrupadas en el paquete Registro que se puede ver, en la figura 4.4; así como la plantilla global con el código común a todas las páginas de la aplicación.

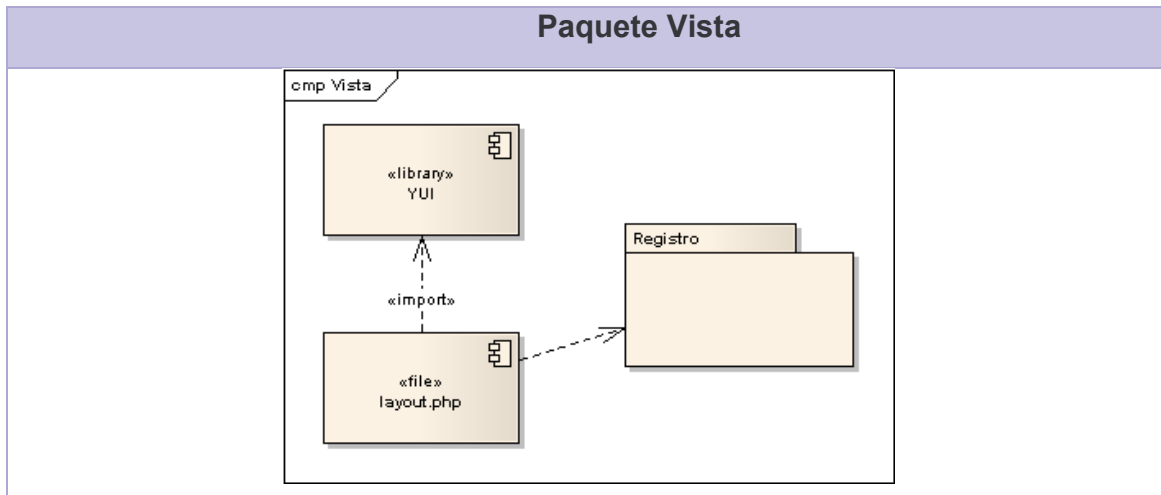


Fig. 4.4 Modelo de componentes "Vista"

En el caso del paquete Controlador, se puede ver en la figura que contiene la representación en términos de componentes tanto del controlador frontal de la aplicación como del controlador del módulo Registro.

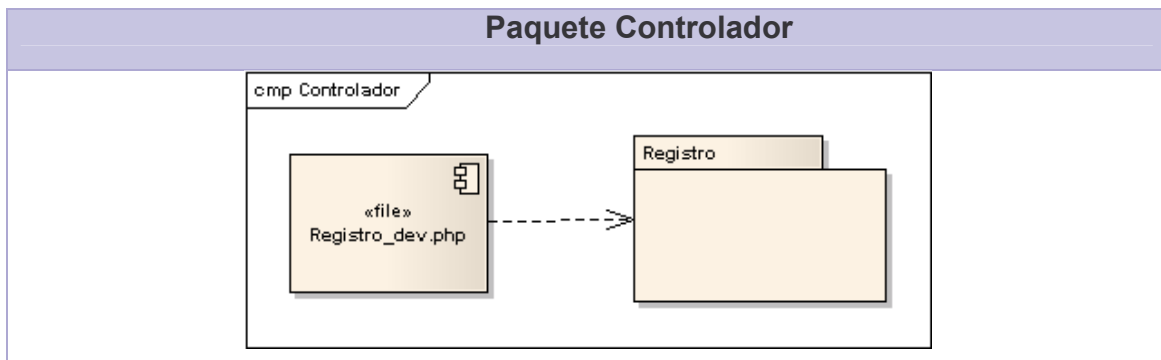


Fig. 4.5 Modelo de componentes "Controlador"

El paquete Modelo, como se aprecia en la figura 4.6, agrupa las clases del modelo en términos de componentes mostrando la relación que existe entre estas.

Paquete Modelo

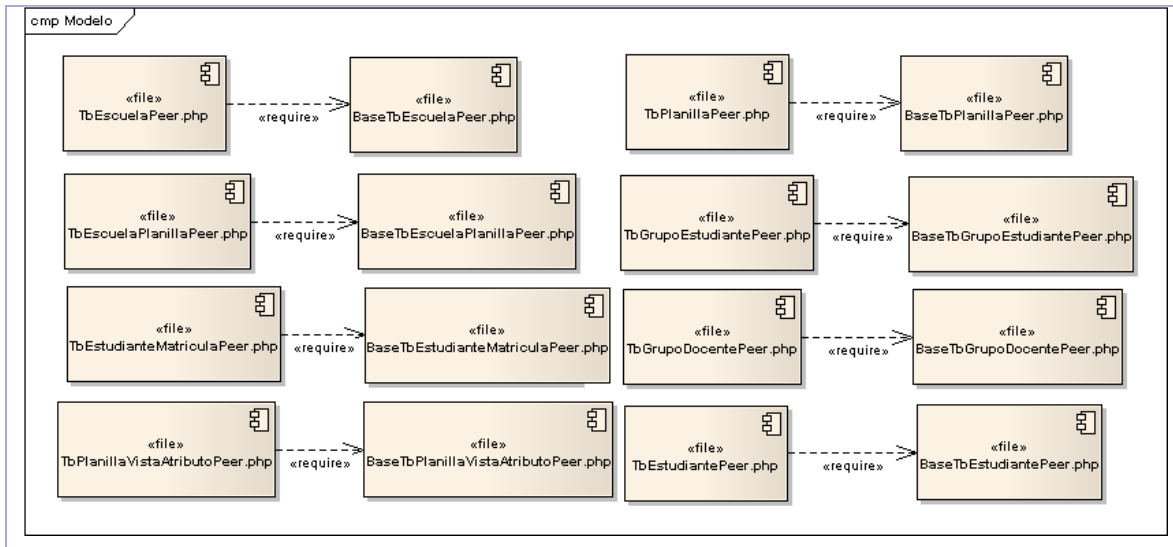
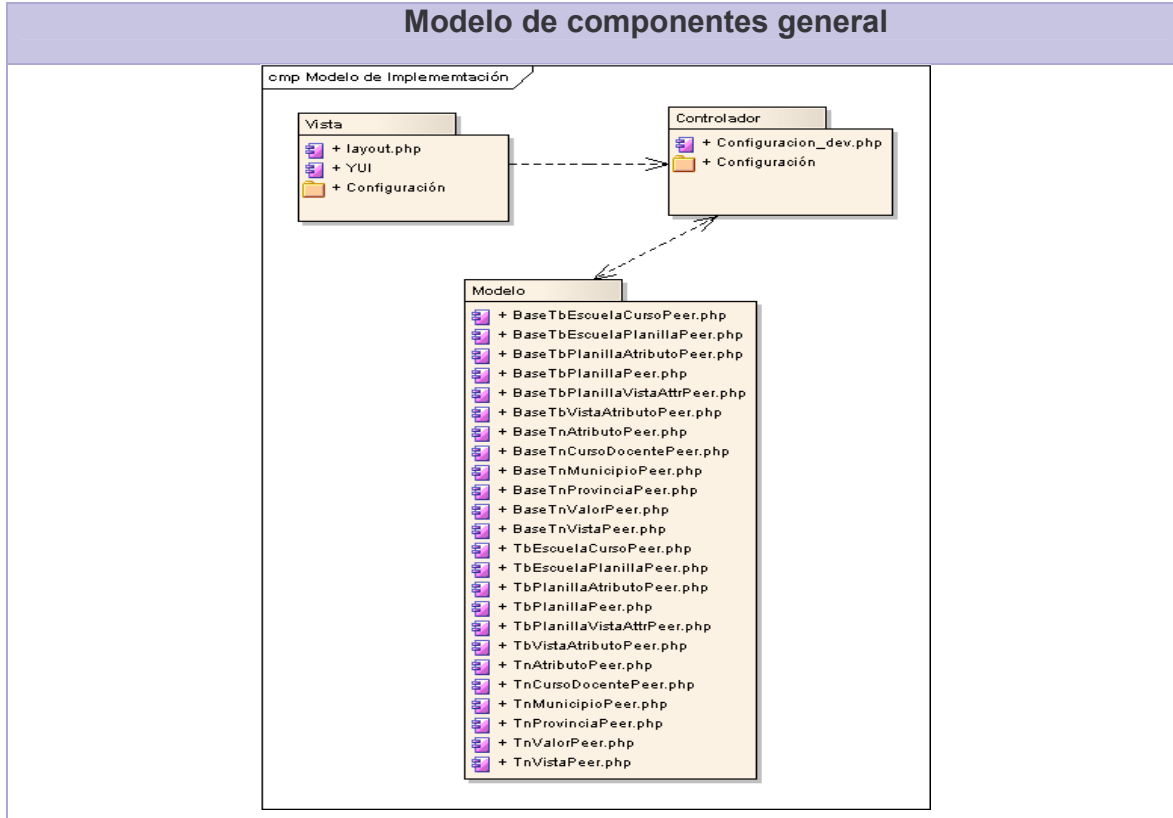


Fig. 4.6 Modelo de componentes "Modelo".

4.3.2 Paquete Configuración

La figura 4.7 muestra el modelo de implementación de forma general del paquete Configuración.



Capítulo 4: Implementación

Fig. 4.7 Modelo de implementación general “Configuración”

El paquete Vista está compuesto por los componentes que representan las librerías utilizadas para la implementación de las interfaces mostradas al usuario, las cuales están agrupadas en el paquete Configuración que se puede ver, en la figura 4.8; así como la plantilla global con el código común a todas las páginas de la aplicación.

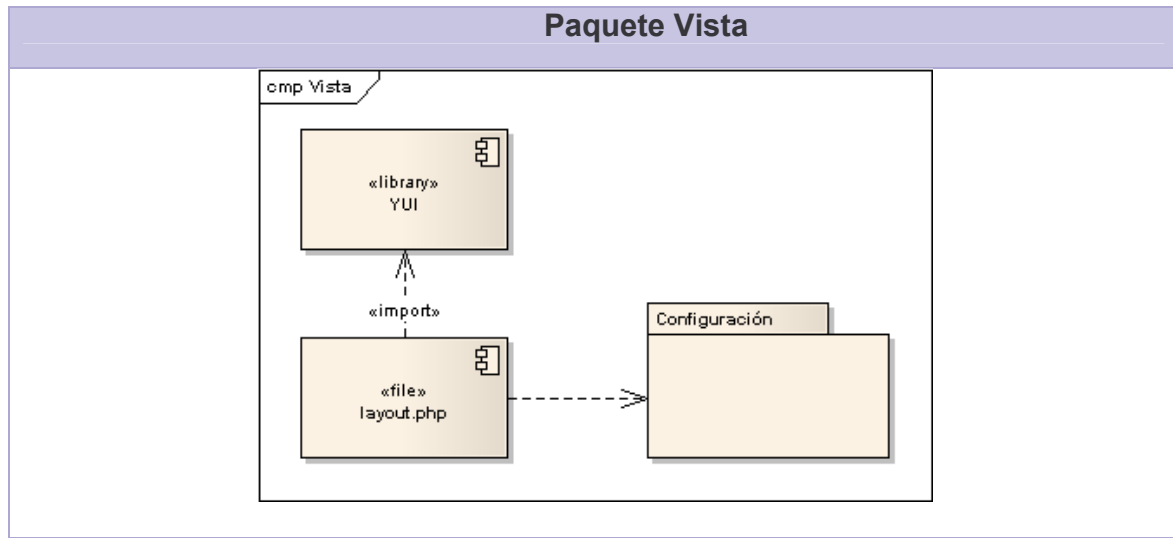


Fig. 4.8 Modelo de componentes “Vista”

En el caso del paquete Controlador, se puede ver en la figura que contiene la representación en términos de componentes tanto del controlador frontal de la aplicación como del controlador del módulo Configuración.

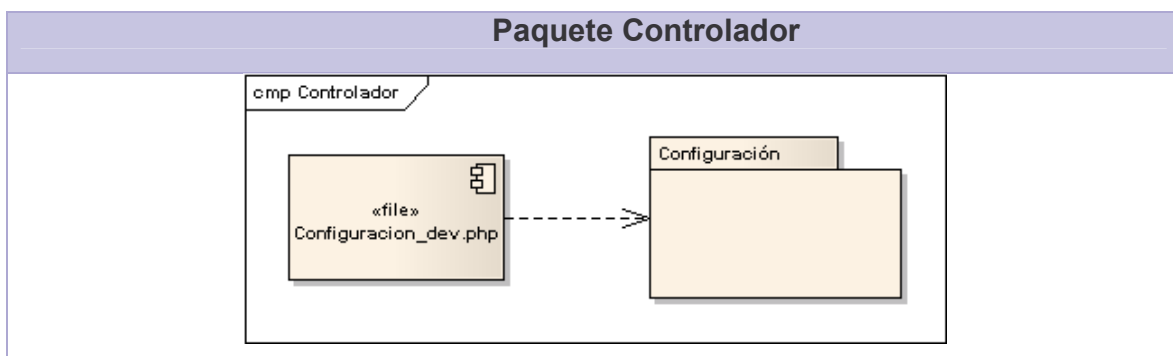


Fig. 4.9 Modelo de componentes “Controlador”

El paquete Modelo, como se aprecia en la figura 4.10, agrupa las clases del modelo en términos de componentes mostrando la relación que existe entre estas.

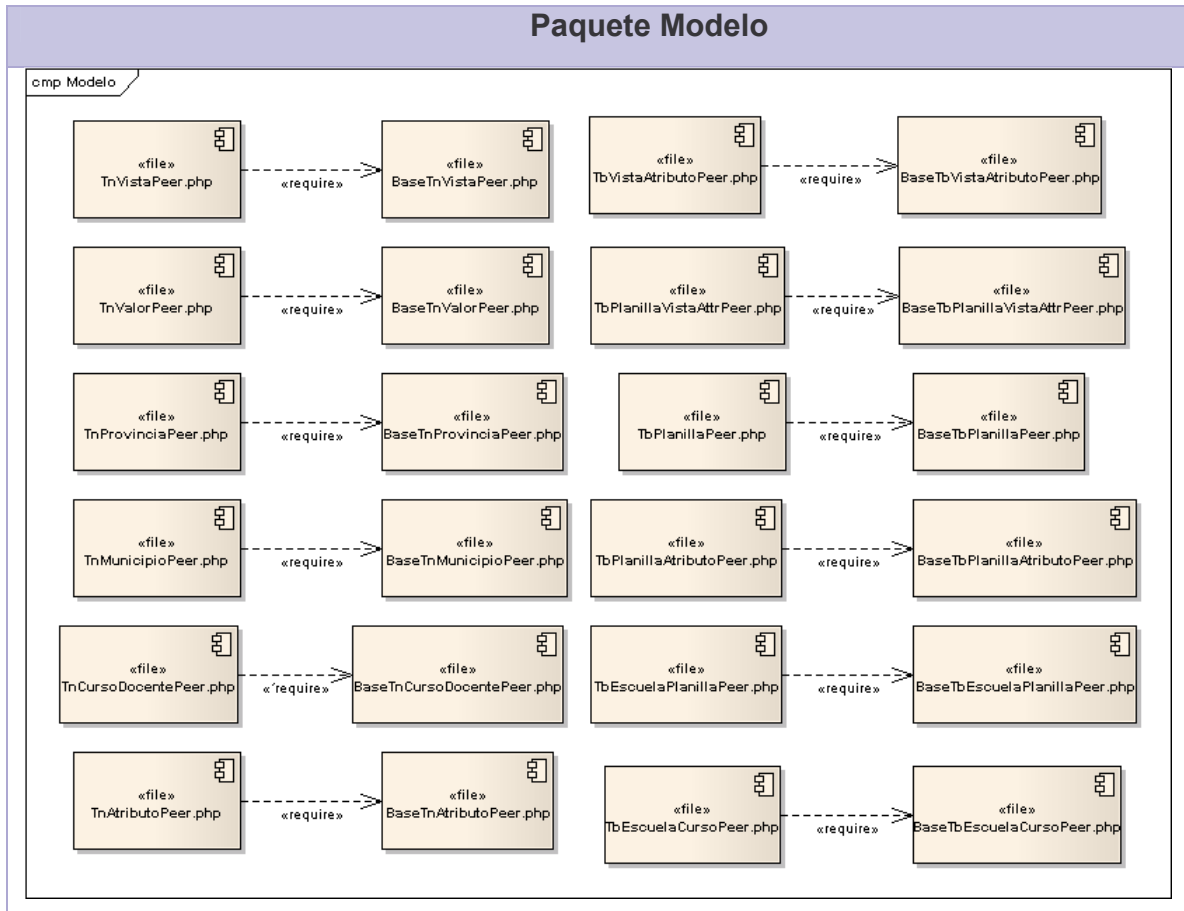


Fig. 4.10 Modelo de componentes “Modelo”.

4.3.4 Diagramas de componentes del paquete Registro

Seguidamente, se muestran los diagramas de componentes de cada caso de uso del paquete Registro.

Diagrama de componentes

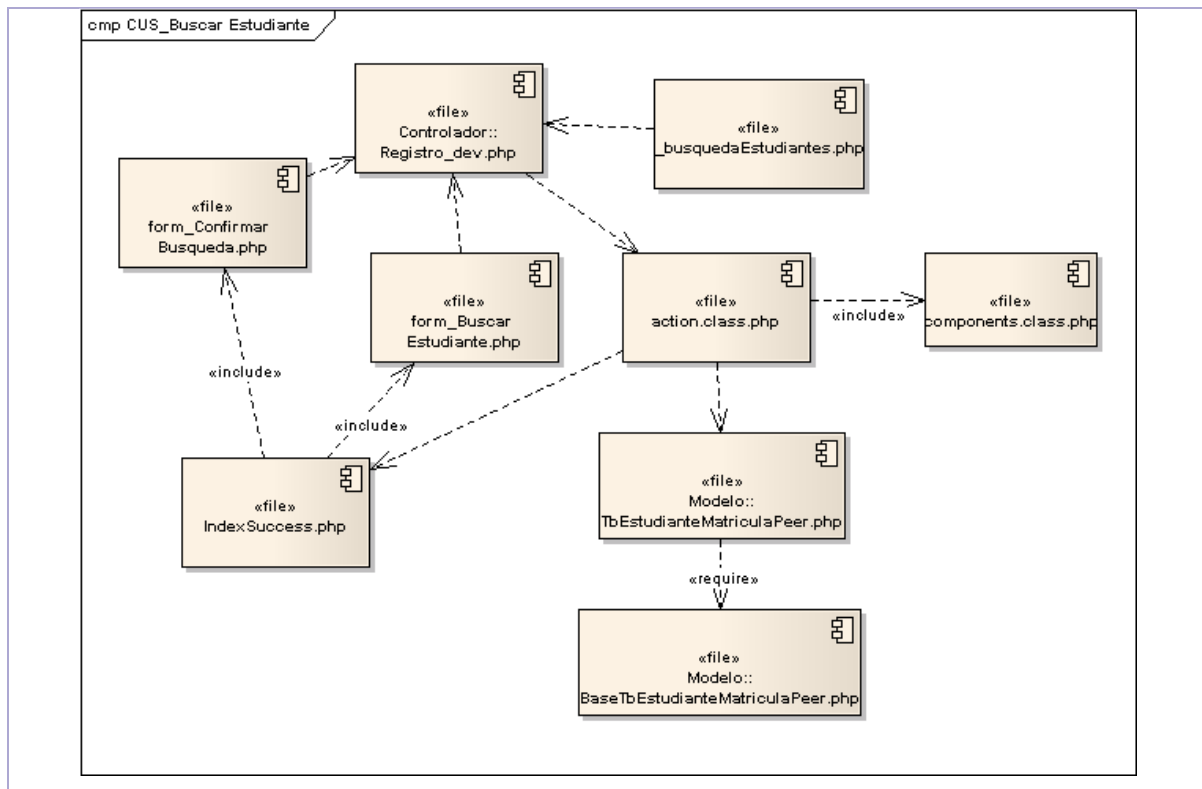


Fig. 4.12 Diagrama de componentes "Buscar Estudiante".

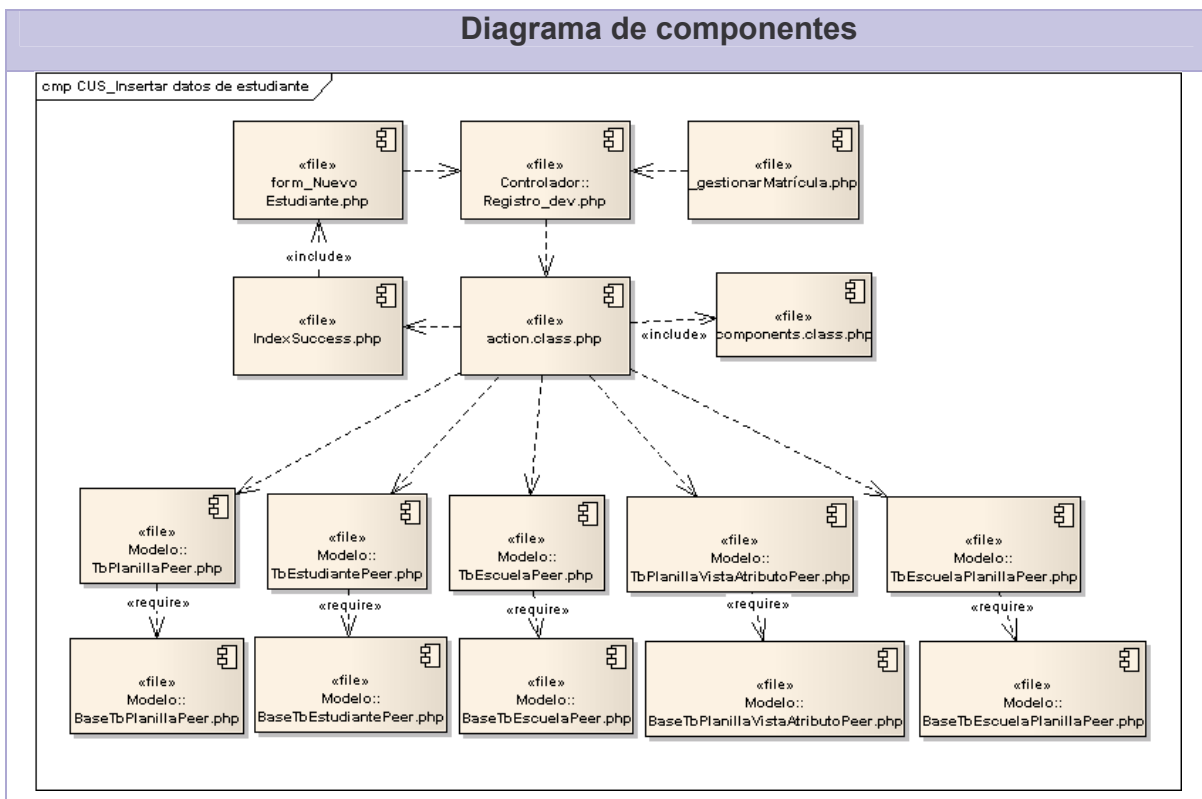


Fig. 4.13 Diagrama de componentes “Insertar datos de Estudiante”.

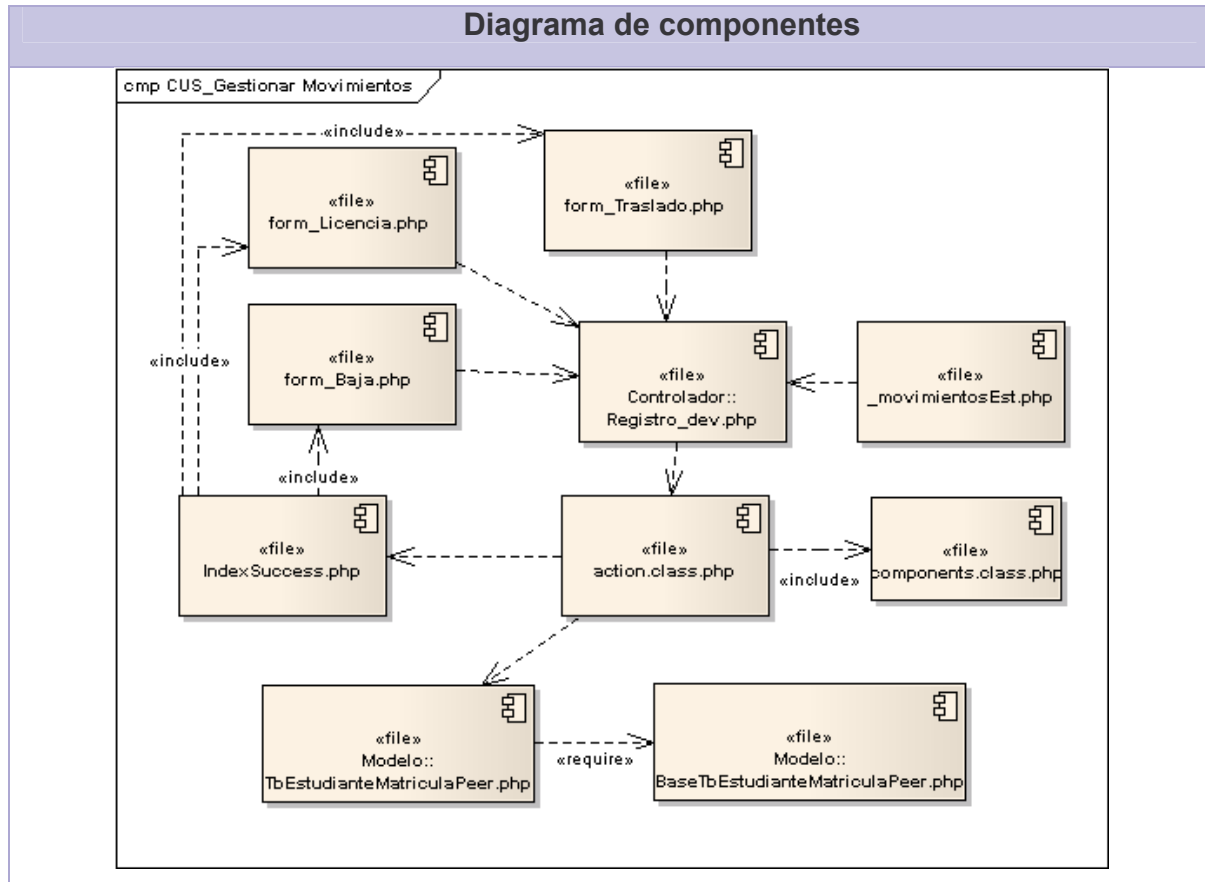


Fig. 4.14 Diagrama de componentes “Gestionar Movimientos”.

4.3.4 Diagramas de componentes del paquete Configuración

A continuación, se muestran los diagramas de componentes de cada caso de uso del paquete Configuración.

Diagrama de componentes

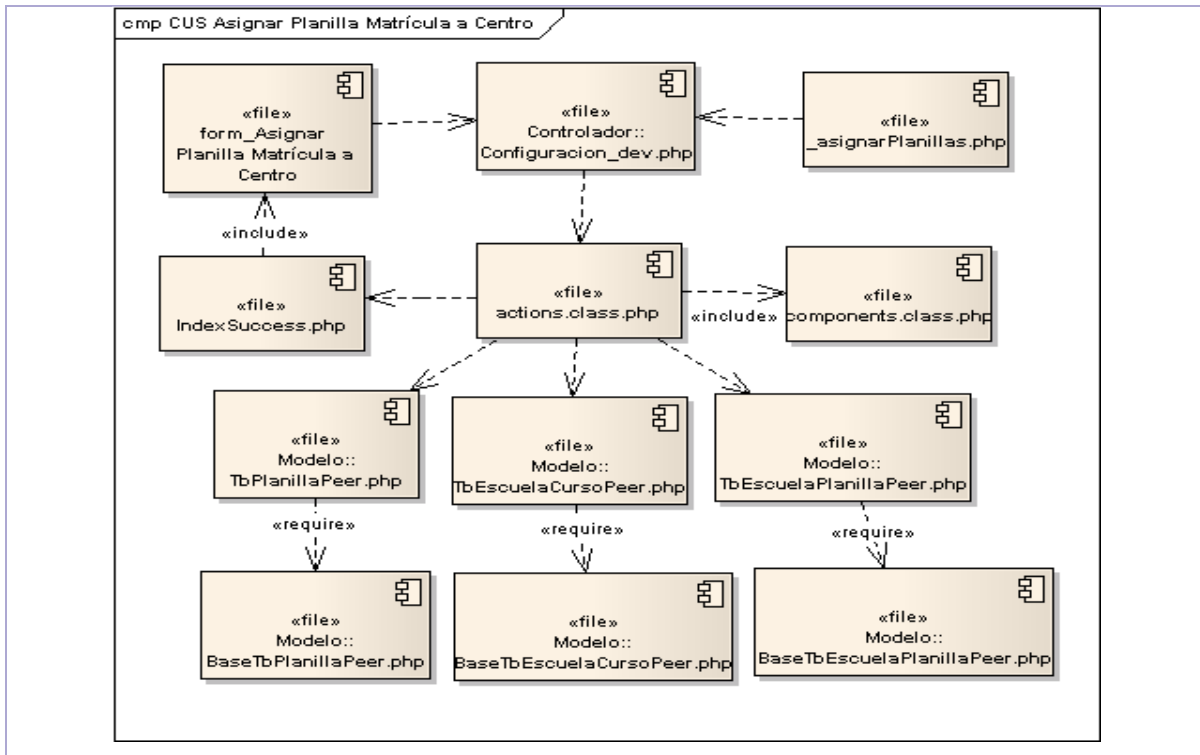


Fig. 4.15 Diagrama de componentes "Asignar Planilla Matrícula a Centro".

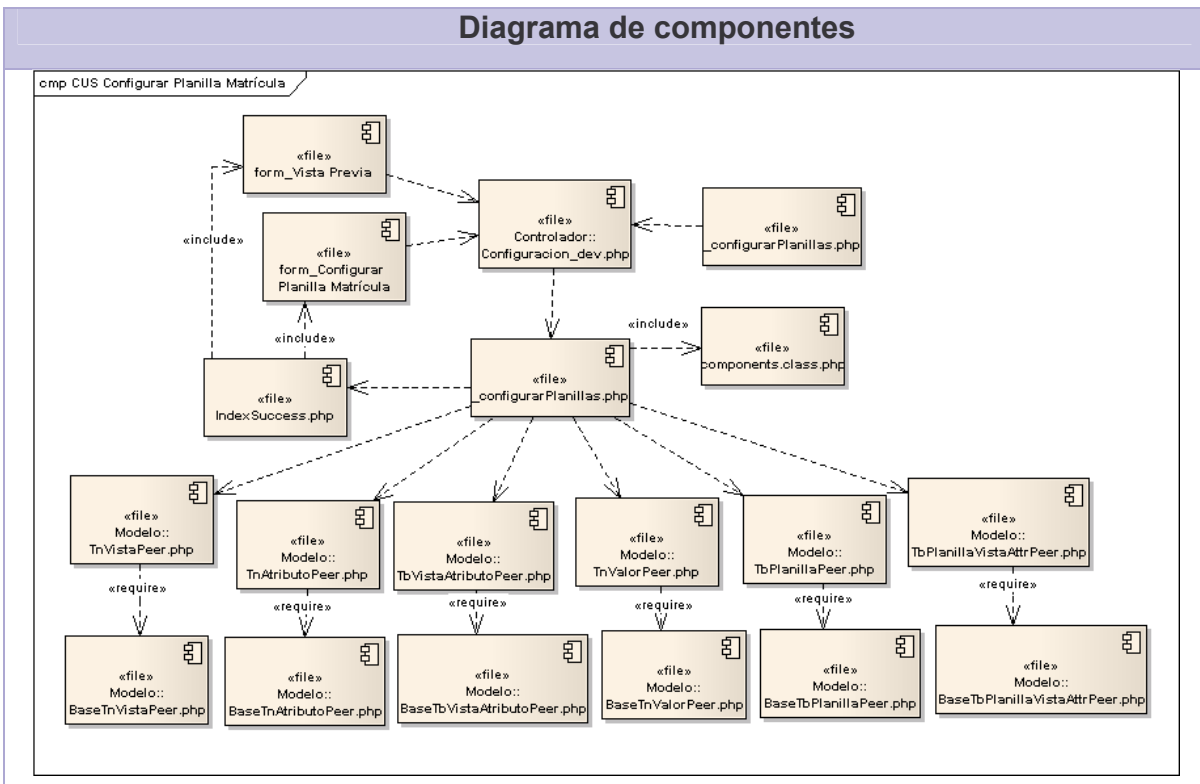


Fig. 4.16 Diagrama de componentes "Configurar Planilla Matrícula".

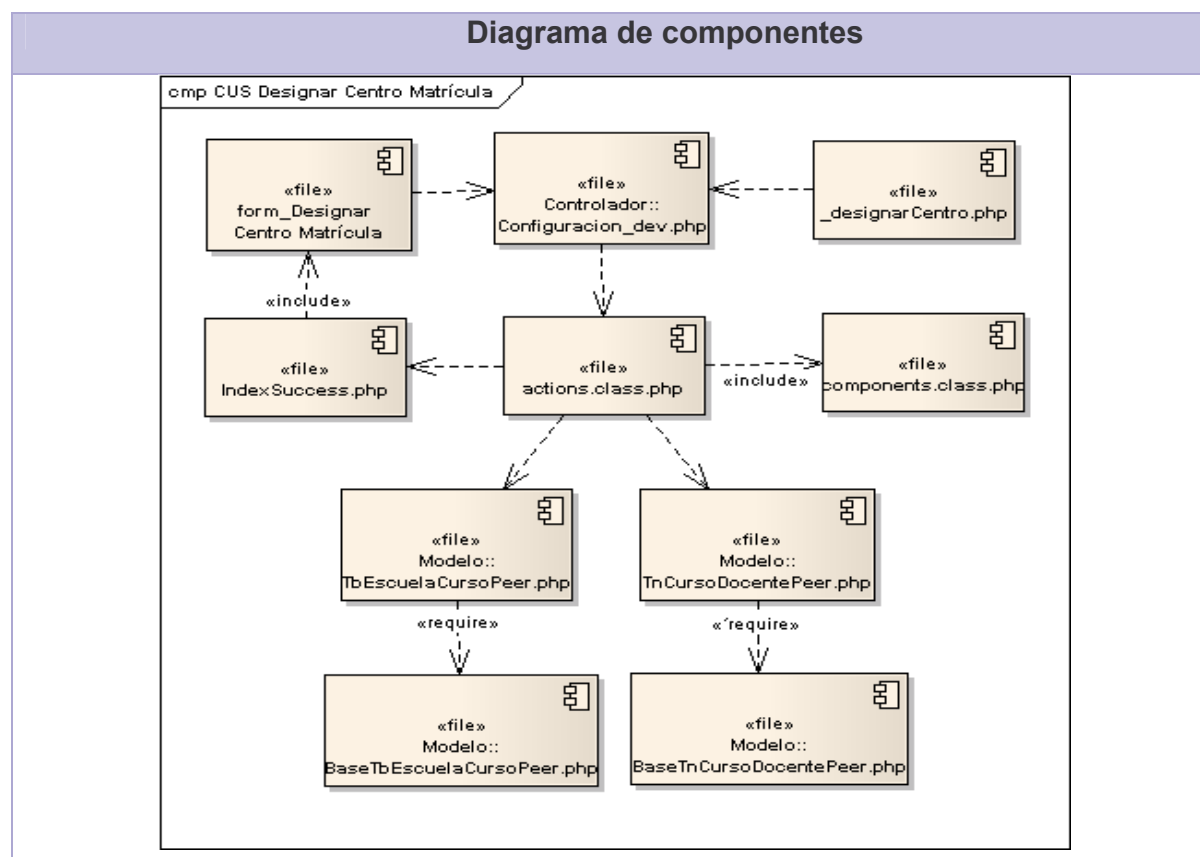


Fig. 4.17 Diagrama de componentes “Designar Centro Matrícula”.

4.4 Integración con otros sistemas

El MRE para su funcionamiento necesita consumir servicios de otros componentes de Sistema de Información de la Salud debido a que la mayor parte de la información necesaria para gestionar los procesos que el componente automatiza ya se gestionan en estos componentes. A continuación se mencionan estos componentes.

➤ Sistema de Autenticación, Autorización y Auditoría (SAAA)

SAAA es un componente que tiene implementado un mecanismo de seguridad basado en el modelo de Autenticación, Autorización y Auditoría. La autenticación es la primera acción del usuario en el sistema y consiste en suministrar un nombre de usuario único y una contraseña. Si el usuario no se encuentra registrado se reportará un error de acceso. Si el usuario que se va a autenticar se encuentra registrado, se autoriza su acceso y se crea un certificado digital que contiene un identificador único (token) de 32 caracteres. Este porta como información: el identificador del usuario, el nivel de acceso, el

identificador de nivel de acceso y el tipo de acceso que será conferido a cada uno de ellos.

La autenticación, autorización y seguridad en el Módulo Registro de Estudiantes se gestionó a través del SAAA, para garantizar el acceso de los diferentes usuarios que se identificarán con la aplicación una vez en funcionamiento.

➤ **Registro de Ciudadano (RC)**

Este es un componente que almacena la información de las personas que son registradas por la Oficina de Carné de Identidad. Entre ellas pueden ser mencionados: datos generales, datos de nacimiento, dirección, datos de los padres. La integración con Registro de Ciudadanos se realizó con el fin de homologar este sistema externo con el MRE. Utilizando el método `Buscar_Total_Ciudadano` que pertenece a este componente, se obtienen los datos de los ciudadanos que permitirán al Editor Sede seleccionar el ciudadano que será matriculado.

➤ **Registro de Unidad de Salud (RUS)**

Los servicios de Registro de Unidad de Salud son consumidos con el objetivo de designar unidades de salud. Estas van a tener matrícula en un curso docente determinado. Además asignan planillas de matrícula a dichas unidades del país. Utilizando el método `Buscar_Total_Entidad` que pertenece a este componente, se obtienen las unidades de salud que permitirán al Usuario Avanzado seleccionar la deseada.

➤ **Registro de Ubicación (RU)**

La necesidad de integración con Registro de Ubicación es similar a la del Registro de Unidad de Salud, pero en los niveles de provincia y municipio. Este registro gestiona el flujo de información relacionado con la creación, modificación o eliminación de las provincias, municipios, localidades, calles y manzanas. Haciendo el uso del método `Listar_Provincias` se obtiene un listado de todas las provincias con sus correspondientes municipios, el cual será utilizado para el proceso de ubicación de los estudiantes.

Este capítulo del documento reveló las generalidades derivadas de la “construcción” del MRE, en los diagramas y artefactos previstos y estandarizados. Se proponen las necesidades para el despliegue de la solución y poder pasar a la fase de pruebas para la

Capítulo 4: Implementación

liberación del producto software. Además se exponen las relaciones que tienen la solución con otros sistemas informáticos, así como la dependencia de esta con otros componentes.

CONCLUSIONES

Con la ejecución de este trabajo se ha cumplido el objetivo y las tareas de la investigación:

- Se diseñó e implementó el Módulo Registro de Estudiantes en su versión 1.1 cumpliéndose con las pautas de arquitectura y metas trazadas para el mismo.
- Se desarrolló el Módulo Registro de Estudiantes utilizando el framework PHP Symfony y con las tecnologías asociadas al mismo, respetando los principios de soberanía tecnológica, primicias estimuladas por el país, para el desarrollo de productos de software.
- La metodología de desarrollo utilizada aportó un marco disciplinado y ordenado durante el desarrollo de la solución, quedando modelados y debidamente registrados todos los artefactos de la ingeniería del software con el objetivo de continuar con el de desarrollo de futuras versiones de la misma.
- La solución entregada es escalable y configurable, además se integra con los demás componentes que se han estado desarrollando para el Sistema de Gestión de la Información en el Proceso de Formación de Recursos Humanos en la Salud y con otras soluciones existentes en el Sistema de Información de la Salud.

RECOMENDACIONES

A partir de los resultados de esta investigación y la experiencia adquirida durante el desarrollo, se recomienda:

- Implementar los reportes estadísticos propios del Módulo Registro de Estudiantes.
- Implementar el versionado de las planillas de matrícula.
- Gestionar las brigadas docentes, teniendo en cuenta la promoción de los estudiantes.
- Poner en práctica el plan de capacitación necesario para que el personal de salud de las diferentes unidades asistenciales se familiarice con el sistema.

REFERENCIAS BIBLIOGRÁFICAS

1. Coto, Jani; Arias, Jorge; Cruz, Aimet. Componente Registro de Estudiantes. Ciudad de la Habana : s.n., 2008.
2. Oficina de Cooperación Universitaria. [En línea] [Citado el: 8 de febrero de 2009.] <http://www.ocu.es/portal/page/portal/inicio/solucion/academico>.
3. Universidad de los Andes, Venezuela. [En línea] [Citado el: 8 de Febrero de 2009.] <http://www.forest.ula.ve/forest/ore.htm>.
4. Unixest. XesCampus. [En línea] [Citado el: 8 de Febrero de 2009.] http://www.microsoft.com/spain/enterprise/perspectivas/numero_7/educacion.msp.
5. Pagés, Daniel; Egaña, Howard; Alonso, José Luís; Sotolongo, Yumar. *SIGENU, Sistema de Gestión de la Nueva Universidad*. Ciudad de la Habana : s.n., 2008.
6. Alonso, Reinier. *Sistema de Gestión de Información en el Proceso de Formación de Recursos Humanos en Salud*. Ciudad de la Habana : s.n., 2007.
7. ¿Qué es Internet? [En línea] [Citado el: 17 de Febrero de 2009.] <http://www.arues.com/queesinternet.htm>.
8. ¿Cómo funciona Internet? [En línea] [Citado el: 17 de Febrero de 2009.] <http://www.webnova.com.ar/articulo.php?recurso=359>.
9. La Web. [En línea] [Citado el: 17 de Febrero de 2009.] <http://www.learnthenet.com/spanish/web/010www.htm>.
10. ¿Arquitectura de Software? [En línea] [Citado el: 17 de Febrero de 2009.] <http://siona.udea.edu.co/~aoviedo/Arquitectura%20de%20Software/Arquitectura%20de%20Software.htm>.
11. Carro, Elena. SOA, arquitectura orientada a servicios. [En línea] [Citado el: 17 de Febrero de 2009.] [http://md2.dei.inf.uc3m.es:8000/PA/Practicas/Exposiciones%20\(2006\)/SOA.ppt](http://md2.dei.inf.uc3m.es:8000/PA/Practicas/Exposiciones%20(2006)/SOA.ppt).
12. Servicios Web. [En línea] [Citado el: 20 de Febrero de 2009.] <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.
13. Idem a la referencia 10.

Referencias Bibliográficas

14. XML. [En línea] [Citado el: 20 de Febrero de 2009.] <http://www.w3c.es/divulgacion/guiasbreves/tecnologiasXML> .
15. Protocolo Simple de Acceso a Objetos. [En línea] [Citado el: 20 de Febrero de 2009.] <http://www.desarrolloweb.com/articulos/1557.php> .
16. WSDL. *Servicios Web*. [En línea] [Citado el: 20 de Febrero de 2009.] http://www.cibernetia.com/manuales/servicios_web/4_wsdl.php.
17. Masadelante.com. *¿Qué significa HTTP?*. [En línea] [Citado el: 20 de Febrero de 2009.] <http://www.masadelante.com/faqs/que-significa-http>.
18. Maestrodelsweb.com. *Lenguajes de programación Web*. [En línea] [Citado el: 21 de Febrero de 2009.] <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
19. Idem a la referencia 18.
20. Idem a la referencia 18.
21. Idem a la referencia 18.
22. SPARX systems. [En línea] [Citado el: 21 de Febrero de 2009.] http://www.sparxsystems.com.au/resources/uml2_tutorial/index.html.
23. Ajax: un nuevo acercamiento a las aplicaciones web. [En línea] [Citado el: 21 de Febrero de 2009.] <http://www.maestrosdelweb.com/editorial/ajax/>.
24. Qué es el Rational Unified Process. [En línea] [Citado el: 21 de Febrero de 2009.] <http://ipst08.wordpress.com/2008/08/22/%C2%BFque-es-rational-unified-process-rup/>.
25. Sistema Gestor de Bases de Datos . [En línea] [Citado el: 21 de Febrero de 2009.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
26. PostgreSQL. [En línea] [Citado el: 25 de Febrero de 2009.] http://wiki.postgresql.org/wiki/FAQ#What_is_PostgreSQL.3F_How_is_it_pronounced.3F_What_is_Postgres.3F.
27. Conceptos básicos del Servidor Web. [En línea] [Citado el: 25 de Febrero de 2009.] http://www.cibernetia.com/manuales/instalacion_servidor.
28. ¿Qué es un servidor web Apache? [En línea] [Citado el: 25 de Febrero de 2009.] <http://www.misrespuestas.com/que-es-un-servidor-web.html>.

Referencias Bibliográficas

29. Idem a la referencia 1.
30. Idem a la referencia 1.
31. SQLManager.net. [En línea] [Citado el: 25 de Febrero de 2009.]
<http://sqlmanager.net/products/postgresql/manager>.
32. Embarcadero ER/Studio. [En línea] [Citado el: 25 de Febrero de 2009.]
http://www.taringa.net/posts/downloads/884761/Modelado-de-Base-de-Datos-,-ER-Studio-6_0.html.
33. Pizzolato, Davide. *Guía del usuario de Enterprise Architect de Sparx Systems*. CXImage.
34. Idem a la referencia 1.
35. Potencier, Fabien y François, Zaninotto. *Symfony, la guía definitiva*. 2008.
36. Idem a la referencia 35.
37. Idem a la referencia 35.
38. Idem a la referencia 35.

BIBLIOGRAFÍA

Ajax: un nuevo acercamiento a las aplicaciones web. [En línea] [Citado el: 21 de Febrero de 2009.] <http://www.maestrosdelweb.com/editorial/ajax/>.

Alonso, Reinier. *Sistema de Gestión de Información en el Proceso de Formación de Recursos Humanos en Salud*. Ciudad de la Habana : s.n., 2007.

Arquitectura de Software [En línea] [Citado el: 17 de Febrero de 2009.] <http://siona.udea.edu.co/~aoviedo/Arquitectura%20de%20Software/Arquitectura%20de%20Software.htm>.

Carro, Elena. SOA, arquitectura orientada a servicios. [En línea] [Citado el: 17 de Febrero de 2009.] [http://md2.dei.inf.uc3m.es:8000/PA/Practicas/Exposiciones%20\(2006\)/SOA.ppt](http://md2.dei.inf.uc3m.es:8000/PA/Practicas/Exposiciones%20(2006)/SOA.ppt).

Colectivo. *Fase de Inicio. Flujo de Análisis y Diseño. Modelo de Análisis*.2008. <http://eva.uci.cu>.

Cómo funciona Internet [En línea] [Citado el: 17 de Febrero de 2009.] <http://www.webnova.com.ar/articulo.php?recurso=359>.

Conceptos básicos del Servidor Web. [En línea] [Citado el: 25 de Febrero de 2009.] http://www.cibernetia.com/manuales/instalacion_servidor.

Coto, Jani; Arias, Jorge; Cruz, Aimet. *Componente Registro de Estudiantes*. Ciudad de la Habana : s.n., 2008.

Craig, Larman. *UML y Patrones*. Upper Saddle River,EUA : Prentice Hall, 1999.

Embarcadero ER/Studio. [En línea] [Citado el: 25 de Febrero de 2009.] http://www.taringa.net/posts/downloads/884761/Modelado-de-Base-de-Datos--,--ER-Studio-6_0.html.

Hernández, Rolando Alfredo; González, Sayda. *El paradigma cuantitativo de la investigación científica*. 2002. ISBN: 959-16-0343-6.

JACOBSON, Ivar; RUMBAUGH, James y BOOCH, Grady, *“El proceso unificado de desarrollo”*.2000. Addison Wesley.

La Web. [En línea] [Citado el: 17 de Febrero de 2009.] <http://www.learnthenet.com/spanish/web/010www.htm>.

Bibliografía

- Masadelante.com. *Qué significa HTTP*. [En línea] [Citado el: 20 de Febrero de 2009.] <http://www.masadelante.com/faqs/que-significa-http>.
- MaestrodelaWeb.com. *Lenguajes de programación Web*. [En línea] [Citado el: 21 de Febrero de 2009.] <http://www.maestrodelaWeb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
- Oficina de Cooperación Universitaria. [En línea] [Citado el: 8 de febrero de 2009.] <http://www.ocu.es/portal/page/portal/inicio/solucion/academico>.
- Pagés, Daniel; Egaña, Howard; Alonso, José Luís; Sotolongo, Yumar. *SIGENU, Sistema de Gestión de la Nueva Universidad*. Ciudad de la Habana : s.n., 2008.
- Pizzolato, Davide. *Guía del usuario de Enterprise Architect de Sparx Systems*. CXImage.
- PostgreSQL. [En línea] [Citado el: 25 de Febrero de 2009.] http://wiki.postgresql.org/wiki/FAQ#What_is_PostgreSQL.3F_How_is_it_pronounced.3F_What_is_Postgres.3F.
- Potencier, Fabien y François, Zaninotto. *Symfony, la guía definitiva*. 2008.
- Protocolo Simple de Acceso a Objetos. [En línea] [Citado el: 20 de Febrero de 2009.] <http://www.desarrolloweb.com/articulos/1557.php>.
- ¿Qué es el Rational Unified Process? [En línea] [Citado el: 21 de Febrero de 2009.] <http://ipst08.wordpress.com/2008/08/22/%C2%BFque-es-rational-unified-process-rup/>.
- ¿Qué es Internet? [En línea] [Citado el: 17 de Febrero de 2009.] <http://www.arues.com/queesinternet.htm>.
- ¿Qué es un servidor web Apache? [En línea] [Citado el: 25 de Febrero de 2009.] <http://www.misrespuestas.com/que-es-un-servidor-web.html>.
- Servicios Web. [En línea] [Citado el: 20 de Febrero de 2009.] <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.
- Sistema Gestor de Bases de Datos . [En línea] [Citado el: 21 de Febrero de 2009.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
- SPARX systems. [En línea] [Citado el: 21 de Febrero de 2009.] http://www.sparxsystems.com.au/resources/uml2_tutorial/index.html.

Bibliografía

SQLManager.net. [En línea] [Citado el: 25 de Febrero de 2009.]
<http://sqlmanager.net/products/postgresql/manager>.

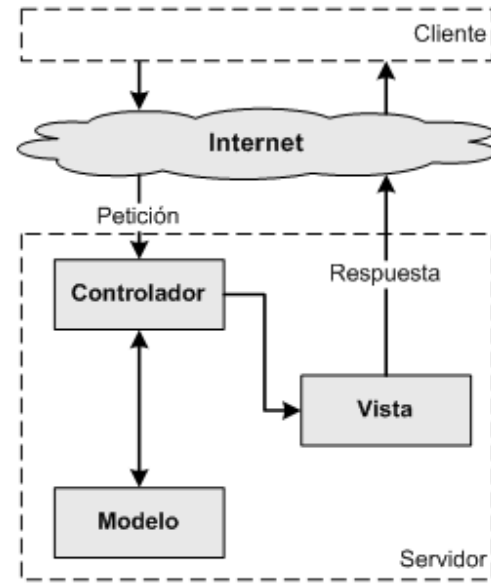
Universidad de los Andes, Venezuela. [En línea] [Citado el: 8 de Febrero de 2009.]
<http://www.forest.ula.ve/forest/ore.htm>.

Unixest. XesCampus. [En línea] [Citado el: 8 de Febrero de 2009.]
http://www.microsoft.com/spain/enterprise/perspectivas/numero_7/educacion.mspix.

WSDL. *Servicios Web*. [En línea] [Citado el: 20 de Febrero de 2009.]
http://www.cibernetia.com/manuales/servicios_web/4_wsdl.php.

XML. [En línea] [Citado el: 20 de Febrero de 2009.]
<http://www.w3c.es/divulgacion/guiasbreves/tecnologiasXML>.

ANEXOS



Anexo 1, patrón Modelo Vista Controlador

Anexo 2, pantalla del CU "Buscar Estudiante"

Planilla de Matrícula:

Datos generales | **Datos Personales 2** | Datos Familiares

Año académico *

Año estudiante *

Tipo curso << Seleccione >> *

Estudio << Seleccione >> *

84020606178					
84020606178	Carlos	Acosta	Montero		M
85120698745	Carlos	Águila	Gonzalez		M
85041587453	Carlos	Del Castillo	Gonzalez		M
85062487452	Carlos	Rojas	Barrios		M

Derechos Reservados UCI

Anexo 3, pantalla del CU "Insertar datos de Estudiante"

Movimientos de estudiantes

Búsqueda de estudiantes | **Movimientos estudiante a brigada** | Baja | Traslado | Licencia

Datos del estudiante

CI/DNI

Nombre

Primer apellido

Segundo apellido

Sexo << Seleccione >>

Selección | Cancelar

Derechos Reservados UCI

Anexo 4, pantalla del CU "Gestionar Movimientos"

Configurar planillas de matrícula

Datos Personales | **Datos Personales 2** | Datos Familiares | Datos Dirección

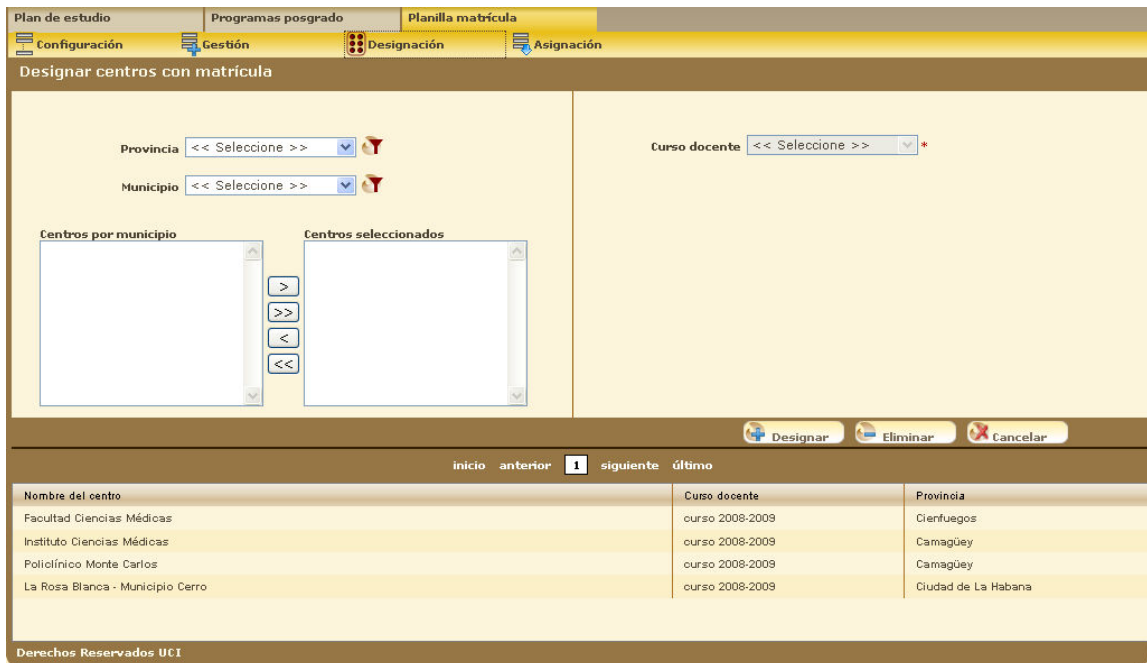
Atributos configurables | Atributos seleccionados | Atributos no configurables

Nombre planilla * | Tipo planilla << Seleccione >> *

Vista | Adicionar | Cancelar | Nueva

Derechos Reservados UCI

Anexo 5, pantalla del CU "Configurar Planilla Matrícula".



Anexo 6, pantalla del CU "Designar Centro Matrícula".



Anexo 7, pantalla del CU "Asignar Planilla Matrícula a Centro".

GLOSARIO

Apache: Es un servidor Web de tecnología Open Source (código abierto). Representa el complemento perfecto para el desarrollo de páginas dinámicas con PHP y MySQL. Puede ser instalado tanto sobre Linux o sobre Windows.

Aplicación Web: Son aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor Web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores Web (HTML, JavaScript, Java, etc.) en la que se confía la ejecución al navegador.

Casos de Usos (CU): es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

Diagramas: Combinación de varios elementos gráficos. En el Lenguaje Unificado de Modelado (UML) la finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo.

Framework: En el desarrollo de software, es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado.

Homogeneidad: Ausencia de diferencias o partes distinguibles en la materia o en un grupo de elementos.

Escalabilidad: Es la propiedad deseable de un sistema, una red o un proceso, que indica su habilidad para, o bien manejar el crecimiento continuo de trabajo de manera fluida, o bien para estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos. También se podría definir como la capacidad del sistema informático de cambiar su tamaño o configuración para adaptarse a las circunstancias cambiantes.

HTML: Lenguaje basado en marcas que indican las características del texto, utilizado para definir documentos de hipertexto en páginas Web.

HTTP: Hypertext Transfer Protocol/ Protocolo de Transferencia de Hipertextos. Modo de comunicación para solicitar páginas Web.

Hardware: Componentes electrónicos, tarjetas, periféricos y equipo que conforman un sistema de computación y se distinguen de los programas (software) porque son tangibles.

INFOMED: Es el Portal de la Salud Cubana y la red de personas e instituciones que comparten el propósito de facilitar el acceso a la información de salud en Cuba.

Informática: disciplina que estudia el tratamiento automático de la información utilizando dispositivos electrónicos y sistemas computacionales.

Informatizar: Aplicar la informática en un negocio, servicio, organismo, etc.

Internet: Sistema de redes de computación ligadas entre sí, con alcance mundial, que facilita servicios de comunicación de datos como registro remoto, transferencia de archivos, correo electrónico y grupos de noticias. Internet es una forma de conectar las redes de computación existentes que amplía en gran medida el alcance de cada sistema participante.

JavaScript: Es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web.

MINSAP: El Ministerio de Salud Pública es el Organismo rector del Sistema Nacional de Salud. Encargado de dirigir, ejecutar y controlar la aplicación de la política del Estado y del Gobierno en cuanto a la salud pública, el desarrollo de las ciencias médicas y la industria médica farmacéutica.

MVC: Modelo Vista Controlador.

Nodos procesadores clientes: Ordenadores que serán utilizados por los usuarios del sistema para acceder a la aplicación y operar la misma.

PHP: Hypertext Preprocessor/Preprocesador de Hipertexto. Es un lenguaje script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y scripts. Con el objetivo de crear aplicaciones potentes.

PostgreSQL: es un Sistema de Gestión de Bases de Datos Objeto-Relacionales libre.

PDO: PHP Data Objects es una extensión que provee una capa de abstracción de acceso a datos para PHP 5, con lo cual se consigue hacer uso de las mismas funciones para hacer consultas y obtener datos de distintos manejadores de bases de datos.

RUP: Rational Unified Process (Proceso Unificado de Desarrollo). Metodología para el desarrollo de Software.

Software: es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

Sitio Web: es un conjunto de páginas Web, típicamente comunes a un dominio de Internet o subdominio en la World Wide Web en Internet.

Symfony: Es un completo framework diseñado para optimizar el desarrollo de las aplicaciones Web mediante algunas de sus principales características.

Servidor de aplicación: Equipo en el que se instala el software servidor Web, se encargará de atender las peticiones de los usuarios del sistema.

Servidor de base de datos: Equipo en el que se instala el software gestor de bases de datos.

UML: Unified Modeling Language/Lenguaje Unificado de Modelado. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

Vice Ministerio de Docencia e Investigaciones del MINSAP: Organismo que rige la docencia e investigaciones dentro del MINSAP en Cuba.

WEB (WWW): red de documentos HTML intercomunicados y distribuidos entre servidores del mundo entero.

YUI: Es una biblioteca conformada por un conjunto de utilidades y controles, escrita en JavaScript, para la construcción de aplicaciones Web interactivas.