

Universidad de las Ciencias Informáticas

Facultad 7



Título: Componente Registro de Estudiantes

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS

Autores: Jorge Arias Sojo
Jany Coto García
Aimet Anolan Cruz Abad

Tutores: Ing. Yenisel Valido Pérez
Ing. Reinier Alonso González

Asesora: Lic. Yaima Margarita Riverí Ruiz

Ciudad de la Habana, julio de 2008

“Año 50 de la Revolución”

“Conectarnos al conocimiento y participar en una verdadera globalización de la información que signifique compartir y no excluir, que acabe con la extendida práctica del robo de cerebros, es un imperativo estratégico para la supervivencia de nuestras identidades culturales de cara al próximo siglo”. Fidel Castro (1999).

DECLARACIÓN DE AUTORÍA

Declaramos ser autores del presente trabajo 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 11 días del mes de julio del año 2008.

Jorge Arias Sojo

Autor

Jany Coto García

Autora

Aimet Anolan Cruz Abad

Autora

Ing. Yenisel Valido Pérez

Tutora

Ing. Reinier Alonso González

Tutor

DATOS DE CONTACTO

Ing. Yenisel Valido Pérez. (yenisel@uci.cu). Graduada de Ingeniería Informática en el Instituto Superior Politécnico “José Antonio Echeverría” (ISPJAE) y la Universidad de Cienfuegos “Carlos Rafael Rodríguez”. Profesora en tercer año de adiestramiento en la Universidad de las Ciencias Informáticas, pertenece al departamento de Ingeniería y Gestión de Software, imparte las asignaturas de Ingeniería de Software y se desempeña como jefa de la misma en la facultad siete. Profesora del Área Temática Sistemas de Apoyo a la Salud, líder del proyecto Docencia Médica.

Ing. Reinier Alonso González. (ralonso@uci.cu). Graduado de Ingeniero en Ciencias Informáticas en la Universidad de las “Ciencias Informáticas” en su primera graduación. Actualmente profesor en adiestramiento en el departamento de Técnicas de Programación de la Facultad 7 y Jefe del Polo de desarrollo para la Salud de la Facultad 7.

Lic. Yaima Margarita Riverí Ruiz (maggie@uci.cu): Graduada con Título de Oro de Licenciatura en Educación. Especialidad Lengua Inglesa. Posee Categoría Docente de Profesor Instructor. En cinco años de labor docente, ha impartido las asignaturas de la disciplina Idiomas Extranjeros. Es Jefa de la asignatura Idioma Extranjero IV.

AGRADECIMIENTOS

Agradecer primeramente a la Revolución y a nuestro siempre Comandante en Jefe Fidel Castro Ruz, por haber creado esta universidad de futuro y depositar en nosotros toda su confianza. Esperamos no defraudarlo.

Agradecer a nuestra familia por todo el apoyo brindado y por confiar en nosotros.

Agradecer especialmente a nuestros padres por saber guiarnos siempre por el camino correcto y por su apoyo en los momentos difíciles de nuestras vidas.

Agradecer a todos nuestros amigos y compañeros, que de una forma u otra nos han brindado su ayuda en los momentos que los hemos necesitado, en especial a Pepe “el amore”, Yandy, Fonseca, Michel, Yutmila, Carlitos,

Agradecer a todos los profesores que durante estos años nos han transmitido sus conocimientos y experiencias para formarnos como futuros profesionales, especialmente a Yenisel, Tiuská, Yaima por aguantar nuestras malcriadeces y soportar tantas noches en vela, Yasel por transmitirnos sus ideas y confiar en nosotros para la realización de este trabajo, Karel por ser uno más en nuestro equipo y a Reinier.

Agradecer a nuestro tribunal por ayudarnos a culminar satisfactoriamente con el Trabajo de Diploma.

DEDICATORIA

Quiero dedicar este trabajo a las dos personitas más especiales que tengo en la vida, que me han acompañado en los momentos difíciles y felices, y me han guiado por el camino correcto de la vida, a mi mamá y mi papá, “Mimi” y “Papi”, los quiero mucho.

A mi novio “Tata” que aunque no le gusta que le diga así, sé que lo sabe aceptar, que me ha aguantado como soy, con mis errores y aciertos, ¡Tata te amo!

Se lo dedico también a mis abuelitas del alma, que sé que cada una a su forma me quiere mucho, “Mami Albertina” aquí tienes a tu nieta mongoliana, espero que este logro sea un orgullo mas para ti, “Mima Sonia” este es tu regalo de cumpleaños de mi parte. A mis abuelos que aunque a uno no lo conocí y de mi “abue Eduardo” que disfruté muy poco, sé que estarían muy orgullosos de mi. Al resto de mi familia, que los quiero mucho pero necesitaría un libro para mencionarlos.

A todos mis amigos, en especial a Lisy, Cathy y Joe, a mi gran amiga Leisy y su esposo, a los nuevos amigos de Santiago que en tan poco tiempo han sabido ganarse mi cariño, a mi amigo y guía “Jorge”.

A mis compañeros de tesis “Jorge y Aimet”: Jorge que ha sido un magnifico compañero de tesis y también es mi novio, juntos hemos pasando largas noches en vela y compartiendo profesional y sentimentalmente momentos difíciles en estos meses de trabajo. Gracias, mi amor. A Aimet, que supo ser una maravillosa compañera de trabajo.

A mis dos tutores Yenisel y Reinier que nos han soportado en estos meses. A mi asesora Yaima que nos soportó horas enteras en su casa y por entendernos siempre.

De Jany.

Dedico este trabajo a las personas más importantes en mi vida, a mi mamá, a mi papá, a mi hermano Javier, a mis abuelos que los quiero mucho, y de manera especial a mis abuelitas, que de donde quiera que estén las tengo presente. Y a mi familia en general que de una forma u otra se han preocupado y me han ayudado mucho.

A mis amigos de Santiago que son los mejores del mundo, a mis tutores Yenisel y Reinier, y en especial a mi novia que la quiero mucho y que me ha enseñado a ser responsable, que ha estado en todo momento a mi lado, para entenderme y ayudarme.

A mis compañeros de aula, en especial a Ale y Pepe, a mis compañeras de tesis Jany y Aimet por brindarme todo su apoyo en estos meses de sacrificio y trabajo.

De Jorge.

Quisiera dedicarle especialmente este trabajo mis abuelos que a pesar de no estar conmigo en estos momentos, sé que hubieran querido ver este gran día.

A mis padres Alicia y Alberto por ser los mejores padres que uno pueda tener y estar siempre dispuestos a ayudarme cada vez que los he necesitado, por apoyarme y guiarme en la vida, gracias.

A mi hermanita linda Arlhiubys que a pesar de ser tan peleona, siempre me da sus consejos y me transmite sus experiencias. A mis otras hermanas, Eliuska y Eliza, por saber que puedo contar con ellas.

A toda mi familia, temo dejar a alguien fuera, que siempre confiaron en mí y me ayudaron a llegar a ser quien soy hoy.

A mi novio Víctor por apoyarme durante todos estos años y querer siempre lo mejor para mí, gracias por tanto amor y dedicación.

A mis amigos con los cuales disfruté muchos momentos lindos de la carrera, en especial a mi querido equipo de fútbol sala, que nunca sentiré algo igual como cuando entrenábamos juntas. A Ariel por formar además de una profesional, una deportista más preparada y segura. A Yunelsis (Yune), por tus buenísimos consejos sobre la vida y otras experiencias que a tu lado aprendí, ¿increíble verdad? A Baby, por sacarme de varios apuros, y ayudarme con todo.

A todos mis profesores, que durante la carrera han sabido enseñarme, no solo conocimientos sino también lo que es vivir con conocimientos.

Y por último, a mis compañeros de tesis Jany y Jorge, que sin ellos nunca hubiera podido lograr el sueño de realizar este trabajo y hacerlo bien.

De Aimet.

RESUMEN

Las secretarías docentes en los centros clínicos educacionales de Cuba cuentan con un sistema Web compuesto por varios módulos, que tiene una base de datos centralizada y puede ser consultada desde cualquier lugar del país. Aunque en ocasiones el sistema se sobrecarga y el trabajo en las secretarías se hace más lento. Además, el sistema no es configurable y es poco flexible, por ello se propone un nuevo diseño de software que sea configurable y que cumpla con los requisitos de flexibilidad ante cambios en la estructura que los organiza.

Este trabajo tiene como objetivo mejorar la gestión de la información relacionada con el registro de estudiantes en el proceso de formación de recursos humanos en el MINSAP, mediante el desarrollo de un componente orientado a servicios Web, basado en el módulo Matrícula del Estudiante realizado en el Área Temática Sistemas de Apoyo a la Salud. El componente se desarrolla sobre tecnologías Web (PHP, Linux, MySQL, Apache, Motor de plantillas Smarty, Ajax, XML), con el uso de herramientas de software libre.

Este componente es configurable, permite obtener los datos según las necesidades específicas de los diferentes centros. Así mismo, brinda servicios Web que pueden ser utilizados por otros sistemas del sector de la salud. La tarea resulta de gran importancia ya que mejora el trabajo de las secretarías docentes y facilita el proceso de la formación de recursos humanos en este sector.

PALABRAS CLAVE

Formación de Recursos Humanos, Sistema, Componente, Servicios Web, Software, Flexible.

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
1.1 Sistemas existentes en el país que se relacionan con la formación de profesionales de la salud.	6
1.2 Tecnologías y tendencias actuales.....	7
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	29
2.1 Conceptos asociados al dominio del problema.....	29
2.2 Objeto de estudio	29
2.3 Propuesta del sistema.....	30
2.4 Modelo de Dominio	30
2.5 Especificación de los requerimientos de software	32
2.6. Diagrama de Casos de Uso del sistema.....	36
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	50
3.1 Análisis.....	50
3.2 Diseño.....	54
3.3 Descripción de las clases.....	61
3.4 Diseño de la BD de Registro de Estudiante.....	71
3.5 Diseño de la BD de Nomencladores.....	77
3.6 Definiciones de Diseño.....	88
3.7 Tratamiento de errores.....	89
3.8 Estándar de Codificación	90
CAPÍTULO 4: IMPLEMENTACIÓN	91
4.1 Diagrama de despliegue	91
4.2 Diagrama de componentes	92
4.3 Publicaciones de Servicios Web.....	98
4.4 Integración con otros sistemas.....	103
CONCLUSIONES	105
RECOMENDACIONES	106
REFERENCIAS BIBLIOGRÁFICAS	107
BIBLIOGRAFÍA.....	109
ANEXO 1: MODELO DE DOMINIO.....	111
ANEXO 2: INTERFACES DEL COMPONENTE.....	112
GLOSARIO	114

INTRODUCCIÓN

Desde el último decenio del siglo XX, la sociedad cubana enfrenta una situación que ha obligado a reflexionar sobre las transformaciones conceptuales. Estas demandan nuevas formas de abordar el estudio del panorama multidimensional de las ciencias médicas en la actualidad. Lograr la estabilidad en el terreno de la salud, se ha convertido en una meta que redundará en el mejoramiento y en el bienestar de las personas. Estas concepciones llegan a los umbrales del siglo XXI con los logros del pasado, pero identificando los objetivos del futuro; para lo cual la función de la información es decisiva.

En estos momentos el proceso para obtener la información ha cambiado, pues existen métodos que permiten compartir, transmitir y comunicar mensajes en breve tiempo. Esta es la *Era de la Información* o más conocida por todos como la *Era de la Tecnología*. Con relación ya hace más de 40 años, el Comandante Ernesto Che Guevara, expresó en 1962, *“el mundo camina hacia la electrónica... todo indica que esta ciencia se constituirá en algo así como una medida del desarrollo; quien la domine será un país de vanguardia...”*.

Actualmente esto ha tenido una transformación vertiginosa ya que la tecnología permite recopilar, analizar, almacenar y recuperar información a velocidades y costos inimaginables. Las computadoras y la comunicación han revolucionado la estrategia de informatización en muchas sociedades, incluyendo el importante sector de la salud. La introducción de estas tecnologías como herramientas de trabajo, en este sector ha sido paulatina; aunque no cabe duda, que han elevado y desarrollan los niveles de salud a nivel mundial.

Cuba enfrenta el reto de informatizar la sociedad con el propósito de poder integrarse con plenitud a la infraestructura de la información y así hacer uso de las nuevas tecnologías. El sector de la salud no se ha quedado atrás con dicha inclusión y desarrolla la informática médica.

Hoy en día, con el avance de las tecnologías aparece el concepto de base de datos, y con estas metodologías de diseño y tratamiento, a partir de que se podían ofrecer mayores posibilidades de almacenaje de información. De esta forma ocupa menos espacio y disminuye considerablemente el tiempo de búsquedas y el tratamiento de los datos. Junto con ellas surgen además, arquitecturas de desarrollo que rompen con todos los esquemas de realización de aplicaciones o sistemas pequeños. Esto da paso al desarrollo de aplicaciones Web que permiten y facilitan el flujo de información y las comunicaciones. La aparición de dichas tecnologías posibilita que se puedan desarrollar sistemas con

grandes volúmenes de información. Entre ellos se pueden mencionar los que se desarrollan para manejar los procesos relacionados con las actividades docentes en una institución.

Actualmente el proceso de Formación de Recursos Humanos se ha convertido para Cuba en un gran desafío. Su principal reto está encaminado a la dinámica en la formación de profesionales de la salud o personal médico calificado. Así como a la aparición y aceptación de nuevos programas de estudio, debido a la necesidad que va teniendo el plan de carreras de Ciencias Médicas, de cambiar sus enfoques tradicionales.

Este proceso tiene lugar en las secretarías docentes de las instituciones clínicas cubanas del MINSAP (Ministerio de Salud Pública), siendo estas la base donde se mantiene y conforma la principal información referente al proceso. Para este fin hasta 1999 se utilizó el sistema SAIDO (Sistema Automatizado de Información Docente); construido por el Centro de Cibernética Aplicada a la Medicina (CECAM), para manejar los datos del proceso.

En el año 2000, el sistema SAIDO -que aún se utilizaba en algunas instituciones- comenzó a tener dificultades debido al llamado “efecto 2000”, además de que no cumplía con las transformaciones actuales de la enseñanza superior del sector de salud. [1]

En este sistema no se considera el control de la marcha del proceso docente de los extranjeros, que deben hacer la nivelación académica en las facultades médicas. Por consiguiente se realizó un software en Microsoft Access, que consiste en una base de datos encaminada a mostrar informes sobre los resultados docentes de los estudiantes de la enseñanza premédica, que ingresan al país para equilibrar el nivel académico. Las aplicaciones Access antes mencionadas, eran soluciones locales de cada una de las provincias, que respondían a sus necesidades. Esto implicaba que cada centro docente contaba con su propio método para gestionar dicha información. [2]

Para darle solución a lo antes planteado, surge un sistema centralizado que permite agrupar toda la información del proceso en la aplicación *Sistema de Gestión de Información en el Proceso de Formación de Recursos Humanos en la Salud*. El mismo, basado en nuevas tecnologías y software libre, resuelve los problemas que existían en el proceso de formación del estudiante de esta esfera. Todos los centros clínicos docentes presentan una uniformidad en cuanto a trabajo se refiere; además de que toda la información se almacena en una única base de datos que puede ser consultada desde cualquier lugar del país.

A pesar de que este sistema resuelve los problemas en las secretarías docentes, expuestos inicialmente, surgen nuevas dificultades, por ejemplo: la base de datos se sobrecarga y en ocasiones demora el trabajo, pues tiene que recoger mucha información o cuando trata de consultar la base de datos. Además, ante cambios que puedan surgir respecto a los datos de matrícula, la aplicación es dependiente de los desarrolladores del software.

Este sistema cuenta con un módulo que gestiona el registro de los estudiantes que inician su formación en las diferentes carreras de Ciencias Médicas. Este módulo carece de una particularidad fundamental y necesaria en estos momentos en el MINSAP, que consiste en la configuración de la planilla de matrícula -que pueda ser asignada con los campos necesarios predefinidos por un usuario determinado- y en el control a nivel nacional de los centros que realizan matrícula. Por los problemas antes mencionados y por la necesidad que presenta la red médica de un rápido flujo de datos de los estudiantes y una reusabilidad del código en otros sistemas, la investigación va a estar enmarcada en dicho módulo, para proponer y desarrollar así, una solución a las nuevas necesidades.

Por la situación anteriormente planteada, el **problema a resolver** de esta investigación consiste en ¿Cómo mejorar la gestión de la información docente, que se maneja en un sistema para la gestión de los procesos de formación de recursos humanos en salud?

El **objeto de estudio** se centra en el proceso de gestión de la información docente en las secretarías de los centros educacionales del MINSAP.

El **campo de acción** se enmarca en el proceso de gestión de la información relacionada con el registro de estudiantes en las secretarías de los centros educacionales del MINSAP.

Para dar solución al problema expuesto se propone como **objetivo general**: Mejorar la gestión de la información relacionada con el registro de estudiantes en el proceso de formación de recursos humanos en el MINSAP, mediante el desarrollo de un componente orientado a servicios Web.

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

- Analizar el estado del arte sobre sistemas que gestionan la información de estudiantes.
- Asimilar las tecnologías (PHP, XML, XML Web Services AJAX, Smarty, MySQL, XHTML, JavaScript) y arquitectura (Orientada a Servicios y Basada en Componentes) definidas por el Área Temática Sistemas de Apoyo a la Salud (SAS) para la implementación de la solución.
- Examinar los servicios públicos de los Componentes RC (Registro de Ciudadanos), US (Unidades de Salud), SAAA (Sistema de Autenticación, Autorización y Auditoría), del Sistema de Información para la Salud, para establecer la integración requerida con los mismos.
- Desarrollar los artefactos correspondientes a los flujos de trabajo del Proceso Unificado de Desarrollo: “Modelado del Negocio”, “Requerimientos”, “Análisis y Diseño” e “Implementación” para el componente.
- Proponer un diseño flexible para el componente orientado a servicios, basado en estándares y patrones.
- Implementar el componente basado en estándares abiertos (XML).
- Implementar un componente que permita la configuración de los nomencladores necesarios para el componente a desarrollar, proporcionando XML Web Services de estos.
- Desarrollar XML Web Services públicos con sus respectivas descripciones WSDL para otros componentes desarrollados por el Área Temática SAS, y otros del Sistema de Información para la Salud.

Esta investigación cuenta con cuatro capítulos donde se realiza el análisis y fundamentación de la solución que se propone y abarca las tecnologías y herramientas que son usadas para el desarrollo del componente orientado a servicios Web y que incluye el modelo del negocio, el análisis y 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 de análisis y diseño, además de los diagramas de interacción en el diseño. También se muestra el diagrama de base de datos, donde se realiza la descripción de sus tablas y campos.

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.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se realiza un análisis del estado del arte y la situación actual de los sistemas para el proceso de formación de recursos humanos, en las secretarías docentes de los centros de salud a nivel nacional. También se estudian los conceptos, tecnologías y herramientas que servirán para el desarrollo del Componente Registro de Estudiantes según las necesidades y características de los distintos centros de salud.

1.1 Sistemas existentes en el país que se relacionan con la formación de profesionales de la salud

Las secretarías docentes de las facultades del país, cuentan entre otros, con el SAIDO (*Sistema Automatizado de Información Docente*). Este fue uno de los primeros sistemas diseñados e implementados por el CECAM (Centro de Cibernética Aplicada a la Medicina). A partir de su explotación se han tenido que incorporar, a otras versiones, nuevos módulos y subsistemas. También se han tenido que elaborar otros sistemas que se relacionan con él, en correspondencia con las necesidades de los usuarios y el avance de las tecnologías de la computación. [3]

El SAIDO, que está implantado a nivel de facultad y de instituto en todo el país es poco flexible a las transformaciones que ha sufrido la educación médica en los últimos tiempos. Estas razones dieron paso a que se desarrollaran diversas soluciones dentro de las mismas instituciones.

Las soluciones antes mencionadas fueron trabajadas bajo el entorno de aplicaciones escritorio. Estas fueron desarrolladas mayormente en Visual Basic y Delphi, utilizando como gestores de base de datos Microsoft Access y en algunos de los casos, llegando a incorporar SQL Server. Esto trajo mayores consecuencias, ya que se vieron limitadas la flexibilidad y funcionalidad ante futuros cambios y nuevas actualizaciones que se le agregaran a cada sistema. Uno de los principales problemas que se enfrentaban era la necesidad de cambiar y actualizar los sistemas en cada una de las diferentes secretarías docentes del país. Esto limitaba el acceso de los niveles superiores de dirección, a la información real y actualizada.

Para mejorar la gestión de la formación de recursos humanos se desarrolló un sistema basado en tecnología Web, llamado Sistema de Gestión de Información en el Proceso de Formación de Recursos Humanos en Salud. Su desarrollo tuvo lugar en la UCI (Universidad de las Ciencias Informáticas). El mismo está basado en la arquitectura Cliente Servidor de Tres Capas (Capa de Acceso a Datos, Capa

Lógica de Negocio y Capa Interfaz de Usuario), utiliza como gestor de base de datos MySQL. Dentro de los lenguajes de programación utilizados para un entorno Web, se utilizó del lado del cliente HTML, JavaScript, Ajax, y del lado del servidor, PHP. Como motor de plantillas se utilizó Smarty. Actualmente se encuentran en explotación, en INFOMED, los módulos de matrícula, movimientos y administración.

1.2 Tecnologías y tendencias actuales

El desarrollo de aplicaciones y soluciones para la informatización de la gestión de la información de todo tipo de proceso, se ha convertido para los desarrolladores de software en un gran reto. Esto se debe a que no sólo deben resolver con sus soluciones las necesidades de sus clientes, sino también deben alcanzar un nivel técnico acorde con el auge de la informatización. Por esta razón se hace necesario para cada situación, hacer un estudio detallado de las herramientas y tecnologías que se van a utilizar. Esto permitirá conocer las ventajas y posibilidades de desarrollo que puedan brindar.

Las tendencias actuales en el desarrollo de soluciones informáticas están encaminadas hacia la filosofía Web, producto de las ventajas que las mismas proporcionan. Dentro ellas se pueden destacar: inmediatez de acceso; las aplicaciones Web no necesitan ser descargadas, instaladas y configuradas; sólo con ser un cliente Web, puede trabajar sin importar cuál es su configuración y su hardware. [4] Se puede mencionar la alta disponibilidad que presentan las aplicaciones Web, pues se pueden consultar desde cualquier parte del mundo y a cualquier hora, con tan sólo tener acceso a Internet. Cualquier empresa puede cambiar de sistema operativo o de hardware, sin afectar el funcionamiento de las aplicaciones del servidor. Las actualizaciones al sistema se hacen en el servidor, acción que es transparente para los clientes. De esta manera se evita que usuarios con poca experiencia cometan errores al intentar la actualización. [5]

1.2.1 Internet. Su funcionamiento

Internet puede definirse como un amplio sistema de redes de ordenadores que están interconectadas. Es un conjunto integrado por diferentes redes de los países del mundo, donde cualquier usuario que tenga acceso, obtiene desde cualquier computador, la información que se publica. También pueden utilizar los recursos de otra computadora o contactar con otro usuario que esté en un ordenador que se encuentre completamente distante al suyo. Actualmente Internet es un medio que se utiliza para la comunicación pública, cooperativa y autosuficiente en términos económicos, que puede ser accedido por millones de personas de todo el mundo. [6]

Físicamente Internet está compuesta por miles de redes independientes que pertenecen a diferentes empresas, instituciones investigativas, universidades y otras organizaciones. Para lograr la comunicación entre ellas comparten un conjunto de normas que la aseguran. Ellas son conocidas como protocolos de Internet. De acuerdo a la función del tipo de comunicación que existe entre los diferentes ordenadores se encuentran: el FTP (transferencia de ficheros), TELNET (conexión con terminales remotas), el HTTP que es la base de la telaraña mundial, que se abrevia en *WWW* (World Wide Web) o más conocida como Web, entre otros. También 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, entre otros; y que las mismas utilicen técnicamente el mismo protocolo, TCP/IP. [7]

1.2.2 La Web. Su funcionamiento

El funcionamiento de la Web sigue el llamado modelo Cliente-Servidor, el cual se ha hecho muy habitual en las aplicaciones que funcionan en una red. En él existe un servidor que presta el servicio, y un cliente que lo recibe. Una vez que el cliente está conectado a Internet, debe instalar un programa que le permita acceder a las páginas Web. Este se utiliza para leer documentos de hipertexto y se denomina navegador. Dentro de los más conocidos están el Internet Explorer de Microsoft, el Mozilla Firefox y el Netscape. [8]

1.2.2.1 Web 2.0

La Web 2.0 es la evolución que ha ocurrido desde aplicaciones tradicionales hacia aplicaciones que funcionan a través de ella, enfocadas al usuario final. Son aplicaciones que brindan colaboración y servicios que reemplazan las aplicaciones de escritorio. Para dar vida a un proyecto basado en Web 2.0 existen varias tecnologías: [9]

- Transformar software de escritorio hacia la plataforma Web.
- Respeto a los estándares del XHTML.
- Separación de contenido del diseño con uso de hojas de estilos.
- Sindicación de contenido.
- Ajax (Asincrónica JavaScript y XML).

- Uso de Flash, Flex o Lazlo.
- Uso de Ruby on Rails para programar páginas dinámicas.
- Utilización de redes sociales al manejar usuarios y comunidades.
- Dar control total a los usuarios en el manejo de su información.
- Proveer XML para que las aplicaciones puedan ser manipuladas por otros.
- Facilitar el posicionamiento con URL sencillo.

1.2.3 Estilos de Arquitectura

Un estilo es un concepto descriptivo que define una forma de articulación u organización arquitectónica. El conjunto de los estilos cataloga las formas básicas posibles de estructuras de software. Mientras que las formas complejas se articulan mediante composición de los estilos fundamentales. Los estilos sirven para sintetizar estructuras de soluciones que luego serán refinadas en el diseño y para definir los patrones posibles para una aplicación. Esto evita errores arquitectónicos y permite evaluar arquitecturas alternativas.

1.2.3.1 Arquitectura Orientada a Servicios (SOA)

Con el paso del tiempo y el avance de las tecnologías, las empresas e instituciones piden aplicaciones de software más complejas, con mayor calidad, menor presupuesto y en menor tiempo. Hoy en día, crear estas aplicaciones requiere de funcionalidades que ya estén implementadas y que forman parte de otros sistemas.

SOA (Service Oriented Architecture) como definición, proporciona una metodología y un marco de trabajo para documentar las capacidades de negocio y puede dar soporte a las actividades de integración y consolidación. Es un estilo arquitectónico y su éxito se debe a la alineación de los procesos de negocio con los sistemas de infraestructura. Se basa en estándares abiertos. Permite interoperabilidad entre distintas tecnologías. Separa la lógica de los procesos de los sistemas base, lo cual proporciona facilidad de cambio en los sistemas. Es gradual, su implementación debe hacerse planeada y paulatinamente.

En otras palabras, SOA nació como una estrategia de integración, y brinda servicios con funcionalidades bien definidas a cualquier aplicación que lo necesite. De esta forma, una aplicación final simplemente se nutre de un conjunto de servicios, lo que ofrece al usuario final su lógica particular y le muestra una interfaz. Esta arquitectura presenta varias ventajas: [10]

- **Reducción del tiempo de desarrollo:** los servicios son fácilmente reusables y pueden ser ensamblados rápidamente. Por su reusabilidad los tiempos de desarrollo se reducen grandemente, pues los servicios que ya están implementados en una aplicación pueden ser utilizados en otras sin necesidad de tener que programarlos nuevamente.
- **Reducción de los costes de desarrollo y mantenimiento:** la reducción del tiempo de desarrollo lleva consigo la disminución del coste. También los servicios reducen la complejidad interna de las aplicaciones, pues la funcionalidad está separada en bloques más pequeños. Esto provoca que se reduzca el coste y el tiempo de mantenimiento de las mismas.
- **Más flexible y escalable:** el crecimiento o el cambio en las funcionalidades de una institución no afecta a las aplicaciones de forma completa. Solamente basta con añadir o cambiar los módulos de servicios necesarios.
- **Más ágil:** por el débil acoplamiento entre las interfaces de los distintos servicios se consiguen aplicaciones más ágiles.
- **Interoperabilidad de aplicaciones:** la complejidad en el proceso de integración disminuye, ya que se interactúa con elementos que se abstraen de la tecnología y la ubicación de los servicios.

1.2.3.1.1 Servicios Web

Sobre los Servicios Web existen varias definiciones, por lo que es muy complejo dar una adecuada definición que abarque lo que son y lo que implican. Se puede decir que un Servicio Web es una colección de protocolos y estándares que sirven para intercambiar datos entre muchas aplicaciones. Diferentes aplicaciones de software, que hayan sido desarrolladas en distintos lenguajes de programación y ejecutados en 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.

Los servicios Web tienen amplia utilidad ya que ellos proporcionan mecanismos de comunicación estándares entre las diferentes aplicaciones, que interactúan entre sí para ofrecer información dinámica al usuario. Para que los servicios proporcionen extensibilidad e interoperabilidad entre las aplicaciones es necesaria una arquitectura de referencia estándar. [11]

Ventajas y desventajas de los Servicios Web

Los Servicios Web presentan ventajas. Entre ellas están: [12]

- Fomentan los protocolos y estándares basados en texto, que hacen acceder más fácil a su contenido y así entender su funcionamiento.
- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas en que son instalados.
- Como se apoyan en el protocolo HTTP, pueden aprovecharse de los sistemas de seguridad firewall sin tener necesidad de cambiar las reglas de filtrado.
- Permiten que software y servicios de distintas compañías que se encuentran en diferentes lugares geográficos, puedan ser combinados para facilitar servicios integrados.

Los Servicios Web también tienen algunas desventajas. Estas son: [12]

- Al apoyarse en el protocolo HTTP, pueden evadir medidas de seguridad basadas en firewall.
- Su rendimiento es bajo si se compara con otros modelos de computación distribuida.

1.2.4 Patrones de Arquitectura y Diseño

Los patrones son un intento de formalizar y plasmar de una forma práctica generaciones de conocimiento arquitectónico. No son principios abstractos que requieran su redescubrimiento para obtener una aplicación satisfactoria, ni son específicos a una situación particular o cultura. Un patrón define una posible solución correcta para un problema dentro de un contexto dado, lo que describe las cualidades invariantes de todas las soluciones. 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: de Diseño, de Arquitectura, de Análisis, de Negocio, entre otros.

1.2.4.1 Alta Cohesión y Bajo Acoplamiento

Alta Cohesión y Bajo Acoplamiento son dos conceptos que se pudieran separar pero ciertamente están ligados. Si se llegara a aumentar mucho la cohesión del sistema, se tendría entonces un alto acoplamiento entre los componentes; y de la forma contraria si se redujera el acoplamiento se vería mermada la cohesión.

Si se vieran por separado, Alta Cohesión es la información que se gestiona en un servicio determinado, es decir, que la información que almacena un componente debe ser coherente y está en mayor medida relacionada con el componente. Un ejemplo de baja cohesión son servicios que realizan demasiadas tareas. El Bajo Acoplamiento hace referencia a las relaciones que tienen los componentes 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. [13]

1.2.4.2 Modelo Vista Controlador (MVC)

MVC (Modelo Vista Controlador) es un patrón de arquitectura de software que separa los datos de una aplicación (Modelo), la interfaz de usuario (Vista) y la lógica de control (Controlador) en tres componentes distintos. El mismo se ve frecuentemente en aplicaciones Web, donde se utilicen diferentes interfaces de usuario y el código que provee de datos dinámicos a la página. A continuación se hace una breve descripción de los tres elementos fundamentales de este patrón: [14]

- **Modelo:** es la representación específica con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos, habitualmente desde el controlador.
- **Vista:** presenta el modelo en un formato adecuado para interactuar. Usualmente la interfaz de usuario.
- **Controlador:** responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Este modelo brinda determinadas ventajas, dentro de las que están: posibilidad de tener diferentes vistas para un mismo modelo; construir nuevas vistas sin necesidad de modificar el modelo subyacente; y proporciona un mecanismo de configuración a componentes complejos, mucho más tratable que el puramente basado en eventos.

1.2.4.3 Arquitectura en 3 Capas

La arquitectura de una aplicación es la vista conceptual de su estructura. La arquitectura en 3 capas es la generalización de la arquitectura Cliente/Servidor donde la carga se divide en tres partes. La aplicación contiene código de presentación (Capa de Presentación), código de procesamiento de datos (Capa de Negocio) y código de almacenamiento de datos (Capa de Datos). Esta arquitectura está soportada sobre un nivel creciente de abstracción, lo que propicia una amplia reutilización. Estos datos pueden ser utilizados por diferentes versiones de una misma capa en la medida que soporten una misma interfaz. Permite distribuir el trabajo de creación de una aplicación por niveles. [15]

- 1. Capa de Presentación o Interfaz de Usuario:** es la capa con la que interactúan los usuarios. Esta es la capa que ve el usuario. Presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso. Esta se comunica únicamente con la Capa de Negocio.
- 2. Capa de Negocio:** esta capa es para las reglas del negocio. En la misma residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se comunica con la Capa de Presentación, para recibir las solicitudes y enviar los resultados. También se comunica con la Capa de Acceso a Datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.
- 3. Capa de Acceso a Datos:** es la capa que gestiona los datos. Está formada por uno o más gestores de bases de datos, que realiza todo el almacenamiento de datos. Reciben además solicitudes de almacenamiento o recuperación de información desde la Capa de Negocio.

1.2.4.4 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, UDDI, HTTP.

1.2.4.4.1 XML (Extensible Markup Language)

XML es un lenguaje de marcas simple, que juega un papel importante en el intercambio de gran cantidad de datos. Es muy similar al HTML con la diferencia de que su función principal es describir datos y no mostrarlos como lo hace el HTML. Además, es un formato que permite la lectura de datos a través de distintas aplicaciones. XML sirve para estructurar, almacenar e intercambiar información.

Ventajas del XML: [16]

- Es extensible, esto quiere decir que una vez diseñado un lenguaje y puesto en producción, es posible extenderlo con la adición de nuevas etiquetas. De esta manera los antiguos consumidores todavía puedan entender el nuevo formato.
- El analizador es un componente estándar, por lo que no es necesario crear un analizador específico para cada lenguaje. De esta forma se evitan los bugs y se acelera el desarrollo de la aplicación.
- Es sencillo entender su estructura y procesarlo. Mejora la compatibilidad entre aplicaciones.

1.2.4.4.2 SOAP (Simple Object Access Protocol)

SOAP (Simple Object Access Protocol) es un protocolo estándar que permite la comunicación y la interoperabilidad entre diversas aplicaciones Web desarrolladas con tecnologías diferentes. Es un protocolo liviano para el intercambio de información en forma descentralizada. Es decir presenta un entorno distribuido. Además, es basado en XML y 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.

Ventajas de SOAP: [17]

- **No está asociado con ningún lenguaje:** los desarrolladores que estén vinculados a nuevos proyectos pueden elegir para desarrollar, el último y mejor lenguaje de programación que exista.

- **No se encuentra fuertemente asociado a ningún protocolo de transporte:** un mensaje SOAP es un documento XML, por lo que puede transportarse utilizando cualquier protocolo que pueda transmitir textos.
- **Permite la interoperabilidad entre múltiples entornos:** SOAP fue desarrollado sobre los estándares existentes de la industria, por lo que las aplicaciones que se ejecuten en plataformas con dichos estándares, pueden comunicarse mediante mensajes SOAP con aplicaciones que se ejecutan en otras plataformas.
- **No está atado a ninguna infraestructura de objeto distribuido:** la mayoría de los sistemas de objetos distribuidos se pueden extender para que admitan SOAP.

1.2.4.4.3 WSDL (Web Services Description Language)

WSDL (Web Services Description Language) es un formato XML que se utiliza para la descripción de los servicios Web. Describe la interfaz pública de los servicios Web. El mismo está basado en XML y describe la forma de comunicación, o sea, 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 soportan, se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje. [18]

1.2.4.4.4 UDDI (Universal, Description, Discovery and Integration of Web Services)

UDDI (Universal, Description, Discovery and Integration of Web Services) es uno de los estándares básicos de los servicios Web. Su objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL. En estos se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros. [19]

1.2.4.4.5 HTTP (Hypertext Transfer Protocol)

HTTP (Hypertext Transfer Protocol) es un protocolo del nivel de aplicación, utilizado para la transferencia de información entre sistemas de forma rápida y clara. El mismo se basa en un paradigma de peticiones y respuestas. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado.

1.2.5 Lenguajes de Programación Utilizados

Los lenguajes de programación fueron creados por el propio hombre con la intención de poder comunicarse con las máquinas. De ahí que, un lenguaje de programación es un lenguaje que se utiliza para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

Precisamente la programación Web, parte de las siglas *WWW*. Dentro de este tipo de programación existen dos grandes grupos: los lenguajes del lado del cliente y los lenguajes del lado del servidor.

En el primer grupo se pueden considerar como algunos de los más utilizados a nivel mundial: JavaScript, Visual Basic Script y XSLT, HTML. Dentro del segundo grupo, se pueden mencionar el PHP, Java, ASP y PERL. Con ellos el desarrollador implementa la lógica del negocio dentro del servidor e incluso cómo acceder a los diferentes gestores de base de datos.

1.2.5.1 Lenguajes del Lado del Cliente

- **JavaScript:** es un lenguaje interpretado, o sea, no requiere de compilación. El código JavaScript es embebido en el código HTML o en otros similares. Esto hace mucho más fácil la creación de páginas Web con contenido dinámico. El mismo fue diseñado para controlar la apariencia y manipular los eventos de los objetos que están contenidos en un formulario, también este lenguaje es soportado por la mayoría de los navegadores. Además, en cuanto a su formato y sintaxis es semejante a los lenguajes Java y C. Otro aspecto a tener en cuenta es que los programas escritos en este lenguaje tienden a ser pequeños y compactos, lo que influye en que no se requiera ni memoria ni tiempo adicional para su transmisión. [20]
- **Ajax:** acrónimo de Asynchronous JavaScript and XML (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.

Es una combinación de tres tecnologías ya existentes: [21]

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor Web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- XML es el formato usado comúnmente para la transferencia de vuelta al servidor; aunque cualquier formato puede funcionar, incluyendo HTML pre formateado, texto plano, JSON y hasta EBML.

1.2.5.1.1 Lenguajes de Marcas

Un lenguaje de marcado o lenguaje de marcas es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto y su presentación. Los lenguajes de marcado se suelen diferenciar en tres clases, aunque pueden combinarse, en la práctica, varias clases en un mismo documento, las mismas son:

- **Marcado de Presentación:** es aquel que indica el formato del texto. Este tipo de marcado es útil para maquetar la presentación de un documento para su lectura; pero resulta insuficiente para el procedimiento automático de la información. El mismo resulta más fácil de elaborar, sobre todo para cantidades pequeñas de información. Sin embargo, resulta complicado mantenerlo o modificarlo, por lo que su uso se ha ido reduciendo en proyectos grandes a favor de otros tipos de marcado más estructurados.
- **Marcado de Procedimientos:** está enfocado hacia la presentación del texto sin embargo, también es visible para el usuario que edita el texto. El programa que representa el documento debe interpretar el código en el mismo orden en que aparecen.
- **Marcado Descriptivo o Semántico:** utiliza etiquetas para describir los fragmentos de texto, pero sin especificar cómo deben ser representados, o en qué orden, lo que le brinda al mismo

una gran flexibilidad. Este tipo de marcado también simplifica la tarea de reformatear un texto, debido a que la información del formato está separada del propio contenido.

Entre los lenguajes de marca se pueden encontrar:

- **HTML** (HyperText Markup Language o Lenguaje de Marcado de Hipertexto): creado en 1989 por Tim Berners-Lee. 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. [22]
- **XML** (Extensible Markup Language o Lenguaje de Marcado Extensible): fue establecido en febrero de 1998. Es un metalenguaje que permite crear etiquetas adaptadas a las necesidades (de ahí lo de extensible). XML proporciona la flexibilidad y consistencia que no se podía alcanzar con el HTML. O sea, XML proporciona un mecanismo estándar para describir cualquier tipo de datos en documentos que son altamente trasportables y reutilizables. [23]
- **XHTML** (Extensible HyperText Markup Language o Lenguaje de Marcado de Hipertexto Extensible): fue pensado para sustituir al HTML como estándar de las páginas Web. Es la versión XML de HTML, por lo que tiene básicamente, las mismas funcionalidades, pero cumple las especificaciones más estrictas de XML. Es considerado el sucesor del HTML. [24]

1.2.5.1.2 Lenguajes del Lado del Servidor

- **PHP** (Hypertext Preprocessor): es un lenguaje de programación interpretado, utilizado fundamentalmente para la creación de páginas Web dinámicas y aplicaciones para servidores. Fue creado por Rasmus Lerdorf en el año 1994. Los scripts PHP son embebidos frecuentemente en otros códigos como el HTML. La interpretación y ejecución de estos scripts ocurren en el servidor. Los clientes que hacen la solicitud desde un navegador solamente reciben la respuesta y nunca ven el código PHP. Este lenguaje permite la conexión a distintos servidores de bases de datos como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite. Corre en múltiples plataformas, de ahí que es un lenguaje multiplataforma.

Este lenguaje presenta un conjunto de ventajas, entre las que se destacan: [25]

- Es un lenguaje multiplataforma, o sea, que puede ser ejecutado en la mayoría de los sistemas operativos de la actualidad y que puede interactuar con los servidores Web más populares.
- Capacidad de conexión con la mayoría de los manejadores de base de datos en la actualidad.
- Es fácil de aprender debido a su simplicidad.
- Es un software libre, lo que lo convierte en una alternativa de fácil acceso para todos.
- Su biblioteca nativa de funciones es amplia e incluida.
- Permite las técnicas de programación orientada a objetos.

A pesar de sus ventajas también destacan varias dificultades, por ejemplo, por sus características promueve la creación de código desordenado y complejo de mantener. Por su diseño dinámico no puede ser compilado y es muy difícil de optimizar. Además, por ser todo su trabajo realizado en el lado del servidor y sin delegar responsabilidades a los clientes, puede ser que en momentos determinados la capacidad de respuesta sea ineficiente, en la medida en que las solicitudes al servidor aumenten considerablemente.

1.2.5.2 Motor de Plantilla

El mecanismo de plantillas es utilizado para separar el código de presentación del resto, en una aplicación Web. Consiste en la codificación de todo lo que tenga que ver con la presentación en una serie de plantillas con código HTML. Presenta expresiones sencillas intercaladas para comunicarse con la otra parte de las aplicaciones y así poder mostrar datos dinámicos.

El motor de plantilla utilizado para el desarrollo del sistema es el Smarty. Se encuentra bajo la licencia LGPL, por lo que puede ser utilizado libremente. Este es un motor de plantillas para PHP, que separa el código PHP como lógica de negocios, del código HTML como lógica de presentación. Genera contenido Web mediante la colocación de etiquetas Smarty en un documento. Es rápido, se pueden crear funciones habituales, y modificadores de variables personalizadas, por lo que este motor de plantilla es altamente extensible.

1.2.6 Sistema Gestor de Base de Datos

Un sistema gestor de base de datos es un tipo de software que se dedica a servir de interfaz entre la base de datos, los usuarios y las aplicaciones que lo utilizan. También es considerado como un conjunto de programas que permiten crear y mantener una base de datos; lo que asegura su integridad, confidencialidad y seguridad. Por lo que debe permitir: definir una base de datos, construir la base de datos y manipular la misma. Para que un sistema gestor de base de datos sea deseable, debe tener varias características: [26]

- **Control de la redundancia:** este tiene varios efectos negativos (duplicar el trabajo al actualizar, desperdicia espacio en disco, puede provocar inconsistencia de datos) aunque a veces es deseable por cuestiones de rendimiento.
- **Restricción de los accesos no autorizados:** cada usuario debe tener permisos de acceso y autorización.
- **Cumplimiento de las restricciones de integridad:** el SGBD debe ofrecer recursos para definir y garantizar el cumplimiento de ellas.

Los sistemas gestores de bases de datos que cumplan estas características presentan varias ventajas, entre las que destacan:

- Facilidad de manejo de grandes volúmenes de información.
- Gran velocidad en poco tiempo.
- Independencia del tratamiento de información.
- Seguridad de la información (acceso a usuarios autorizados), protección de información, modificaciones, inclusiones, consultas.
- Integridad referencial al terminar los registros.

1.2.6.1 MySQL

MySQL es la base de datos Open Source más popular. Su continuo desarrollo y su creciente popularidad hacen de él un competidor cada vez más directo en la materia de las bases de datos. Un

ejemplo de ello es Oracle. MySQL es un sistema de administración de base de datos (Database Management System, DBMS) para bases de datos relacionales. El mismo tiene un sistema multihilo y multiusuario con más de seis millones de instalaciones.

MySQL fue escrito en C y C++. Se destaca por su gran adaptación a diferentes entornos de desarrollo, lo que permite su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java. No por gusto son conocidos PHP y MySQL como la pareja explosiva, además de su integración en diferentes sistemas operativos. También es muy destacable, la condición de Open Source, que hace que su utilización sea gratuita e incluso se pueda modificar con tal libertad que se pueda descargar su código fuente. Esto ha favorecido su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet.

Este gestor de base de datos brinda varias ventajas:

- Mayor rendimiento.
- Mejores utilidades de administración.
- Integración perfecta con PHP.
- Sin límites en los tamaños de los registros.
- Mejor control de acceso a usuarios.

1.2.7 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 (versión cifrada o autenticada). Un servidor Web básico cuenta con esquema de funcionamiento simple, basado en ejecutar infinitamente el siguiente bucle: [27]

1. Espera peticiones en el puerto TCP (Transmission Control Protocol) indicado (el estándar por defecto para el protocolo HTTP es el puerto 80).
2. Recibe una petición.
3. Busca el recurso.

4. Envía el recurso utilizando la misma conexión por la que recibió la petición.

5. Vuelve al segundo punto.

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

1.2.7.1 Servidor Web Apache

Apache es el servidor Web hecho por excelencia. Su coste gratuito, su robustez, su gran estabilidad y gran extensibilidad, lo han convertido en una herramienta potente y configurable. Lo que lo convierte en uno de los servidores más utilizados en la actualidad.

La licencia de este servidor es descendiente de las licencias BSD, no es GPL. Esta licencia permite hacer lo que desee el cliente con el código fuente, siempre que sea reconocido su trabajo.

Apache presenta algunas razones por la cual se ha hecho tan popular en el mundo. Entre ellas se pueden mencionar: [28]

- Corre en una gran cantidad de sistemas operativos, lo que lo hace prácticamente universal.
- Es una tecnología gratuita de código fuente abierta.
- Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades de este servidor.
- Trabaja con gran cantidad de Perl, PHP, y otros lenguajes de script. También trabaja con Java y con páginas jsp. Además de tener todo el soporte que se necesita para tener páginas dinámicas.
- Permite personalizar la respuesta ante los posibles errores que se pueden presentar en el servidor.
- Tiene una alta configurabilidad en la creación y gestión de logs. También permite la creación de ficheros de log a la medida del administrador. De este modo se puede tener un mayor control de lo que sucede en el servidor.

1.2.8 UML (Lenguaje Unificado de Modelado)

UML es un lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software. Es un sistema de notaciones, que incluso incluye el significado de estas, destinado a los sistemas de modelado que utilizan conceptos orientados a objetos. Este lenguaje nació en el año 1994 por iniciativa de Grady Booch y James Rumbaugh. Desde 1997 UML es un estándar, de acuerdo con la asociación para fijar los estándares de la industria, Object Management Group.

UML es un lenguaje para construir modelos. No guía al desarrollador en la forma de realizar el análisis y diseño orientados a objetos ni le indica cuál proceso de desarrollo adoptar. Está compuesto por un conjunto de símbolos y elementos gráficos que al unirse conforman diagramas. Los diagramas tienen como finalidad representar diversas perspectivas de un sistema, las cuales se conocen como modelo. [29]

1.2.9 RUP (Proceso Unificado de Desarrollo de Software)

Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un software. Sin embargo, un Proceso Unificado o RUP es más que un simple proceso; 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. Este proceso está basado en componentes; lo que quiere decir que el sistema o aplicación en construcción está formado por componentes interconectados a través de interfaces bien definidas.

RUP utiliza el UML (Lenguaje Unificado de Modelado) para representar y preparar todos los esquemas del sistema de software que se va a desarrollar, siendo parte clave de RUP. No obstante, sus verdaderos aspectos definitorios se pueden resumir en tres ideas claves: dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental. Resumiendo, un sistema de software se define para brindar servicios a sus usuarios. Entonces, para construirlo con el éxito requerido, se necesita conocer qué es lo que los futuros usuarios solicitan. Además de que es conveniente dividir el trabajo en pequeñas partes o miniproyectos, representando lo que se conoce como iteración. [30]

Características de RUP:

- **Dirigido por casos de uso:** la principal tarea de un sistema de software es brindar servicios a los usuarios. Los usuarios representan a alguien o algo que interactúa con la aplicación que se

desarrolla. Una interacción de este tipo es un caso de uso. RUP lo define como el conjunto de acciones que debe realizar el sistema para proporcionar un resultado importante a un usuario determinado. Representa a los requerimientos funcionales de la aplicación y sirve de guía para el proceso de desarrollo, ya sea el diseño, implementación y prueba.

- **Centrado en la arquitectura:** la arquitectura incluye los aspectos estáticos y dinámicos más significativos del sistema. Además de incluir las necesidades de los usuarios e inversores, también se ve influida por otros factores técnicos, como la plataforma en la que va a funcionar el software (sistema operativo, hardware, sistema de gestión de base de datos, protocolos con los que debe coexistir el sistema). Para obtener un producto con éxito, debe existir un equilibrio entre funcionalidad y características técnicas. La arquitectura representa la forma del sistema.
- **Iterativo e incremental:** Debido al alto grado de complejidad de los sistemas actuales, se hace factible dividir el trabajo en pequeñas partes o en miniproyectos. Cada uno es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo y los incrementos al crecimiento del producto. En cada iteración, los desarrolladores seleccionan un grupo de casos de uso y seguidamente siguen con todo el proceso de diseño, implementación y prueba.

1.2.10 Herramientas a Utilizar

Para la realización de cualquier desarrollo Web se necesitan un conjunto de herramientas. Estas van a ser utilizadas por los miembros de un equipo, tanto para la etapa de diseño como para la construcción de una aplicación.

1.2.10.1 Dreamweaver 8

Dreamweaver 8 es la opción profesional para la creación de sitios y aplicaciones web. Es un software fácil de usar que permite crear páginas web profesionales. Las funciones de edición visual del mismo permiten agregar rápidamente diseño y funcionalidad a las páginas, sin la necesidad de programar manualmente el código HTML. 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. Proporciona una combinación potente de herramientas visuales de disposición, características de desarrollo de aplicaciones y soporte para la edición de código.

1.2.10.2 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, integración y las etapas de los procesos. Sólo Studio 5 brinda todo lo que necesita para construir, probar y entregar aplicaciones PHP de alto rendimiento. El mismo se ha diseñado para una amplia gama de programadores y existen dos ediciones: Standard y Professional. Zend Studio TM 5 proporciona una facilidad de uso inigualable, escalabilidad, fiabilidad.

1.2.10.3 Notación JSON

JSON es el acrónimo del JavaScript Object Notation. Es un formato alternativo de envío y recepción de datos; es decir reemplaza a XML o al envío de texto plano. Este formato de datos es más liviano que XML.

La notación de objetos mediante JSON es una de las características principales de JavaScript y un mecanismo definido en los fundamentos básicos del lenguaje. JSON permite definir arrays y objetos de una manera concisa. Esto supone una gran ventaja respecto a la notación tradicional de los objetos y los arrays. Los arrays normales se definen de forma abreviada como una lista de valores separados por comas y encerrados entre corchetes, JSON permite definir los arrays asociativos de una forma mucho más breve.

1.2.10.4 YUI

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 para el desarrollo rápido y sencillo de aplicaciones web complejas. La librería completa está dividida en módulos y componentes relacionados con CSS, DOM, eventos, AJAX y otros. Entre las utilidades disponibles se encuentran calendarios, tooltips, cuadros que autocompletan texto, árboles jerárquicos, entre otros. Además de esas utilidades, la YUI incluye un completo módulo de menús que permite realizar decenas de tipos de menús diferentes: horizontales, verticales, desplegados, estáticos, menús contextuales, menús de aplicación, menús realizados con XHTML o con JavaScript.

Entre sus características se encuentran: [31]

- Capacidad de generar código ordenado para el programador.
- Posibilidad de reutilizar controladores.
- Manejo de AJAX.
- Personalización de los componentes.
- Soporte por parte de Yahoo Developers.

Entre sus componentes/controles se encuentran:

- DataTable: Para mostrar datos de forma tabular.
- TreeView: Una especie de menú desplegable.
- Calendar: Control para seleccionar la fecha.
- Slider: Para seleccionar un valor numérico.

La versión 2.5.1 se trata de un parche de liberación general con 67 cuestiones que se abordan en toda la biblioteca. YUI (2.5.1) mejora el apoyo a los desarrolladores de ejecutar YUI en medio ambiente y se ocupa de un importante problema de seguridad con el JSON en FireFox. La versión mejora las **Tablas de Datos** que siguen en un rollo de actualizaciones, incluidos 2.5.1 con una importante mejora del rendimiento al añadir, actualizar, borrar filas dinámicamente. La versión ha añadido un **Configurador YUI** a las páginas web, proporcionando un panel de estilo de interfaz de selección. Al seleccionar los componentes que desea utilizar, el Configurador YUI le incluye archivos configurables para su aplicación.

1.2.10.5 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. Sus funcionalidades son las siguientes:

- Capacidad fuerte en el diseño lógico.
- Sincronización bidireccional de los diseños lógico y físico.
- Construcción automática de bases de datos.

- Reingeniería inversa de bases de datos.
- Documentación basada en HTML.
- Un repositorio para el modelado.

1.2.10.6 Rational Rose

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación de UML 1.1. La misma propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema. Utiliza una vista estática y otra dinámica de los modelos del sistema: uno lógico y otro físico. Permite crear y refinar estas vistas, lo que establece un modelo completo que representa el dominio del problema y el sistema de software.

Como características principales del mismo se pueden destacar:

- Admite notaciones como: UML, COM, OMT y Booch.
- Realiza chequeo semántico de los modelos.
- Desarrollo multiusuario.
- Ingeniería inversa.
- Generación de documentación.
- Presenta un lenguaje de script para poder ampliar su funcionalidad.

1.2.10.7 EMS SQL Manager 2005

Es una excelente aplicación para administrar y desarrollar bases de datos de SQL Server. Es una herramienta que sirve para autocompletar código, crear Triggers, realizar procedimientos almacenados, vistas, consultas, y otros. Su interfaz es sumamente atractiva e incluye un modo guiado de trabajo.

En este capítulo, se han analizado conceptos y definiciones que posibilitan comprender el proceso de desarrollo del Componente Registro de Estudiantes. Se realizó un estudio de sistemas anteriores, para darle solución a la situación problemática. También se realizó un estudio y análisis de tecnologías, patrones y arquitectura, lenguajes y herramientas que se tendrán en cuenta para la creación del sistema antes mencionado. Se define por el Área Temática SAS (Sistemas de Apoyo a la Salud) trabajar con tecnología SO Linux, Servidor Web Apache, Gestor de Base de Datos MySQL y como lenguaje de programación PHP. Se utiliza además el motor de plantillas Smarty y Ajax como Metodología. Además se definió trabajar con una arquitectura en tres capas implementando el patrón Modelo Vista Controlador.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En este capítulo se describe la propuesta de solución para ser utilizada en las secretarías docentes del Ministerio de Salud Pública (MINSAP). Debido a que existe una incipiente estructura de procesos de negocio, se determina desarrollar un Modelo de Dominio, para poder comprender mejor el contexto en el cual se enmarca el sistema; lo que describe las relaciones entre sus definiciones. Se muestra además los requerimientos funcionales y no funcionales, donde los primeros formarán parte de los Casos de Uso del Sistema.

2.1 Conceptos asociados al dominio del problema

Aquí se representan algunos conceptos que se identifican en el Modelo de Dominio para un mejor entendimiento del sistema.

Secretaria(o) Docente: Es el responsable de ingresar nuevos estudiantes al sistema. Sólo tiene acceso a la información de su centro. Gestiona las brigadas docentes que se crean en el centro de estudios.

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

Servicio: Operación proporcionada por un componente determinado.

2.2 Objeto de estudio

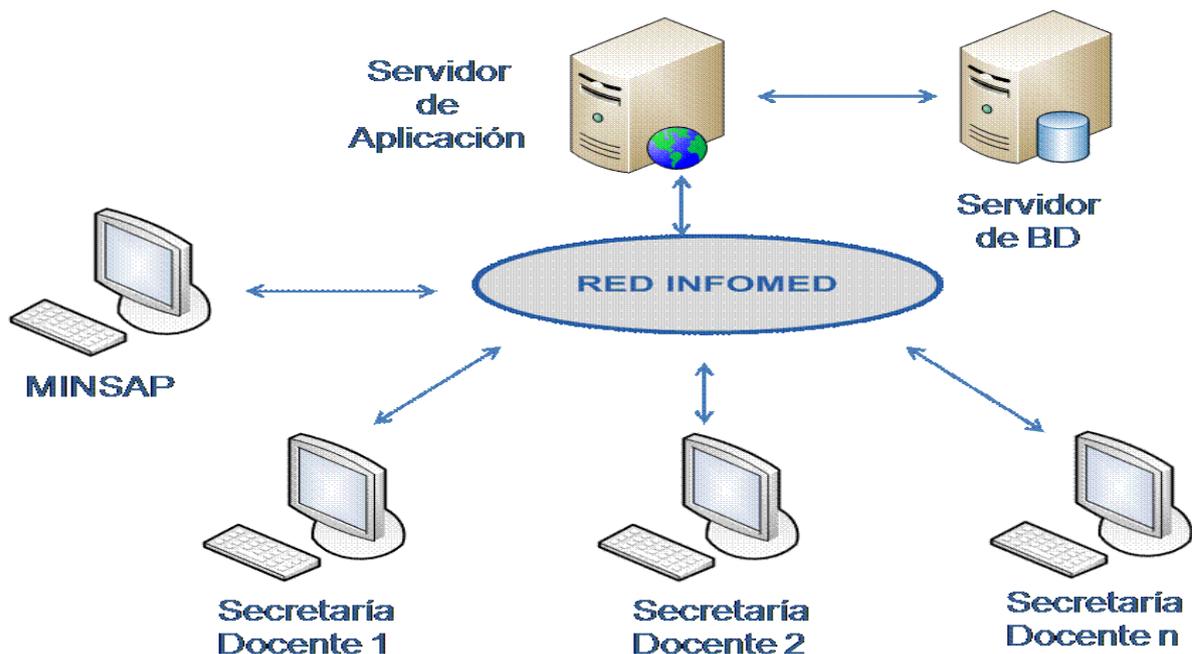
2.2.1 Situación problemática

Actualmente en Cuba la gestión de información en la formación de los recursos humanos del Ministerio de Salud, y específicamente de sus instituciones clínico docentes se hace manualmente o por sistemas no profesionales de tecnología obsoleta, con bajas prestaciones, que fueron desarrolladas sin un continuo mejoramiento. Esto conlleva a que el nivel de errores y demora en la entrega de los datos para la toma de decisiones a todas las instancias de dirección sea considerable. Este proceso se desarrolla en las secretarías docentes de las instituciones educativas de esta esfera. En ellas se llevan a cabo los diferentes procesos en la formación curricular de un estudiante; la clasificación de toda la información que contiene el trabajo de secretaría por perfiles; matrícula y ratificación de la

misma; la confección de los registros y control de las brigadas de los estudiantes por año, y toda actividad de forma general, lo cual se incluye dentro de las tareas de gestión de registro.

2.3 Propuesta del sistema

El sistema que se propone en el trabajo 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 administración que brindará flexibilidad ante los aspectos relacionados con la configuración del área de trabajo. Este dará la posibilidad de configurar las planillas de matrícula a nivel nacional, que serán destinadas luego a cada unidad de salud de acuerdo a sus necesidades. Para ello se implementará un Componente de Nomencladores que brindará servicios que serán utilizados para el Componente Registro de Estudiantes que se desarrollará durante el transcurso de este Trabajo de Diploma. A continuación se muestra un esquema sobre cómo funcionará la aplicación.



2.4 Modelo de Dominio

Un Modelo de Dominio tiene como propósito mostrar cómo se capturan los tipos más importantes de objetos en el contexto del sistema. En él se representan las actividades o eventos que surgen en el entorno en que trabaja el sistema.

2.4.1 Diagrama del Modelo de Dominio

En un Modelo de Dominio se capturan los tipos más importantes de objetos en el contexto del sistema. Aquí se representan las actividades o eventos que surgen en el entorno en que se trabaja. Se muestran las clases del dominio así como las relaciones entre ellas, para un mejor entendimiento de cómo trabaja el sistema.

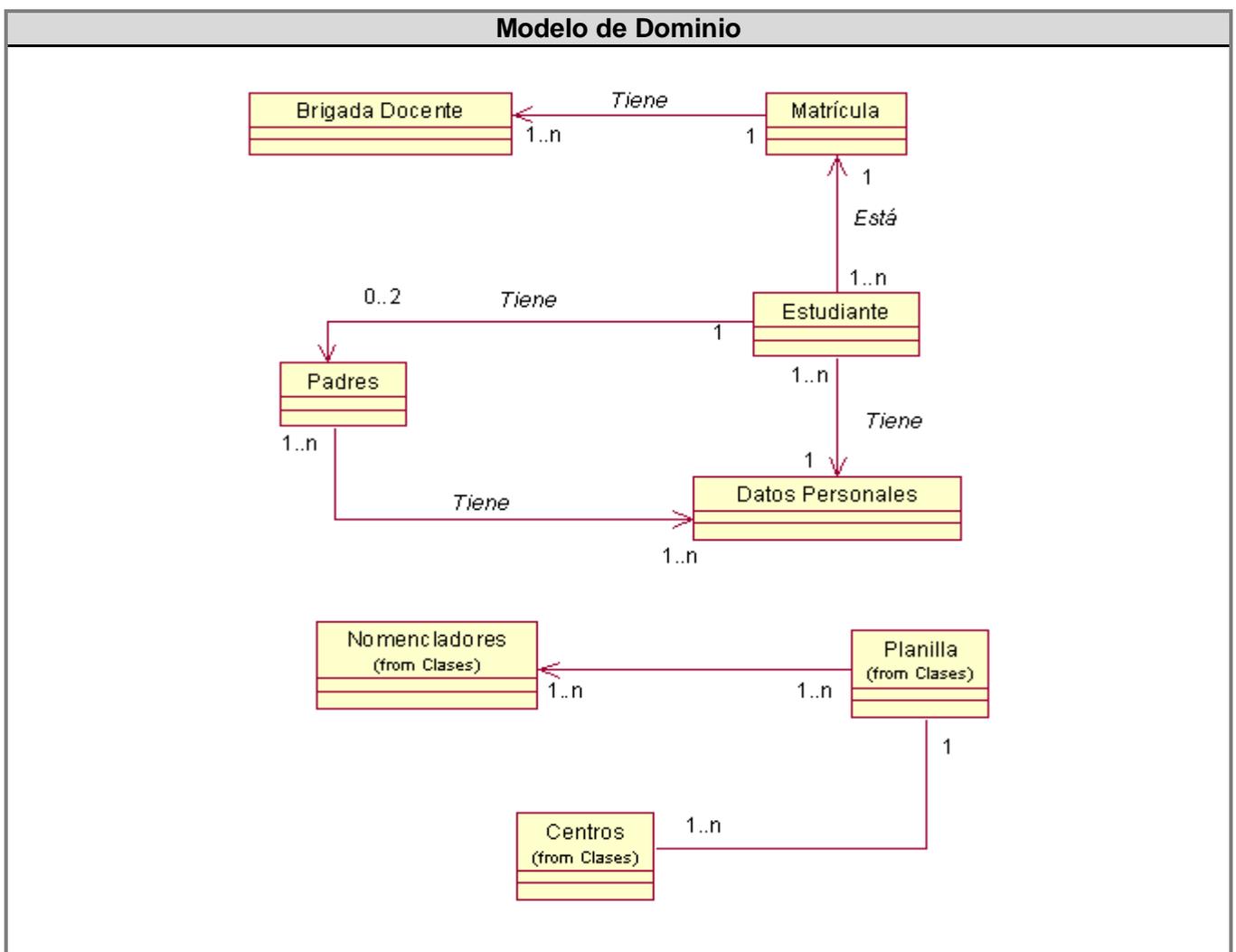


Fig. 2.1 Diagrama del modelo de dominio

Se puede observar además la relación existente entre los trabajadores y las entidades que responden a las actividades fundamentales del sistema. Ver [Anexo1](#).

2.5 Especificación de los requerimientos de software

Este proceso es fundamental, pues a partir de él se mantiene una línea de trabajo organizado con metas a desarrollar. 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 condiciones o capacidades que debe tener un sistema para satisfacer un contrato entre los interesados, con un final denominado producto.

2.5.1 Requerimientos Funcionales

Los requerimientos funcionales son condiciones o capacidades que el sistema debe cumplir. Son las acciones que el sistema debe permitir realizar para cumplir con las metas trazadas inicialmente y de esta forma dar solución a los problemas anteriormente identificados. Las acciones son llevadas a cabo por los usuarios autorizados. En este caso el secretario docente o el sistema que se realiza internamente. Para este sistema se definen:

LISTADO DE REQUERIMIENTOS FUNCIONALES	
<p>R1. Gestionar Datos de Matrícula de un estudiante (Planilla de matrícula: datos generales, personales, familiares, de la carrera).</p> <p>1.1 Insertar datos del estudiante.</p> <p>1.2 Actualizar datos del estudiante.</p> <p>1.3 Eliminar a un estudiante.</p> <p>R2. Buscar un estudiante.</p> <p>R3. Gestionar Brigadas Docentes.</p> <p>3.1 Insertar una brigada docente.</p> <p>3.2 Actualizar una brigada docente.</p> <p>3.3 Eliminar una brigada docente.</p> <p>R4. Conformar brigadas.</p> <p>4.1 Conformar brigada.</p> <p>4.2 Mover estudiante de brigada.</p> <p>4.3 Eliminar estudiante de la brigada.</p>	<p>R5. Configurar Planilla Matrícula.</p> <p>5.1 Vista previa de la planilla.</p> <p>R6. Gestionar Planilla.</p> <p>6.1 Editar planilla.</p> <p>6.2 Eliminar planilla.</p> <p>R7. Designar Centro Matrícula.</p> <p>7.1 Editar designación de centro.</p> <p>7.2 Eliminar designación de centro.</p> <p>R8. Asignar Planilla Matrícula a los Centros.</p> <p>8.1 Editar asignación de planilla.</p> <p>8.2 Eliminar asignación de planilla.</p> <p>R9. Gestionar Nomencladores.</p> <p>9.1 Insertar datos de nomencladores.</p> <p>9.2 Editar datos de nomencladores.</p> <p>9.3 Eliminar datos de nomencladores.</p>

Tabla 2.1 Listado de requerimientos funcionales

Los requerimientos funcionales son el aporte esencial de los Casos de Uso pues estos son las funciones que realizan los actores en el sistema:

DEFINICIÓN DE CASOS DE USO	
<p>CUS-1 Gestionar Datos de Matrícula de un estudiante.</p> <p>R1. Gestionar Datos de Matrícula de un estudiante (Planilla de matrícula: datos generales, personales, familiares, de la carrera).</p> <p>1.1 Insertar datos del estudiante.</p> <p>1.2 Actualizar datos del estudiante.</p>	<p>CUS-5 Configurar Planilla Matrícula.</p> <p>R5. Configurar Planilla Matrícula.</p> <p>5.1 Vista previa de la planilla de matrícula.</p> <p>CUS-6 Gestionar Planilla.</p> <p>R6. Gestionar Planilla.</p> <p>6.1 Editar Planilla.</p> <p>6.2 Eliminar Planilla.</p>
<p>CUS-2 Buscar un estudiante.</p> <p>R2. Buscar un estudiante.</p>	<p>CUS-7 Designar Centro Matrícula.</p> <p>R7. Designar Centro Matrícula.</p> <p>7.1 Editar designación de centro.</p> <p>7.2 Eliminar designación de centro.</p>
<p>CUS-3 Gestionar Brigadas Docentes.</p> <p>R3. Gestionar Brigadas Docentes.</p> <p>3.1 Insertar una brigada docente.</p> <p>3.2 Actualizar una brigada docente.</p> <p>3.3 Eliminar una brigada docente.</p>	<p>CUS-8 Asignar Planilla Matrícula a los Centros.</p> <p>R8. Asignar Planilla Matrícula a los Centros.</p> <p>8.1 Editar asignación de planilla.</p> <p>8.2 Eliminar asignación de planilla.</p>
<p>CUS-4 Conformar brigadas.</p> <p>R4. Conformar brigadas.</p> <p>4.1 Conformar brigada.</p> <p>4.2 Mover estudiante de brigada.</p> <p>4.3 Eliminar estudiante de la brigada.</p>	<p>CUS-9 Gestionar Nomencladores.</p> <p>R9. Gestionar Nomencladores.</p> <p>9.1 Insertar datos de nomencladores.</p> <p>9.2 Editar datos de nomencladores.</p> <p>9.3 Eliminar datos de nomencladores.</p>

Tabla 2.2 Definición de casos de uso del sistema

2.5.2 Requerimientos No Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Estas cualidades son las que hacen de un producto un atractivo para ser utilizado por su calidad, usabilidad, seguridad, y facilidad de entendimiento.

2.5.2.1 Usabilidad

El sistema podrá ser utilizado por cualquier usuario, incluso con mínimos conocimientos en el manejo de la computadora y el ambiente Web en sentido general, debido a que contará con una ayuda a fin de documentar al usuario en su utilización.

2.5.2.2 Rendimiento

El sistema deberá ser rápido antes las posibilidades de los usuarios y en el procesamiento de la información. La eficiencia de la aplicación estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo Cliente/Servidor, y la velocidad de las consultas a la base de datos. Se realizará la validación de los datos en el cliente y en el servidor de aquellas que por cuestiones de seguridad o de acceso a los datos lo requieran. Se logra así un tiempo de respuesta más rápido, una mayor velocidad de procesamiento, y un mayor aprovechamiento de los recursos.

2.5.2.3 Requerimiento de ayuda y documentación

Se documentará la aplicación con una ayuda dirigida a los diferentes tipos de usuarios para garantizar el soporte del sistema. Se realizará mantenimiento al sistema a fin de aumentar las funcionalidades a través de versiones posteriores.

El sistema debe tener una interfaz sencilla, agradable, legible y de fácil uso para el usuario. El contenido será mostrado de manera comprensible y fácil de leer. Deben evitarse imágenes almacenadas en archivos de gran tamaño.

2.5.2.4 Soporte

Se requiere un servidor de bases de datos con las siguientes características:

- Soporte para grandes volúmenes de datos y velocidad de procesamiento.
- Tiempo de respuesta rápido

Por parte del cliente se requiere un navegador capaz de interpretar JavaScript.

2.5.2.5 Seguridad

Teniendo en cuenta que la aplicación almacenará información referente al proceso de matrícula de los estudiantes de los centros docentes del MINSAP, estará protegida contra accesos no autorizados utilizando el mecanismo de autorización y auditoría que brinda el SAAA (Sistema de Autorización Autenticación y Auditoría) para garantizar el nivel de acceso dado por la asignación de roles, de manera que cada uno tenga datos de acceso propios.

Se crearán usuarios con diferentes niveles de acceso a la aplicación, tanto para la matrícula de estudiantes como para la parte que contiene la configuración. No obstante los usuarios accederán de forma rápida y operativa al sistema sin que los requerimientos de seguridad sean un retardo para ellos.

2.5.2.6 Requerimientos de Software

MySQL versión 5.0.18 o cualquier otra que soporte trabajo con procedimientos almacenados, vistas, y otros. Servidor Apache versión 2.0.55 o compatible.

PHP versión 5 que soporte trabajo orientado a objetos.

2.5.2.7 Requerimientos de Hardware

Server de rack de 19 'con:

- NIC de 1 Gbit Ethernet.
- Array SCSI de 3 Hard Disk > 70 Gb.
- 1 Gbyte RAM.
- Dual processor.
- Fuente redundante.
- Certificación del fabricante para correr Linux (Debian sarge).

2.5.3 Modelos de Casos de Uso

El Modelo de Casos de Uso del Sistema es un artefacto de Ingeniería de Software que describe, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Esto permite que el establecimiento de un acuerdo entre clientes y desarrolladores sobre las

condiciones y requerimientos que debe cumplir el sistema. Este modelo está formado por actores, casos de usos y las relaciones que se establecen entre estos. Es decir, representa gráficamente a los procesos y su interacción con los actores y constituye una entrada de gran valor para las siguientes fases de construcción de un software.

2.5.3.1 Definición de los actores

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.
Editor Nacional.	Es el encargado de la configuración de la planilla de matrícula a nivel nacional. Designa los centros que realizarán matrícula y además asigna las planillas a los distintos centros.

Tabla 2.3 Justificación de actores del sistema

2.6. Diagrama de Casos de Uso del sistema

Un diagrama de casos de uso (Use Case Diagram) es una representación gráfica de parte o el total de los actores y casos de uso del sistema, incluyendo sus interacciones. Se denota una representación gráfica del entorno del sistema (actores) y su funcionalidad principal (casos de uso). Se muestran, por tanto, los distintos requerimientos funcionales que se esperan del sistema y cómo se relaciona con su entorno (usuarios).

En este diagrama se representan las relaciones entre actores y casos de uso del sistema. A continuación se muestra el Diagrama de Casos de Uso por paquetes del Componente Registro de Estudiantes:

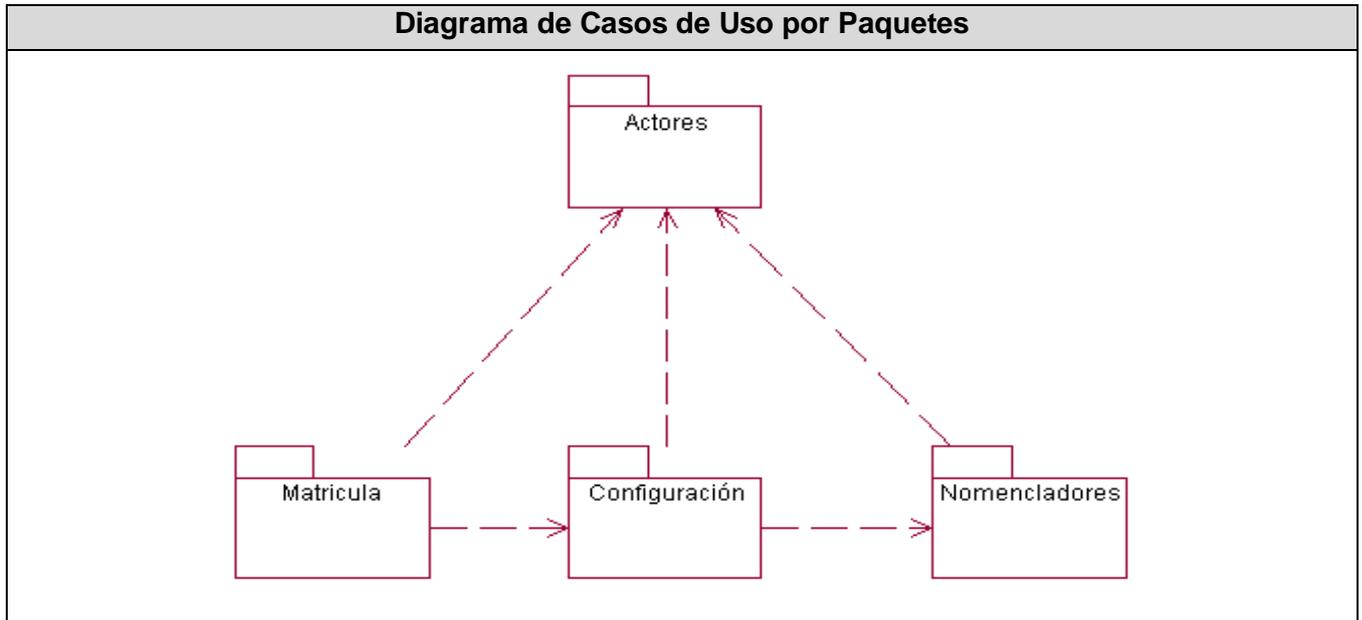


Fig. 2.2 Diagrama de casos de uso del sistema por paquetes

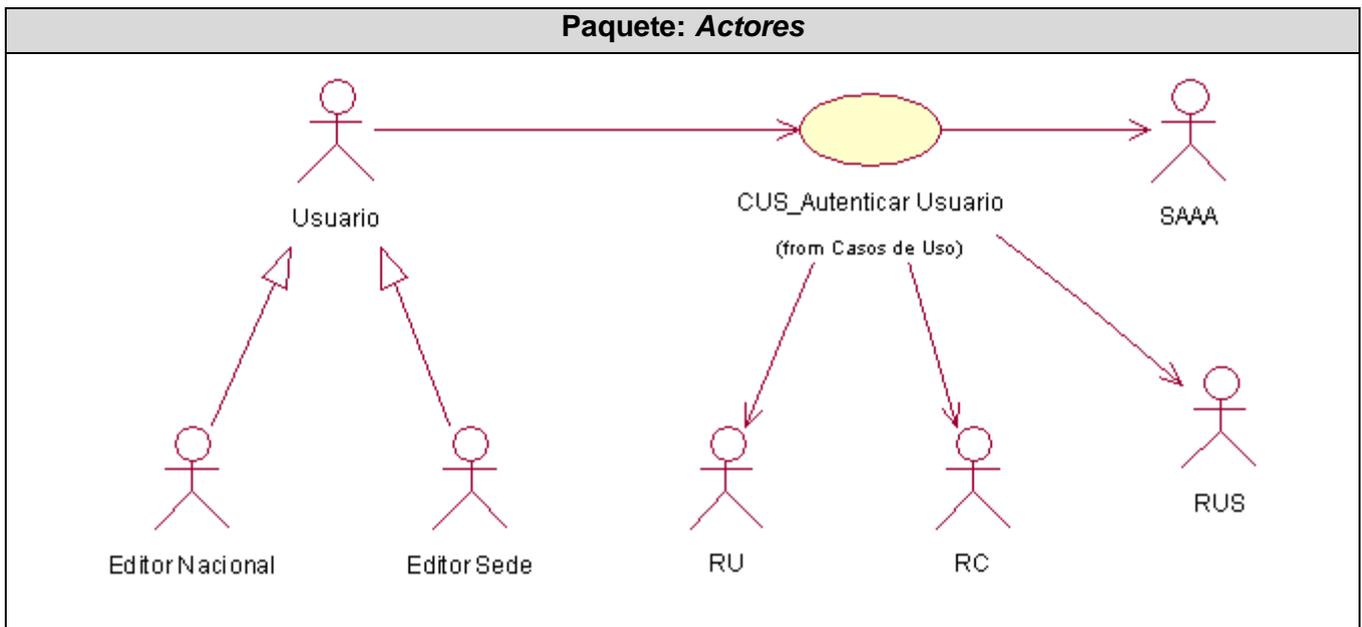


Fig. 2.3 Diagrama de casos de uso del paquete "Actores"

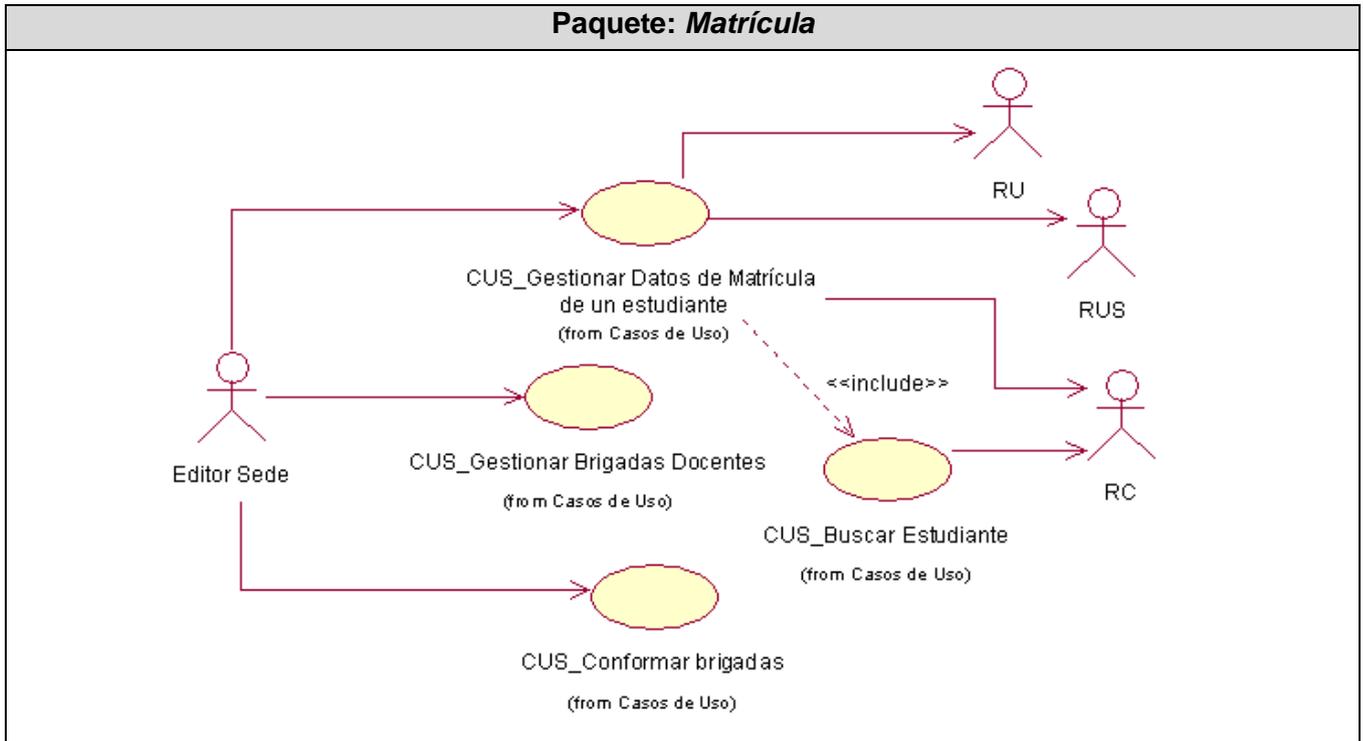


Fig. 2.4 Diagrama de casos de uso del paquete “Matrícula”

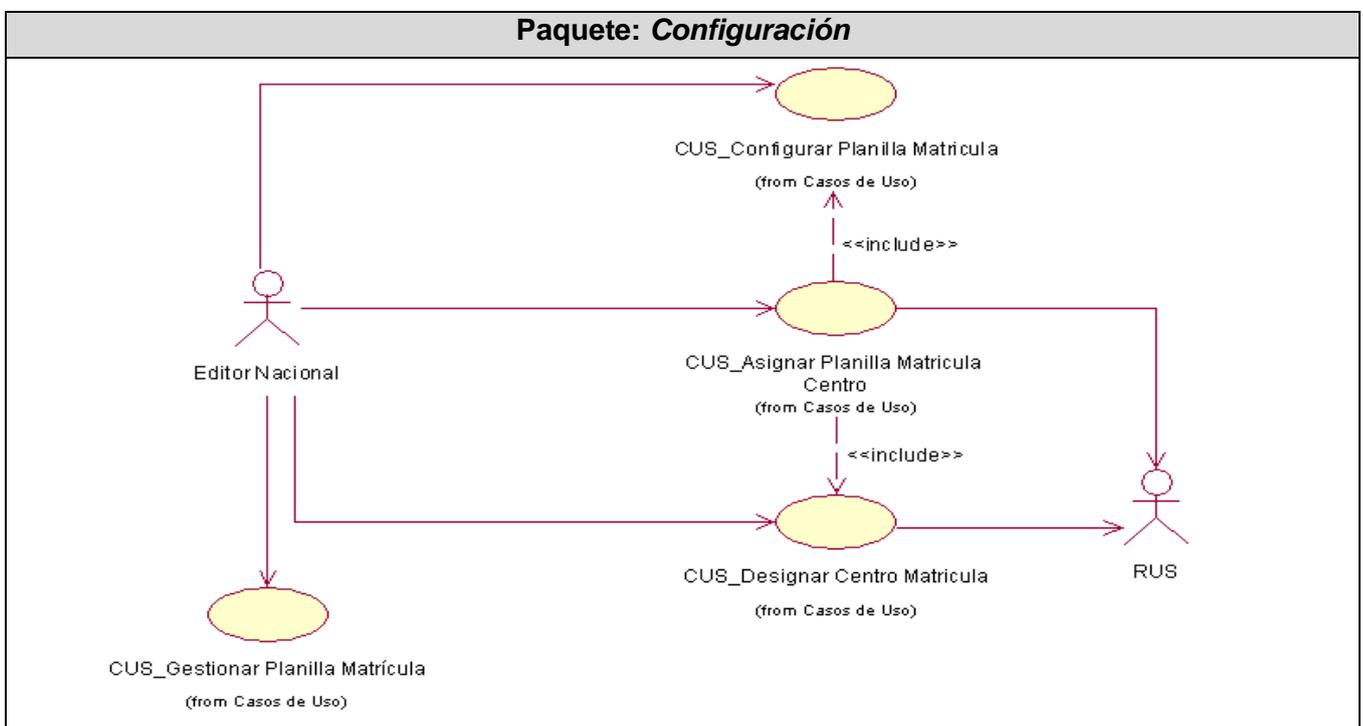


Fig. 2.5 Diagrama de casos de uso del paquete “Configuración”

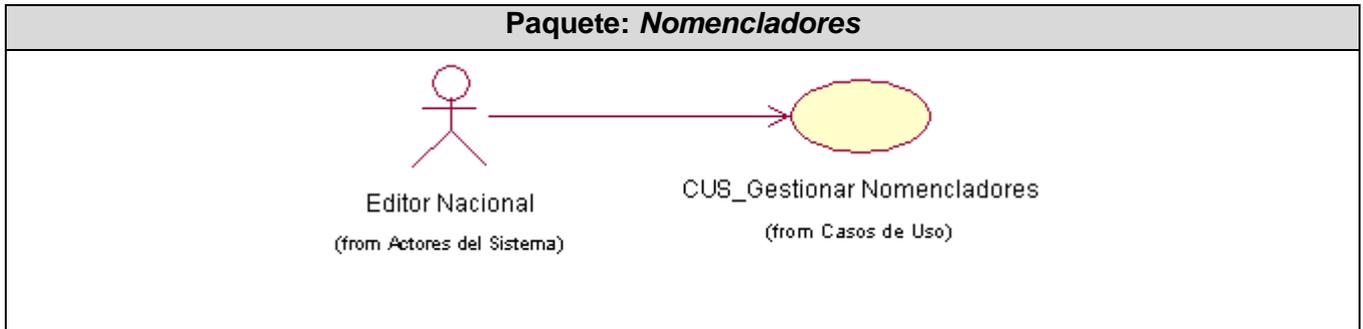


Fig. 2.6 Diagrama de casos de uso del paquete “Nomencladores”

2.6.1 Descripción textual de los Casos de Uso (CU)

A continuación se describen los casos de uso identificados, mediante los cuales se satisfacen los requerimientos funcionales del sistema en desarrollo:

Caso de Uso.	Gestionar Datos de Matrícula de un Estudiante.	
Actores.	Editor Sede (inicia).	
Propósito.	Registrar o actualizar los datos de matrícula del estudiante en el sistema.	
Resumen.	El caso de uso se inicia cuando el editor sede necesita insertar los datos de un nuevo estudiante al sistema, o actualizar los datos de uno matriculado. Finaliza una vez registrados o actualizados estos datos.	
Pre-condiciones.	Editor Sede autenticado satisfactoriamente por el sistema.	
Referencias.	R1.	
Prioridad.	Crítico.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Editor Sede, si desea matricular a un nuevo estudiante o actualizar los datos de matrícula de uno matriculado, selecciona la opción “Nuevo Estudiante” o “Actualizar matrícula”, respectivamente.	1.1 Se realiza el caso de uso Buscar Estudiante. 1.2 Dependiendo de la operación a realizar: a) Si la opción seleccionada es “Nuevo estudiante”, ir a la sección “Matricular estudiante”. b) Si la opción seleccionada es “Actualizar matrícula” ir a la sección “Actualizar datos de un estudiante”.	
Sección “Matricular estudiante”		

Acción del Actor	Respuesta del Sistema
<p>2. El Editor Sede llena los campos de datos de la matrícula mostrados en los <i>tabviews</i>, y en el último pulsa el botón “Guardar”.</p> <p>3. El Editor Sede puede matricular otro estudiante pulsando el botón “Nuevo”.</p>	<p>1.3 El sistema muestra los campos de datos de la matrícula de un estudiante en diferentes <i>tabviews</i> “Datos de la Carrera Pregrado”, “Datos Generales”, “Datos de la Dirección”, “Datos Personales” (divididos en dos <i>tabview</i>), “Datos Familiares”, y “Datos Referentes a la Carrera”.</p> <p>2.1 El sistema registra los datos del estudiante y muestra un mensaje de confirmación.</p>
Sección “Actualizar datos de estudiante”	
Acción del Actor	Respuesta del Sistema
<p>2. El Editor Sede actualiza los campos de datos que desee y pulsa el botón “Actualizar”.</p>	<p>1.3 El sistema muestra todos los datos de la matrícula de un estudiante en diferentes <i>tabviews</i> “Datos de la Carrera Pregrado”, “Datos Generales”, “Datos de la Dirección”, “Datos Personales” (divididos en dos <i>tabview</i>), “Datos Familiares”, y “Datos Referentes a la Carrera”.</p> <p>2.1 El sistema modifica los datos del estudiante y muestra un mensaje de confirmación.</p>
Flujos Alternos	
<p>2.1 Si cometió algún error al llenar los campos de datos, el sistema muestra un mensaje indicando el error cometido.</p>	
Pos-condiciones.	Se matricula un estudiante en el sistema, o se actualizan los datos de un estudiante matriculado.

Tabla 2.4 Descripción del CU “Gestionar Datos de Matrícula de un Estudiante”

Caso de Uso.	Buscar Estudiante.
Actores.	Editor Sede (inicia).
Propósito.	Buscar por diferentes criterios de búsqueda a un estudiante determinado.
Resumen.	El Caso de Uso se inicia cuando se desea insertar, actualizar o eliminar los

	datos de un estudiante. Para ello primeramente se debe buscar ese estudiante. Finaliza el caso de uso una vez que se encuentra el estudiante.	
Pre-condiciones.	Editor Sede autenticado satisfactoriamente por el sistema. El estudiante debe estar registrado en el sistema.	
Referencias.	R2.	
Prioridad.	Crítico.	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	<p>2. El Editor Sede debe llenar alguno de los campos para buscar al estudiante: nombre, apellidos, carné de identidad, o sexo del mismo, y pulsa el botón “Buscar”.</p> <p>3. El Editor Sede selecciona el estudiante que desee haciendo clic en la fila del estudiante correspondiente.</p>	<p>1.1 El sistema muestra la página de búsqueda.</p> <p>2.1 El sistema muestra un listado de los estudiantes que cumplan con los criterios de búsqueda, se muestra el código del estudiante y su nombre y apellidos.</p> <p>3.1 El sistema muestra la página correspondiente que invocó la búsqueda.</p>
Flujos Alternos		
	2.1 Si no se encuentran estudiantes con los criterios de búsqueda deseados, el sistema muestra un mensaje indicando que no existe ningún estudiante con estos datos.	
Pos-condiciones.	Se muestra el estudiante buscado.	

Tabla 2.5 Descripción del CU “Buscar Estudiante”

Caso de Uso.	Gestionar Brigada Docente.
Actores.	Editor Sede (inicia).
Propósito.	Crear una brigada docente, modificar sus datos, o eliminarla.
Resumen.	El Caso de Uso se inicia cuando el Editor Sede necesita crear una brigada docente en el sistema, modificar los datos de una brigada o eliminarla. Finaliza una vez creada la brigada docente, actualizada o eliminada.
Pre-condiciones.	Editor Sede autenticado satisfactoriamente por el sistema.

Referencias.	R3.
Prioridad.	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1. El Editor Sede selecciona la opción “Gestionar brigadas” del menú.</p> <p>2. El Editor Sede si desea crear una brigada debe seleccionar el tabview “Adicionar Brigadas”, si desea actualizar o eliminar alguna brigada, entonces debe seleccionar el tabview “Listar Brigadas”.</p>	<p>1.1 El sistema muestra una página con los <i>tabviews</i> “Listar Brigadas” y “Adicionar Brigadas”.</p> <p>2.1 Dependiendo de la operación a realizar:</p> <p>a) Si desea crear una brigada ir a la sección “Crear brigada docente”.</p> <p>b) Si desea actualizar datos de una brigada ir a la sección “Actualizar brigada docente”.</p> <p>c) Si desea eliminar una brigada ir a la sección “Eliminar brigada docente”.</p>
Sección “Crear brigada docente”	
Acción del Actor	Respuesta del Sistema
<p>3. El Editor Sede debe llenar los datos de la brigada: Brigada docente, Matrícula total, Año académico, Idioma, Estudio y Modelo de formación; y pulsar el botón “Adicionar”.</p>	<p>2.2 El sistema muestra la interfaz de “Adicionar Brigadas”.</p> <p>3.1 El sistema crea la brigada docente, registrando los datos de la misma, y muestra un mensaje confirmando la operación realizada.</p>
Sección “Actualizar brigada docente”	
Acción del Actor	Respuesta del Sistema
<p>3. El Editor Sede debe seleccionar del listado la brigada a actualizar.</p> <p>4 El Editor Sede debe modificar los datos de la brigada y pulsar el botón “Editar”.</p>	<p>2.2 El sistema muestra la interfaz de “Listar Brigadas”, que muestra el listado de brigadas del centro y el formulario para modificar los datos de las mismas.</p> <p>3.1 El sistema carga los datos de la brigada en el formulario.</p> <p>4.1 El sistema actualiza los datos de la brigada y muestra un mensaje de confirmación.</p>
Sección “Eliminar brigada docente”	
Acción del Actor	Respuesta del Sistema

<p>3. El Editor Sede debe seleccionar del listado la brigada a eliminar y pulsar el botón “Eliminar”.</p>	<p>2.2 El sistema muestra la interfaz de “Listar Brigadas”, que muestra el listado de brigadas del centro.</p> <p>3.1 El sistema elimina la brigada docente, e inmediatamente esta desaparece del listado de brigadas. Muestra un mensaje de confirmación.</p>
Flujos Alternos	
<p>3. Si desea cancelar la operación pulsa el botón “Cancelar” y el sistema saldrá de la opción.</p> <p>3.1 Si cometió algún error al llenar los datos de la brigada, el sistema mostrará un mensaje indicando el error cometido.</p>	
Pos-condiciones.	Se crea una nueva brigada docente, se actualiza o elimina una brigada existente.

Tabla 2.6 Descripción del CU “Gestionar Brigada Docente”

Caso de Uso.	Configurar Planilla Matrícula.
Actores.	Editor Nacional (inicia).
Propósito.	Dejar configurada la Planilla Matrícula.
Resumen.	El Caso de Uso se inicia cuando el Editor Nacional necesita configurar una o varias Planillas que serán utilizadas en algún centro de salud. Finaliza el caso de uso una vez configurada la Planilla.
Pre-condiciones.	Editor Nacional autenticado satisfactoriamente por el sistema. Los atributos necesarios para configurar la planilla deben estar creados (Nomencladores).
Referencias.	R5.
Prioridad.	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1. El Editor Nacional selecciona la opción “Configurar Planilla Matrícula” del menú.</p>	<p>1.1 El sistema muestra en diferentes <i>tabviews</i>: “Datos de la Carrera Pregrado”, “Datos Generales”, “Datos de la Dirección”, “Datos Personales” (divididos en dos <i>tabview</i>), “Datos Familiares”, y “Datos Referentes a la Carrera”, los</p>

<p>2 El Editor Nacional debe seleccionar los atributos necesarios para la matrícula en cada uno de los <i>tabviews</i> y en el último (“Datos Referentes a la Carrera”) debe escribir el nombre de la planilla y pulsar el botón “Salvar”.</p>	<p>listados de atributos con que se puede conformar la planilla de matrícula.</p> <p>2.1 El sistema almacena la planilla de matrícula y muestra un mensaje de confirmación.</p>
Flujos Alternos	
<p>2. El Editor Nacional puede ver cómo quedará la configuración de la planilla antes de salvarla, para ello debe pulsar el botón “Vista previa”.</p> <p>El Editor Nacional puede configurar una nueva planilla, para ello debe pulsar el botón “Nueva”.</p>	
Pos-condiciones	Se configuró una Planilla Matrícula.

Tabla 2.7 Descripción del CU “Configurar Planilla Matrícula”

Caso de Uso.	Designar Centro Matrícula.
Actores.	Editor Nacional (inicia).
Propósito.	Designar los centros que realizarán matrícula.
Resumen.	El Caso de Uso se inicia cuando el Editor Nacional necesita designar los centros que realizarán matrícula en el curso. Finaliza el caso de uso una vez designados los centros.
Pre-condiciones.	Editor Nacional autenticado satisfactoriamente en el sistema.
Referencias.	R 7.
Prioridad.	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

<p>1. El Editor Nacional selecciona la opción “Designar centros con matrícula” del menú.</p> <p>2. El Editor Nacional si desea cambiar o eliminar los centros que deben realizar matrícula por curso, debe seleccionar el tabview “Listar Centros”. Si desea designar los centros a realizar matrícula en un curso, entonces debe seleccionar el tabview “Designar Centros”</p>	<p>1.1 El sistema muestra una página con los <i>tabviews</i> “Listar Centros” y “Designar Centros”.</p> <p>2.1 Dependiendo de la operación a realizar:</p> <p>a) Si desea designar los centros a realizar matrícula en un curso ir a la sección “Designar Centros”.</p> <p>b) Si desea cambiar el curso en que se va a realizar matrícula, ir a la sección “Actualizar centros por curso”.</p> <p>c) Si desea eliminar un centro que no va a realizar matrícula, ir a la sección “Eliminar centro”.</p>
Sección “Designar Centros”	
Acción del Actor	Respuesta del Sistema
<p>3. El Editor Nacional debe seleccionar la provincia, municipio, y de estos, los centros que van a realizar matrícula, el curso, y pulsar el botón “Adicionar”.</p>	<p>2.2 El sistema muestra la interfaz de “Designar Centros”.</p> <p>3.1 El sistema designa los centros que deben realizar matrícula en un determinado curso. Muestra un mensaje confirmando la operación realizada.</p>
Sección “Actualizar centros por curso”	
Acción del Actor	Respuesta del Sistema
<p>3. El Editor Nacional debe seleccionar del listado, el centro a actualizarle el curso.</p> <p>4 El Editor Nacional debe modificar el curso al centro y pulsar el botón “Cambiar”.</p>	<p>2.2 El sistema muestra la interfaz de “Listar Centros”, que muestra el listado de centros y el formulario para actualizar los centros por curso.</p> <p>3.1 El sistema carga los datos del centro en el formulario.</p> <p>4.1 El sistema designa el curso al centro y muestra un mensaje de confirmación.</p>
Sección “Eliminar centro”	
Acción del Actor	Respuesta del Sistema

3. El Editor Nacional debe seleccionar del listado el centro a eliminar y pulsar el botón “Eliminar”.	2.2 El sistema muestra la interfaz de “Listar Centros”, que muestra el listado de centros. 3.1 El sistema elimina el centro, e inmediatamente este desaparece del listado de centros a realizar matrícula; muestra un mensaje de confirmación.
Flujos Alternos	
1.1 Si desea retirar las opciones de datos que ha seleccionado debe pulsar los botones que son de categoría atrás. 3, 4 Si desea cancelar la operación pulsa el botón “Cancelar” y el sistema saldrá de la opción.	
Pos-condiciones	Se designa el (los) centros que realizarán matrícula.

Tabla 2.8 Descripción del CU “Designar Centro Matrícula”

Caso de Uso.	Asignar Planilla Matrícula a Centro.
Actores.	Editor Nacional (inicia).
Propósito.	Asignarle una planilla de matrícula a un centro, o cambiarle o eliminarle las que tiene asignadas.
Resumen.	El Caso de Uso se inicia cuando el Editor Nacional necesita asignarle una planilla de matrícula a un centro o cambiar o eliminar las planillas asignadas. Finaliza el caso de uso una vez asignada la planilla, o cambiada o eliminada las asignadas.
Pre-condiciones.	Editor Nacional autenticado satisfactoriamente por el sistema. Configuradas las planillas de matrícula. Designado algún centro para realizar matrícula.
Referencias.	R8.
Prioridad.	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Editor Nacional selecciona la opción “asignar planilla a los centros” del menú.	1.1 El sistema muestra una página con los <i>tabviews</i> “Centros con Planillas Asignadas” y “Asignar Planillas”.
2. El Editor Nacional si desea asignar una planilla a un centro, entonces debe seleccionar el	2.1 Dependiendo de la operación a realizar: a) Si desea asignar una planilla a un centro ir a la

<p>tabview “Asignar Planillas”, si desea cambiar las planillas a un centro o eliminarle todas las planillas que tiene asignadas, debe seleccionar el tabview “Centros con Planillas Asignadas”.</p>	<p>sección “Asignar Planillas”.</p> <p>b) Si desea cambiar las planillas a un centro, ir a la sección “Cambiar plantillas a centro”.</p> <p>c) Si desea eliminar las planillas que tiene asignadas un centro, ir a la sección “Eliminar plantillas a centro”.</p>
Sección “Asignar Planillas”	
Acción del Actor	Respuesta del Sistema
<p>3. El Editor Nacional debe seleccionar planilla y los centros que van a realizar matrícula con esta, y pulsar el botón “Asignar”.</p>	<p>2.2 El sistema muestra la interfaz de “Asignar Planillas”.</p> <p>3.1 El sistema asigna a los centros las planillas por las cuales deberán realizar matrícula. Muestra un mensaje confirmando la operación realizada.</p>
Sección “Cambiar plantillas a centro”	
Acción del Actor	Respuesta del Sistema
<p>3. El Editor Nacional debe seleccionar del listado, el centro a cambiarle la planilla.</p> <p>4 El Editor Nacional debe cambiar la plantilla al centro, seleccionando la nueva plantilla y pulsando el botón “Cambiar”.</p>	<p>2.2 El sistema muestra la interfaz de “Centros con Planillas Asignadas”, que muestra el listado de centros y el formulario para cambiar la plantilla asignada.</p> <p>3.1 El sistema carga los datos del centro en el formulario, y las plantillas que existen.</p> <p>4.1 El sistema asigna la nueva planilla al centro y muestra un mensaje de confirmación.</p>
Sección “Eliminar plantillas a centro”	
Acción del Actor	Respuesta del Sistema
<p>3. El Editor Nacional debe seleccionar del listado, el centro para eliminarle las planillas y pulsar el botón “Eliminar”.</p>	<p>2.2 El sistema muestra la interfaz de “Centros con Planillas Asignadas”, que muestra el listado de centros.</p> <p>3.1 El sistema elimina las planilla al centro, e inmediatamente este desaparece del listado de centros a realizar matrícula; muestra un mensaje de confirmación.</p>
Flujos Alternos	

3 Si desea cancelar la operación pulsa el botón “Cancelar” y el sistema saldrá de la opción.	
Pos-condiciones	Se asignó una planilla de matrícula a un centro, o se cambió la planilla de un centro, o se eliminaron las planillas de un centro.

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

Caso de Uso.	Gestionar nomencladores.	
Actores.	Editor Nacional (inicia).	
Propósito.	Insertar, modificar o eliminar datos de un nomenclador.	
Resumen.	El caso de uso se inicia cuando el Editor Nacional inserta, modifica o elimina datos de un nomenclador para ser utilizado en la configuración de la planilla matrícula. Finaliza una vez que se insertan, modifican o eliminan estos datos.	
Precondiciones.	Editor Nacional autenticado satisfactoriamente por el sistema.	
Referencias.	R 9.	
Prioridad.	Crítico.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Editor Nacional selecciona la opción “Nomencladores”. 2. El Editor Nacional selecciona un nomenclador y pulsa el botón “Gestionar”.	1.1 El sistema muestra una página con la lista de los nomencladores a gestionar. 2.1 El sistema muestra la página de entrada de datos del nomenclador seleccionado, así como un listado de los que ya tiene. 2.2 Dependiendo de la operación a realizar: a) Si desea insertar datos a un nomenclador ir a la sección “Insertar datos a un nomenclador”. b) Si desea modificar los datos de un nomenclador ir a la sección “Editar datos a un nomenclador”. c) Si desea eliminar datos a un nomenclador ir a la sección “Eliminar datos a un nomenclador”.	
Sección “Insertar datos a un nomenclador”		
Acción del Actor	Respuesta del Sistema	

3. El Editor Nacional entra los datos requeridos y pulsa el botón "Adicionar".	3.1 El sistema adiciona los datos, muestra un mensaje confirmando la adición de los mismos, y los incorpora al listado de datos del nomenclador.
Sección "Editar datos a un nomenclador"	
Acción del Actor	Respuesta del Sistema
3. El Editor Nacional selecciona del listado de datos del nomenclador, el que desea modificar.	3.1 El sistema carga los datos del nomenclador en los campos de texto.
4. El Editor Nacional modifica los datos y pulsa el botón "Editar".	4.1 El sistema modifica los datos, muestra un mensaje confirmando la operación, y se muestran estos modificados en el listado de datos del nomenclador.
Sección "Eliminar datos a un nomenclador"	
Acción del Actor	Respuesta del Sistema
3. El Editor Nacional selecciona del listado de datos del nomenclador, el que desea eliminar y pulsa el botón "Eliminar".	4.1 El sistema elimina el dato del nomenclador. Muestra un mensaje confirmando la operación, e inmediatamente desaparece este del listado de datos del nomenclador.
Flujos Alternos	
3. Si desea retirar los datos entrados pulse el botón "Cancelar".	
Pos-condiciones	Se adicionan, modifican o eliminan datos del nomenclador al sistema.

Tabla 2.10 Descripción del CU "Gestionar nomencladores"

En este capítulo, se profundizó en los conceptos y definiciones necesarias para comprender el proceso de desarrollo del Componente Registro de Estudiantes como parte del Proceso de Formación de Recursos Humanos en Salud. Este componente se encarga de gestionar todo lo referente al registro de matrícula de los estudiantes tanto extranjeros como nacionales de la carrera de Ciencias Médicas en las instituciones del MINSAP. Además el sistema es configurable; permite que cada centro del país tenga una configuración propia del centro de los datos que desea tener del estudiante. Se estudiaron los aspectos teóricos necesarios para la realización de este componente. Se analizaron procesos actuales de matrícula y de registros de las secretarías docentes de las instituciones del MINSAP.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Este capítulo está dedicado a la etapa de desarrollo, donde se profundiza en el análisis del sistema. Se utiliza un lenguaje apropiado como UML para el entendimiento del mismo, dando paso posteriormente al diseño del sistema. El modelo de análisis ayuda a refinar los requisitos. Además ofrece un mayor poder expresivo y una mayor formalización, como por ejemplo, la que brindan los diagramas de interacción que se utilizan para describir los aspectos dinámicos del sistema. Tras las definiciones y descripciones abordadas en el anterior capítulo, se hace necesario definir cómo se desarrolla.

En esta etapa del capítulo 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. El modelo de análisis puede considerarse una primera aproximación al modelo de diseño, aunque es un modelo propio. Posteriormente, en el modelo de diseño, a través de los diagramas de interacción, se muestra una descripción gráfica de la interacción entre los actores y el sistema. También se describen las clases que se encuentran en el diseño. Igualmente se obtiene el Diagrama Entidad-Relación de la base de datos y la descripción de sus tablas.

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 Diagrama de clases del análisis

En estos diagramas se muestra la relación existente entre las clases de cada caso de uso desarrollado.

3.1.2 Paquete de Administración

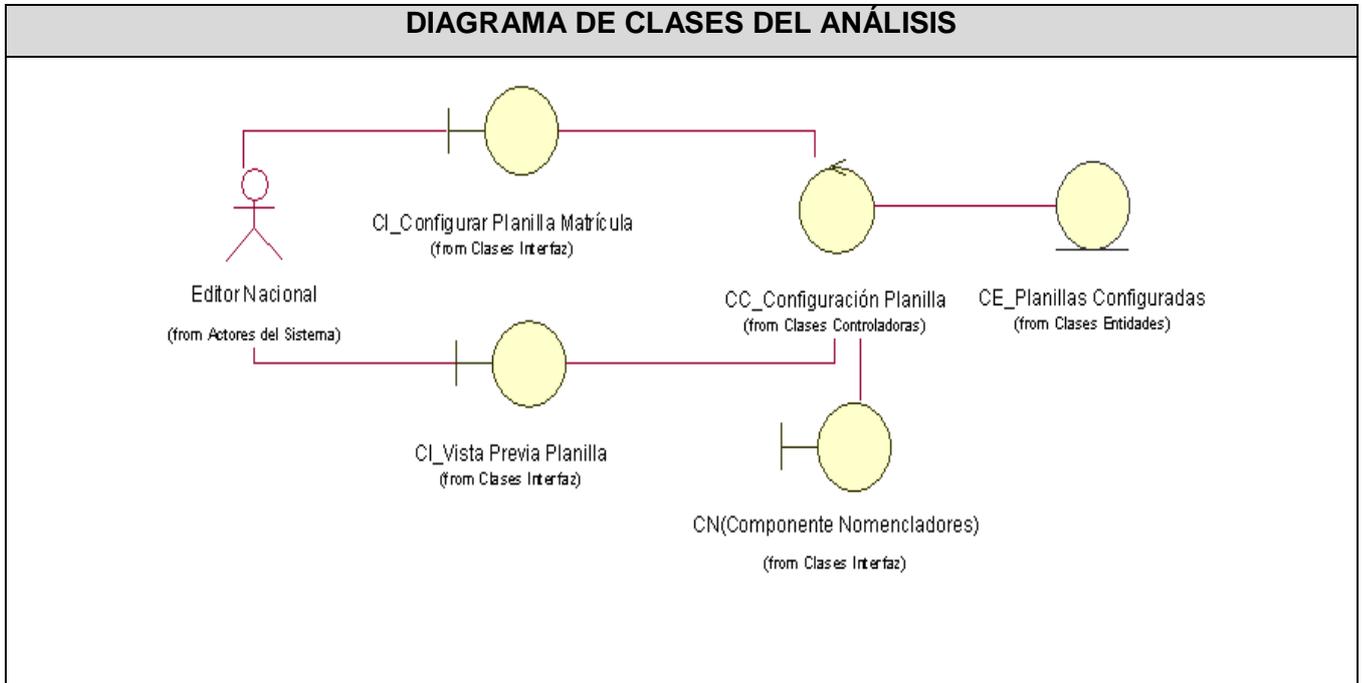


Fig.3.1 Diagrama de clases de análisis “CU Configurar Planilla Matrícula”

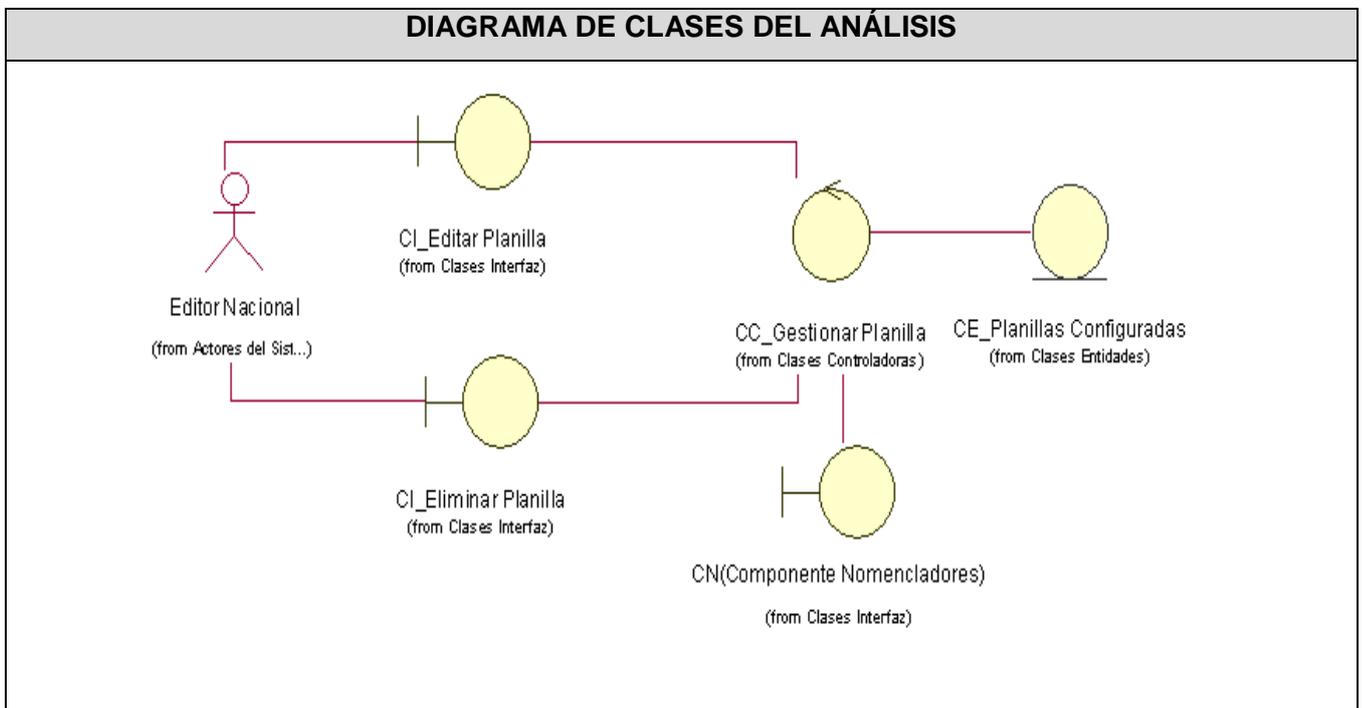


Fig. 3.2 Diagrama de clases de análisis “CU Gestionar Planilla Matrícula”

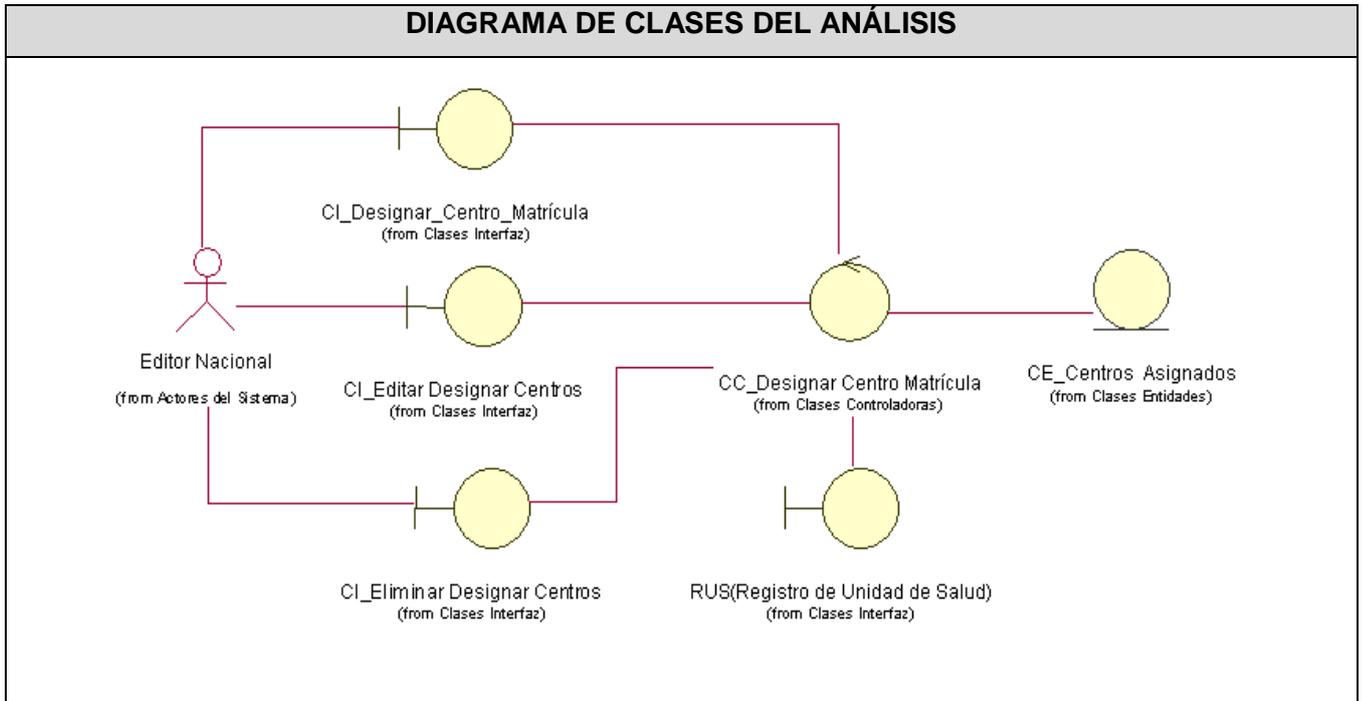


Fig. 3.3 Diagrama de clases de análisis "CU Designar Centro Matrícula"

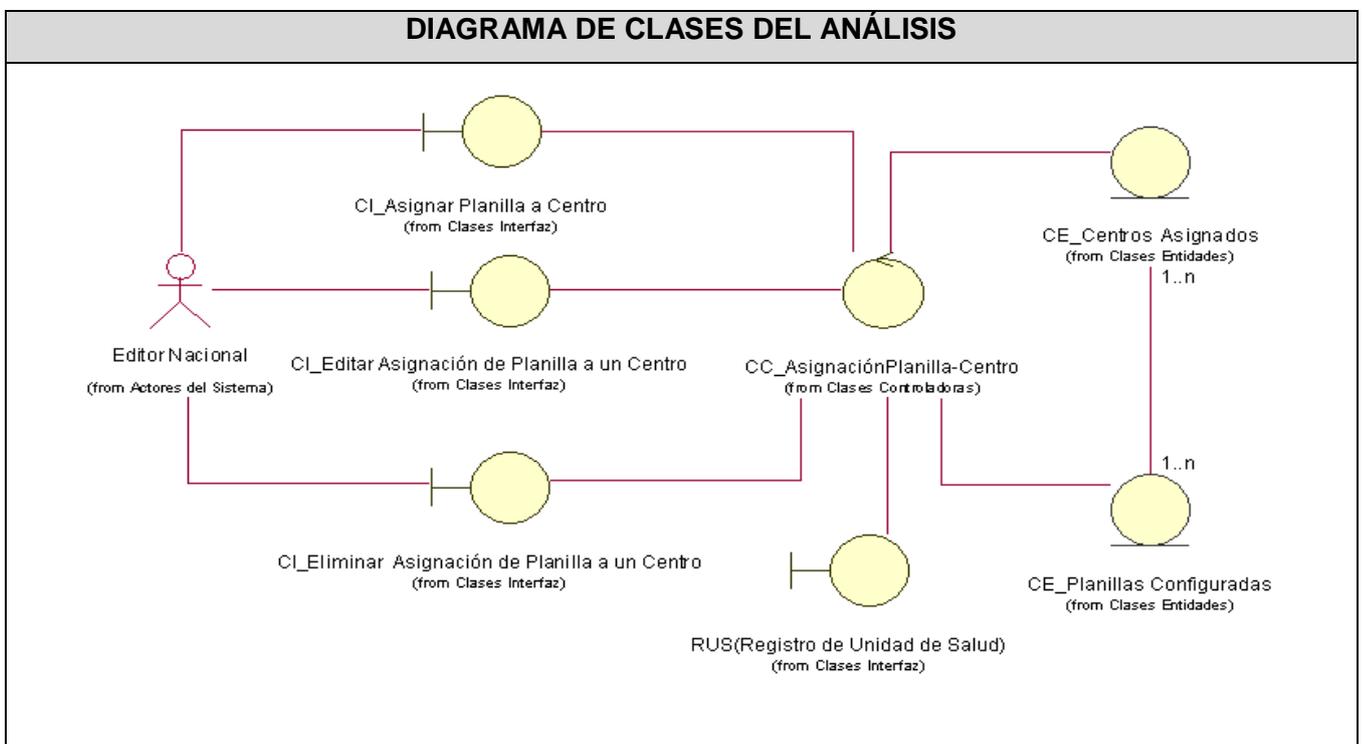


Fig. 3.4 Diagrama de clases de análisis "CU Asignar Planilla Matrícula a Centro"

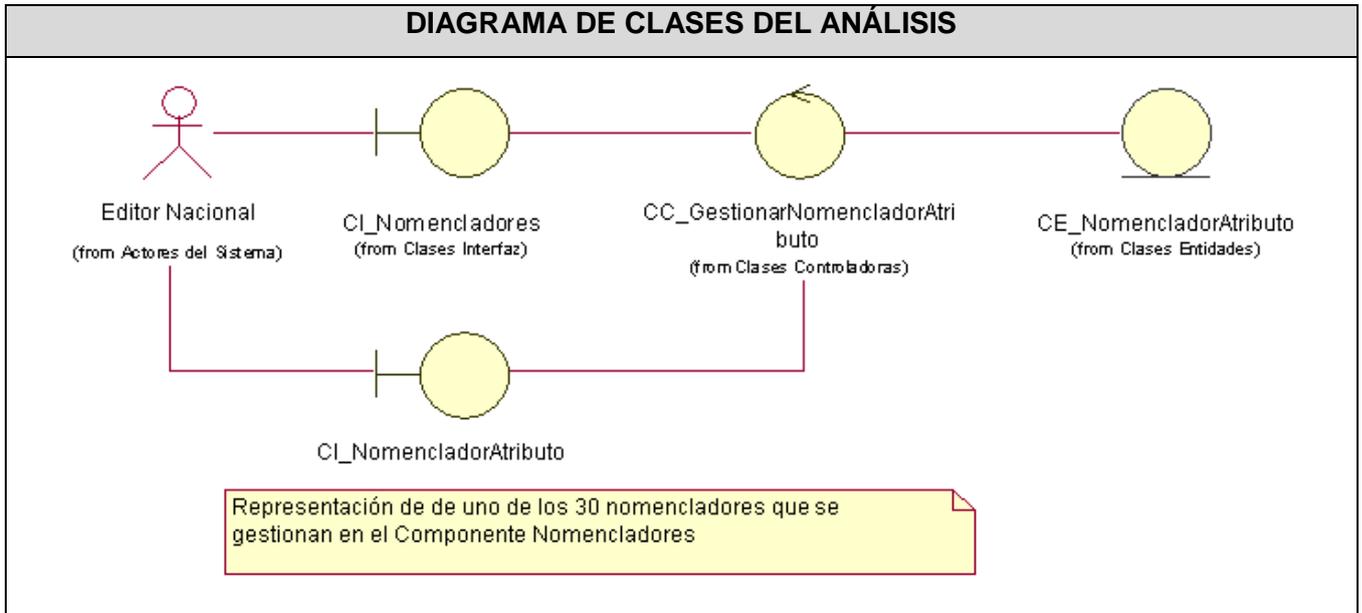


Fig. 3.5 Diagrama de clases de análisis “CU Gestionar Nomencladores”

3.1.3 Paquete de Matrícula

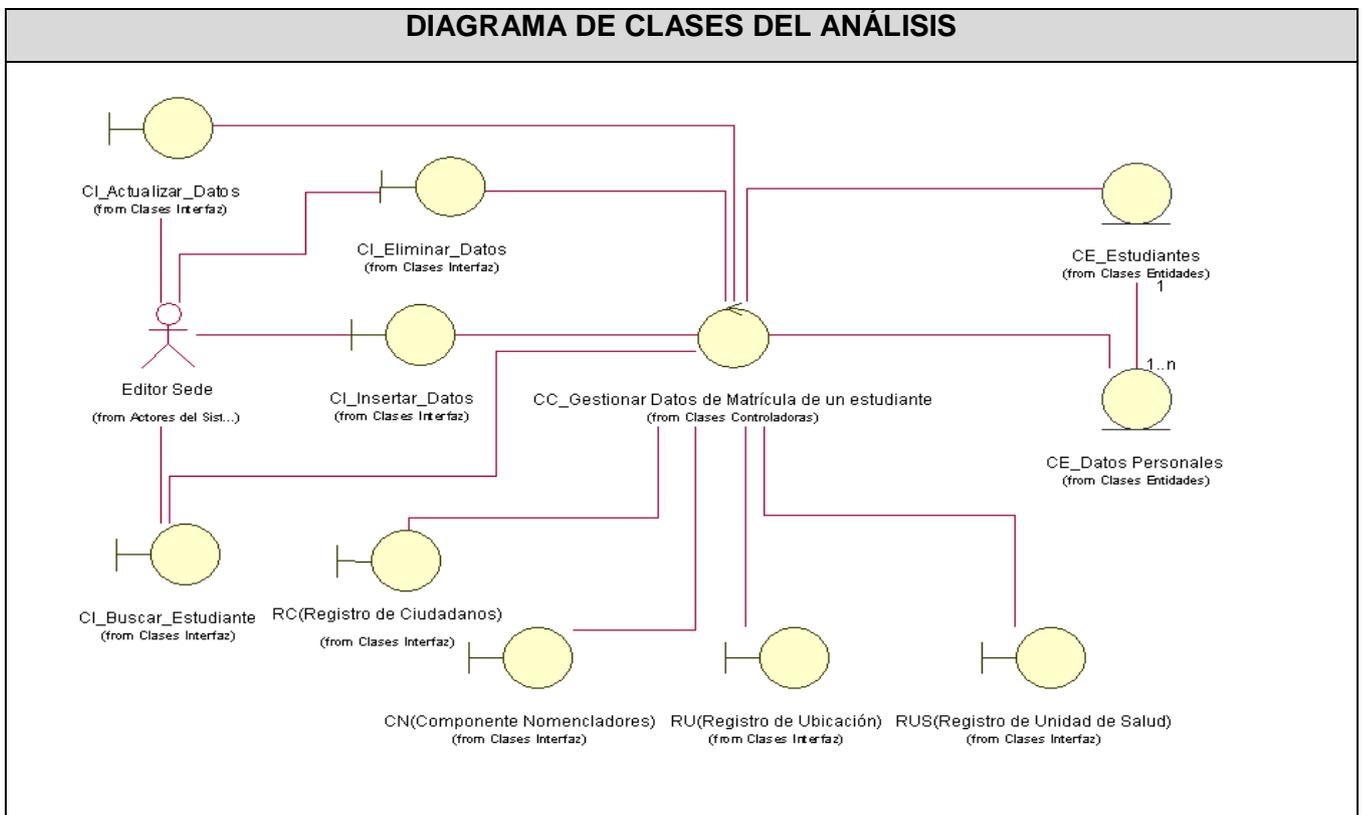


Fig. 3.6 Diagrama de clases de análisis “CU Gestionar Datos de Matrícula de un estudiante”

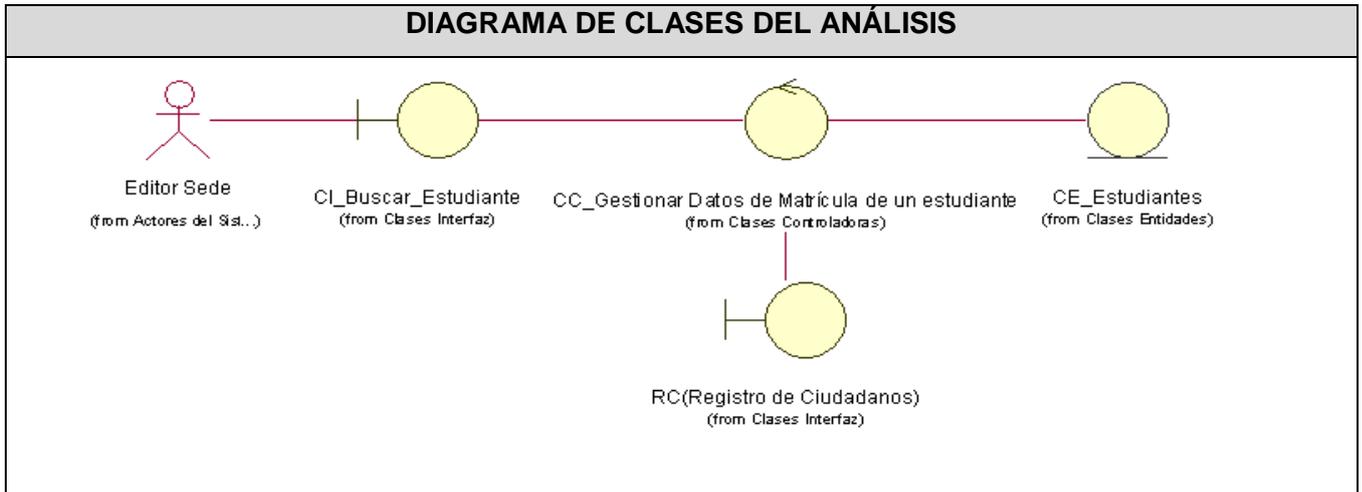


Fig. 3.7 Diagrama de clases de análisis “CU Buscar Estudiante”

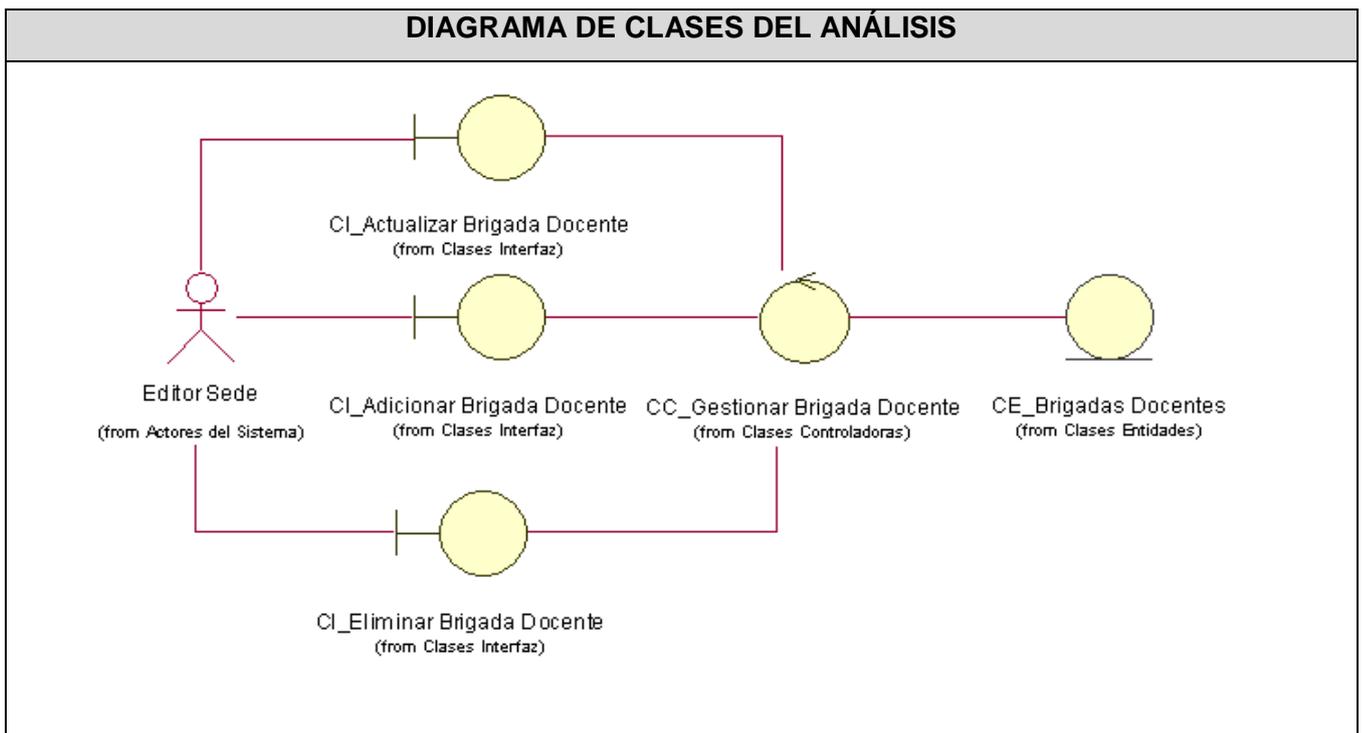


Fig. 3.8 Diagrama de clases de análisis “CU Adicionar Brigada Docente”

3.2 Diseño

Un modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso. Se centra en cómo los requerimientos funcionales y no funcionales, junto a otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema.

Esta etapa tiene como propósito modelar el sistema y encontrar su forma para que soporte todos los requisitos; lo que crea una entrada apropiada y un punto de partida para la implementación. El artefacto más importante que se obtiene en esta etapa es el modelo de diseño en el cual se desarrolla la realización de los casos de uso del sistema, mediante diagramas de clases y de interacción.

3.2.1 Diagrama de clases del diseño

En estos diagramas se muestra la relación existente entre las clases de cada caso de uso desarrollado. Estos se elaboran para tener en cuenta los detalles concretos de la implementación del sistema.

3.2.2 Paquete Administración

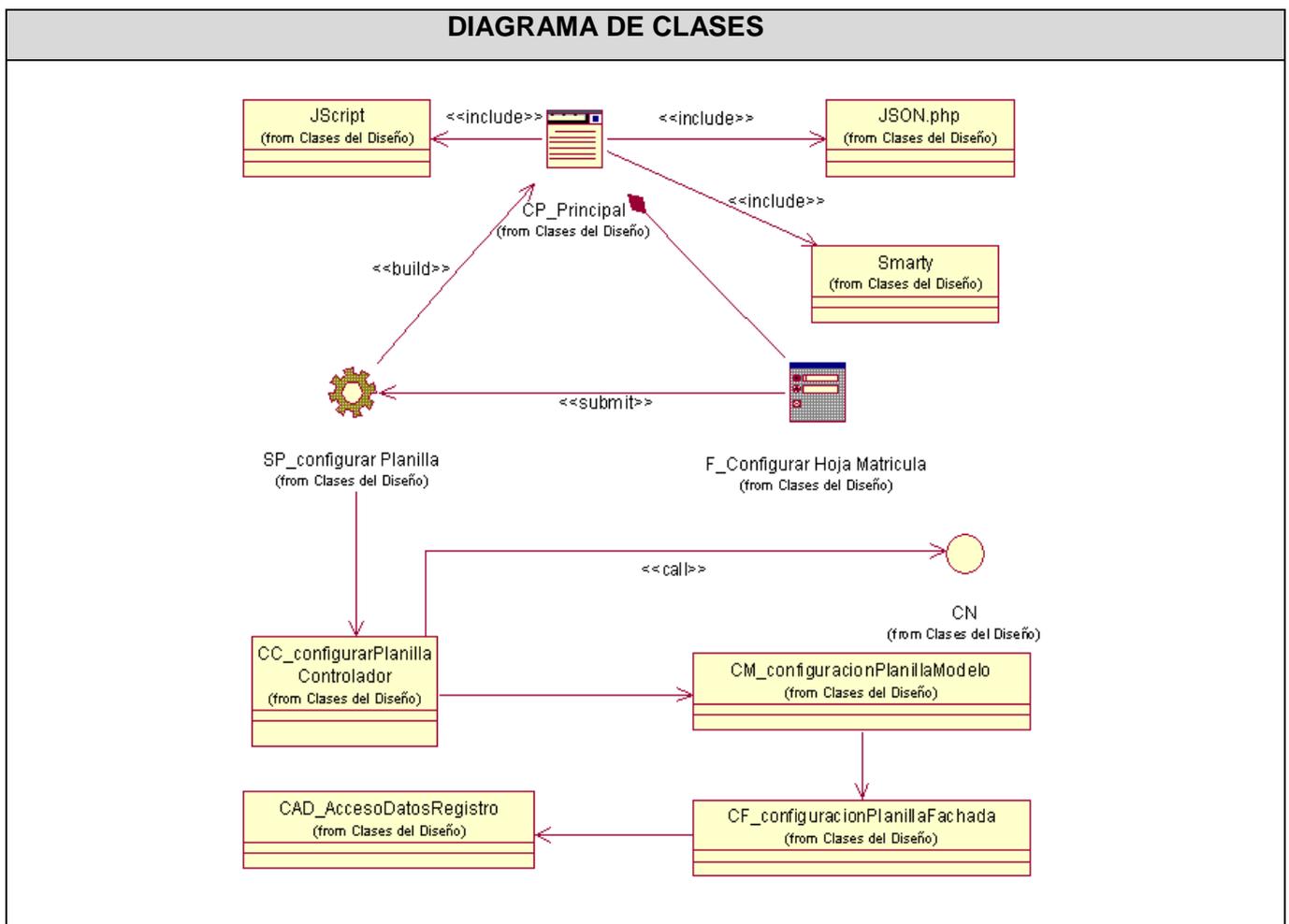


Fig. 3.9 Diagrama de clases de diseño “Configurar Planilla Matricula”

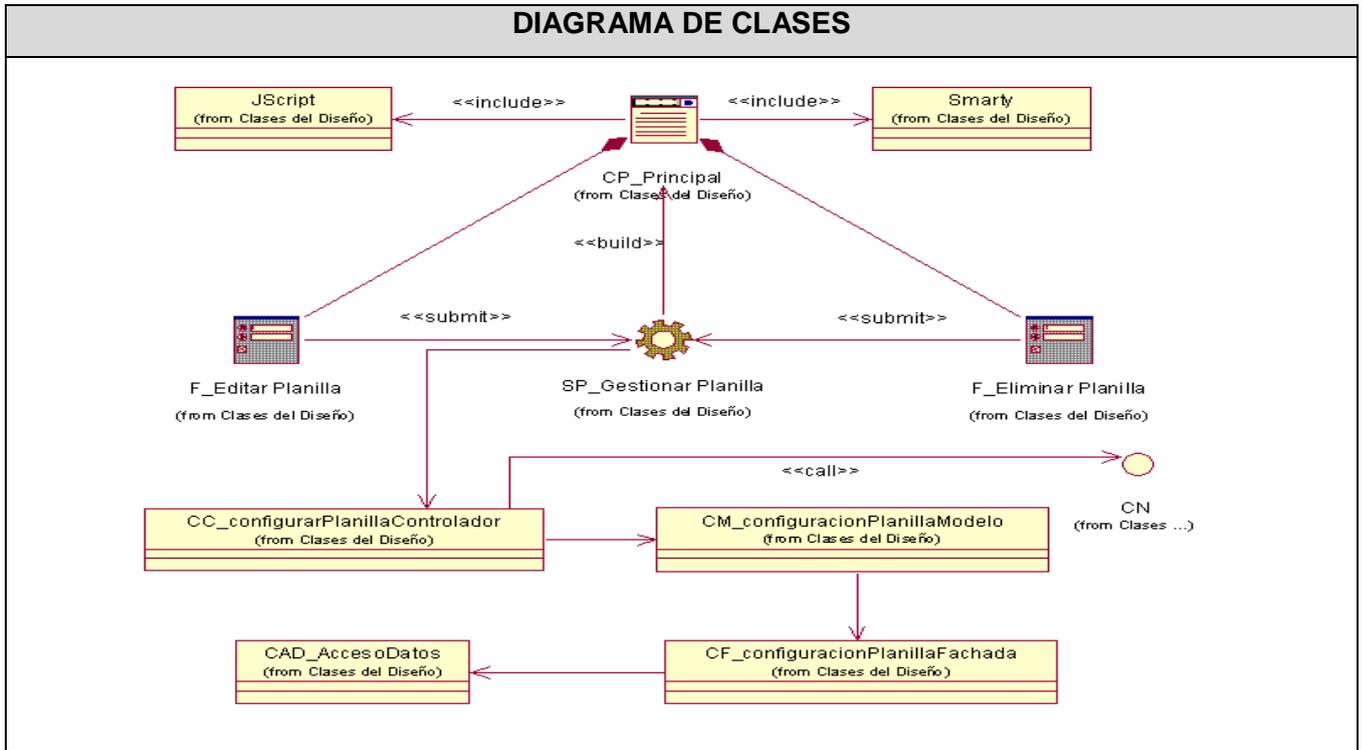


Fig. 3.10 Diagrama de clases de diseño "Gestionar Planilla Matrícula"

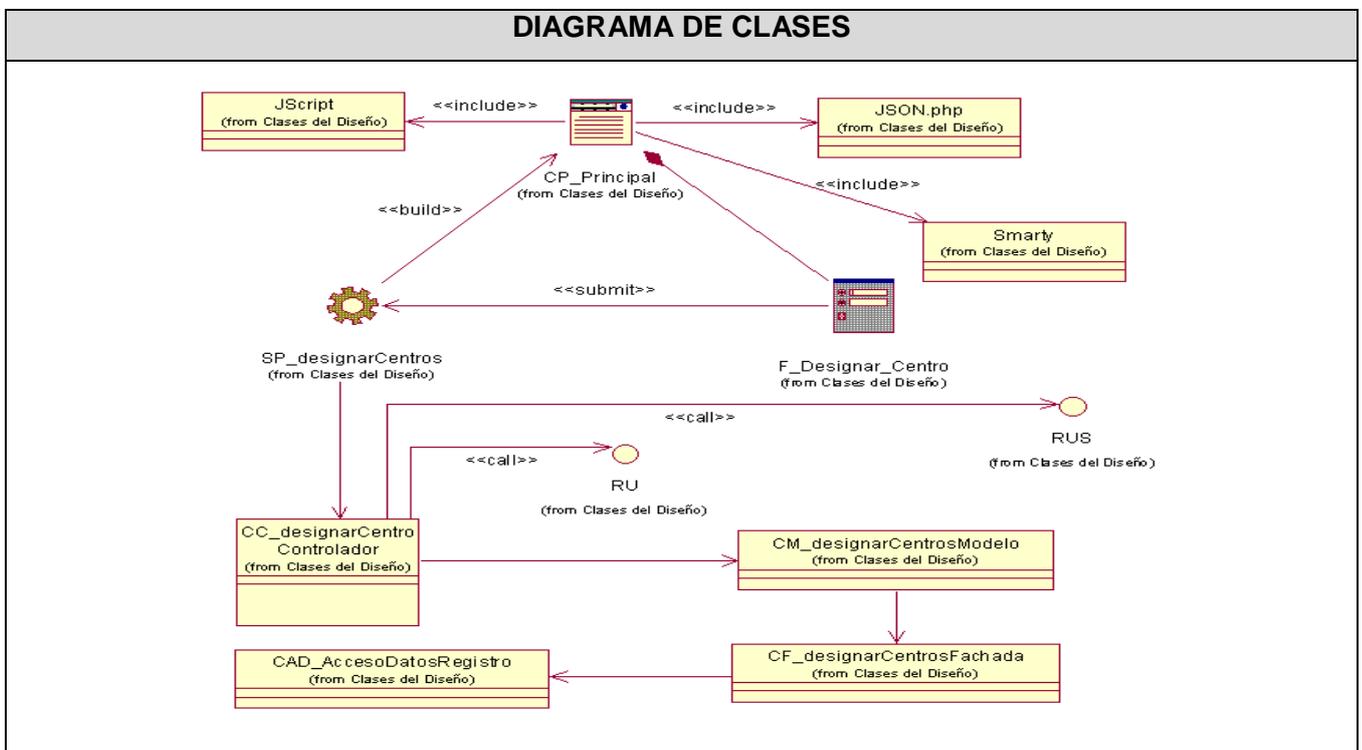


Fig. 3.11 Diagrama de clases de diseño "Designar Centro Matrícula"

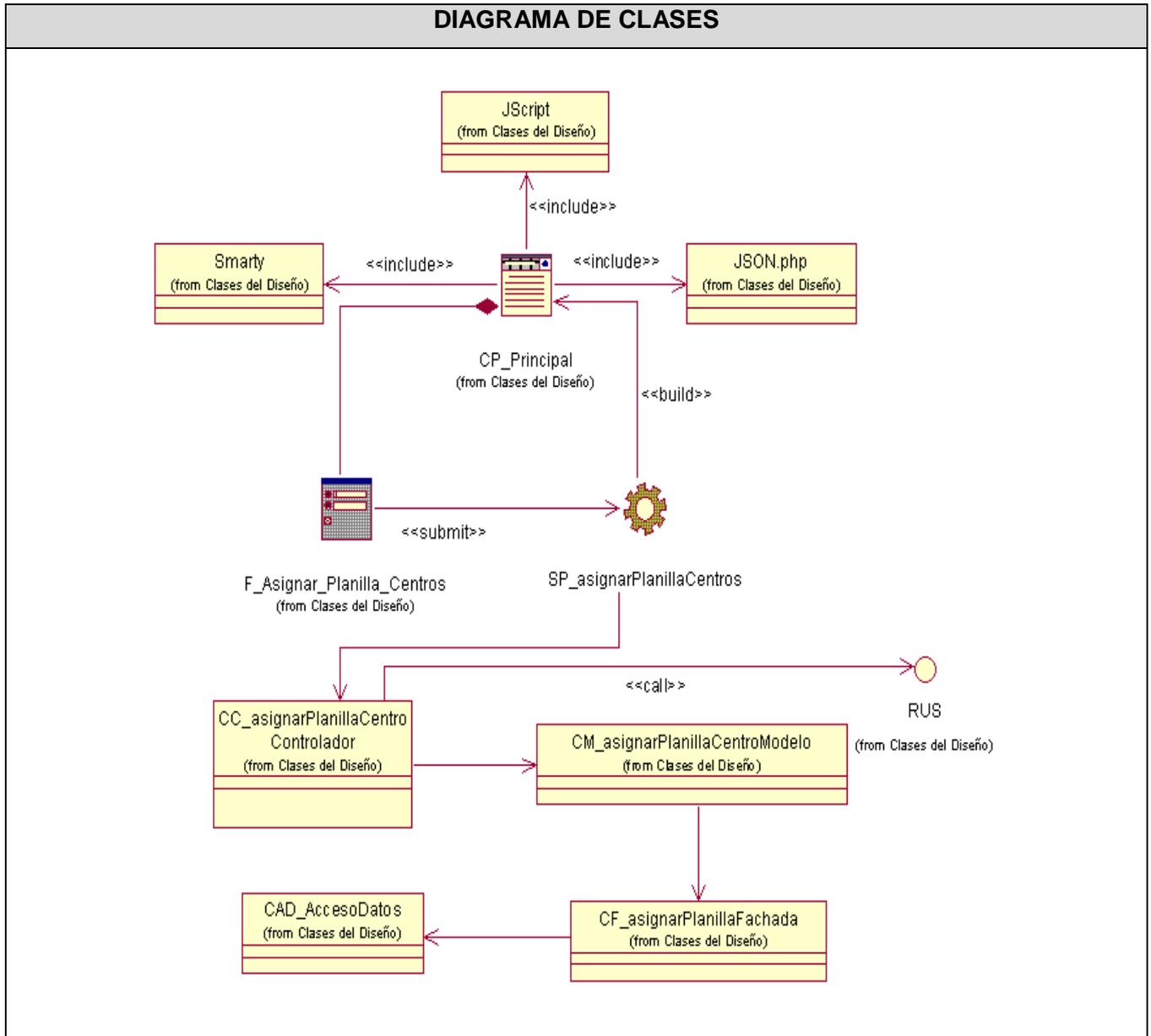


Fig. 3.12 Diagrama de clases de diseño “Gestionar Asignación de Planilla Matrícula a los Centros”

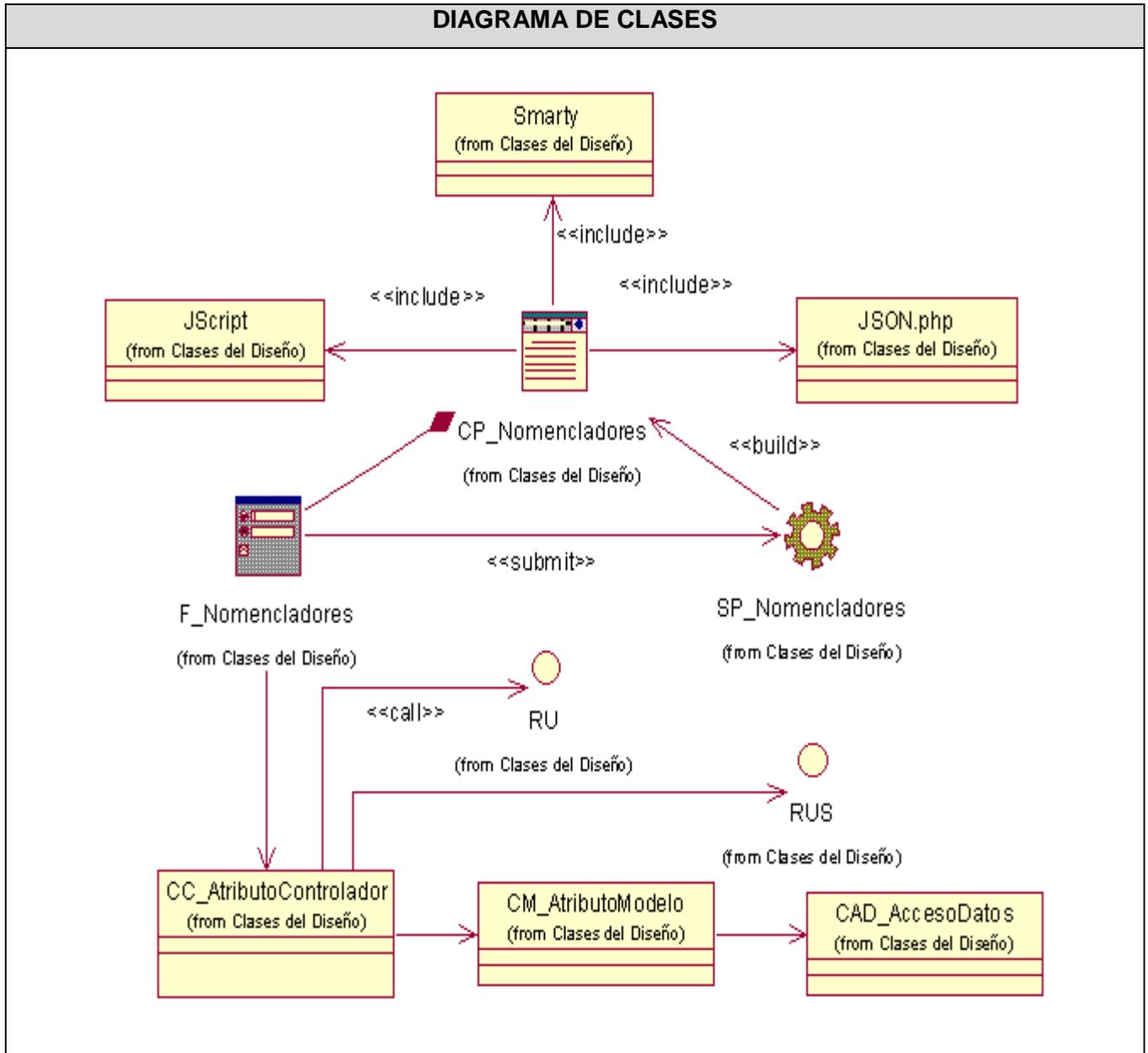


Fig. 3.13 Diagrama de clases de diseño "Gestionar Nomencladores"

3.2.3 Paquete de Matrícula

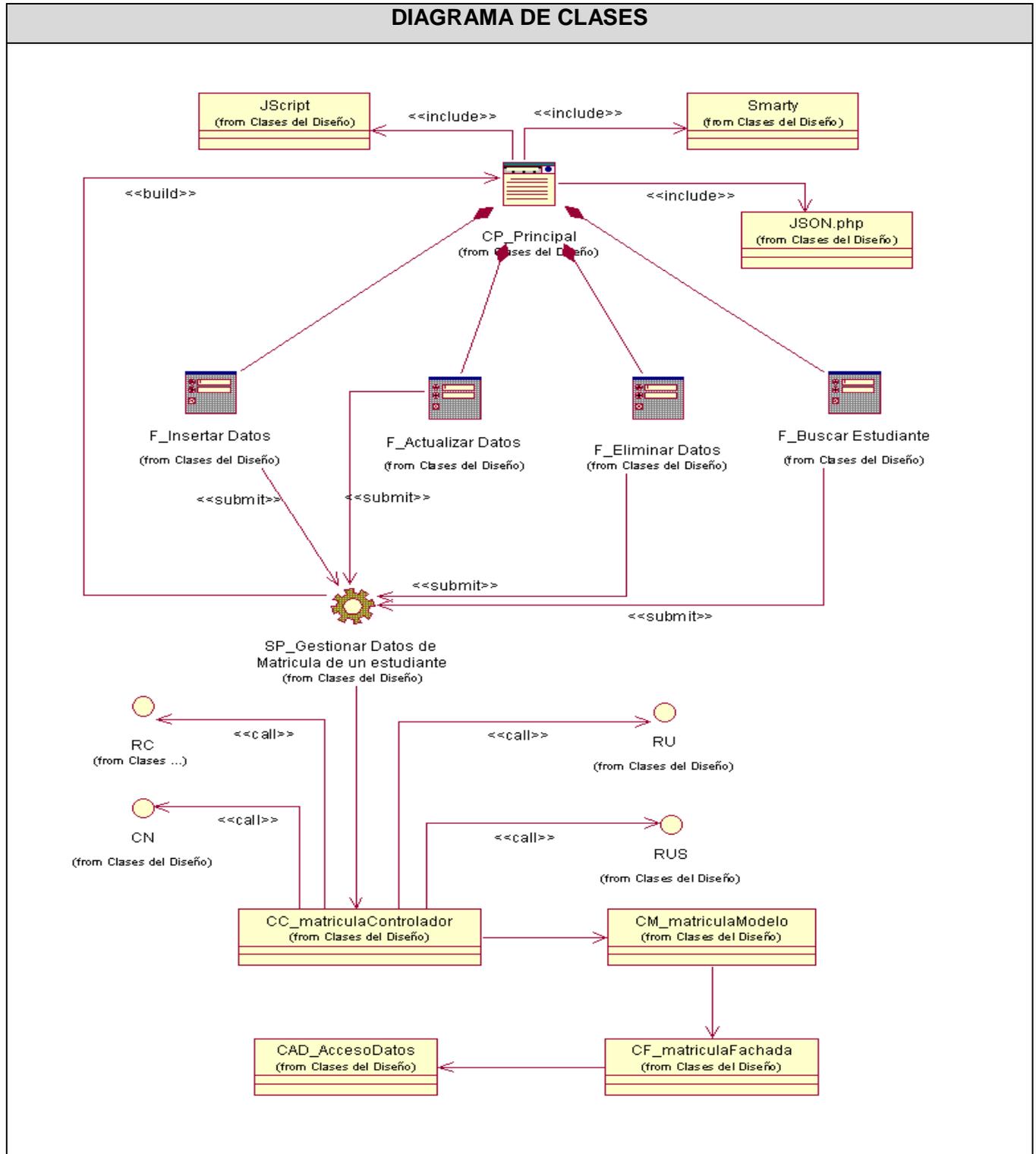


Fig. 3.14 Diagrama de clases de diseño “Gestionar Datos de Matrícula de un estudiante”

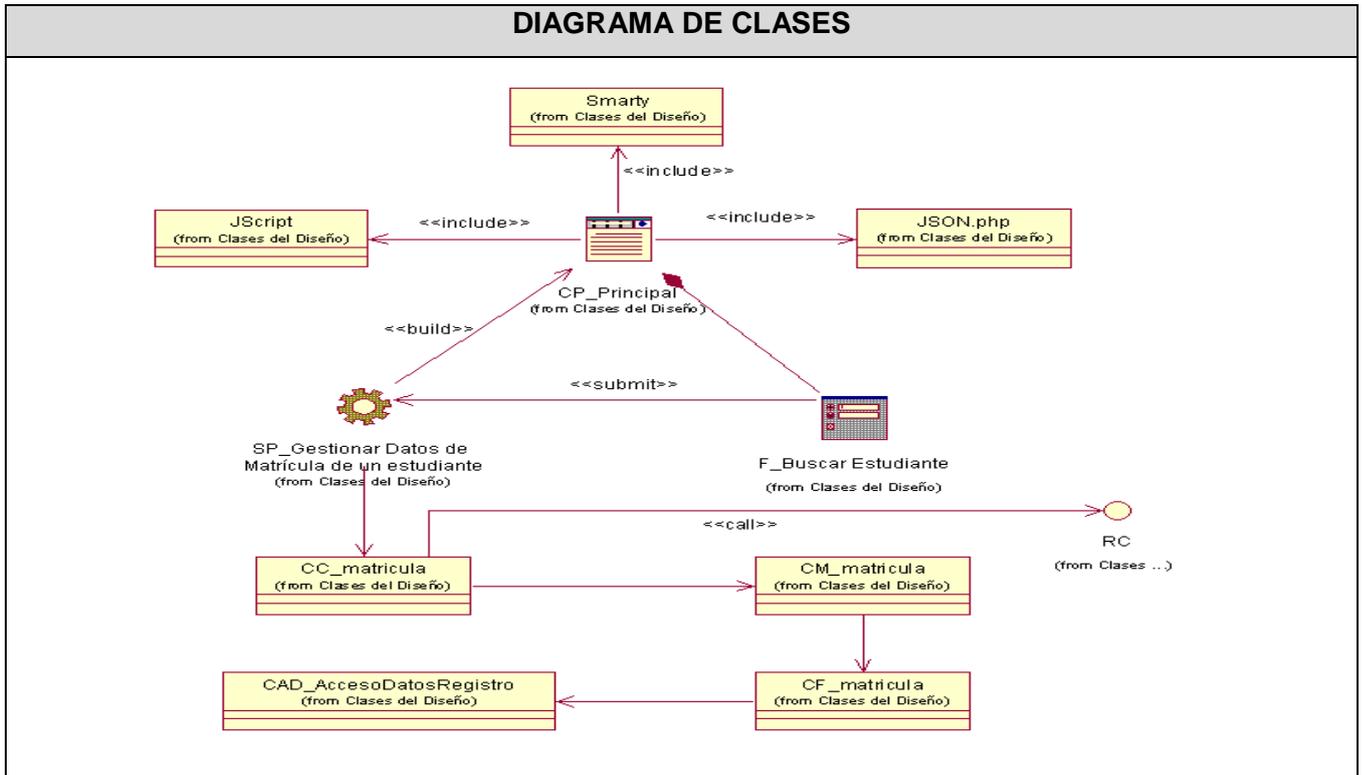


Fig. 3.15 Diagrama de clase de diseño “Buscar Estudiante”

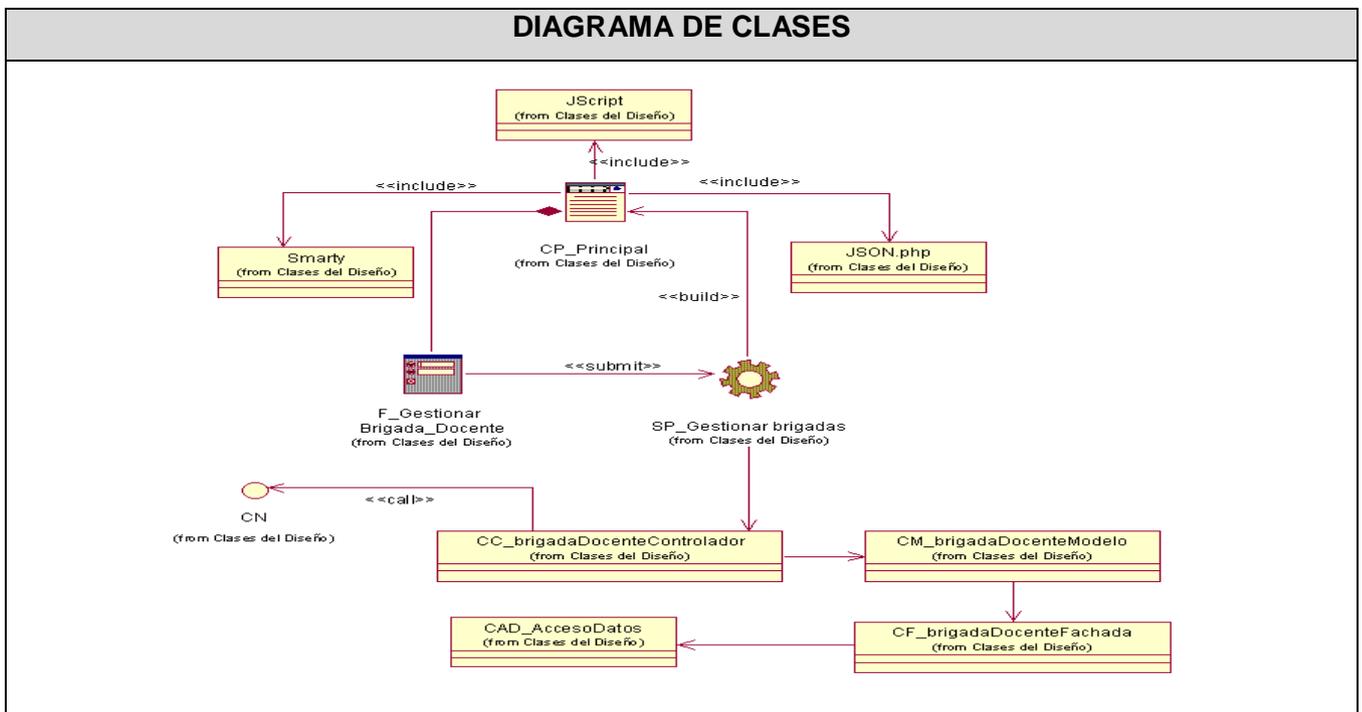


Fig. 3.16 Diagrama de clases de diseño “Gestionar Brigada Docente”

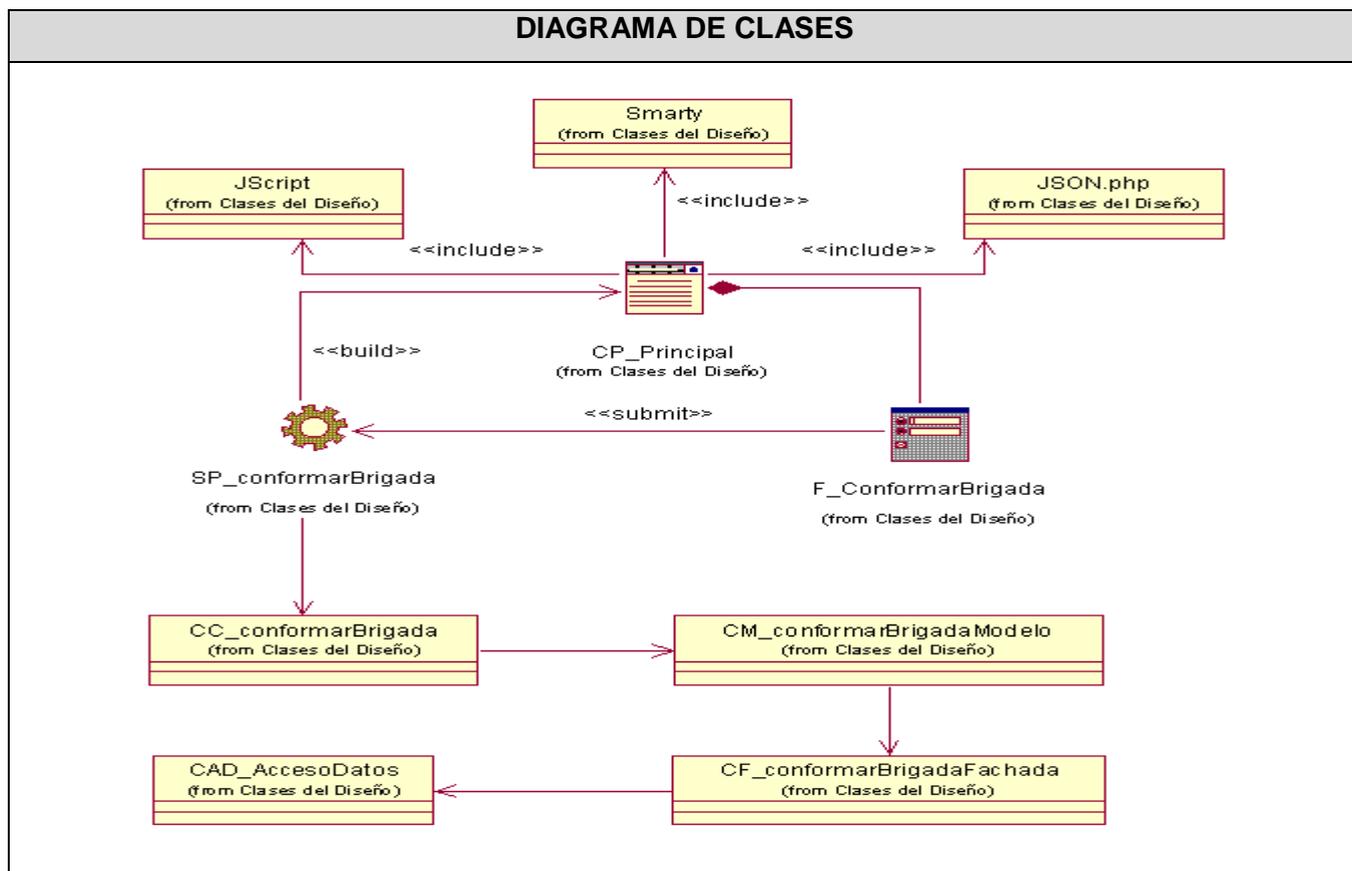


Fig. 3.17 Diagrama de clases de diseño “Gestionar Conformar Brigada Docente”

3.3 Descripción de las clases

Seguidamente serán explicadas algunas de las clases que han sido identificadas para la futura implementación del componente. La estructura general de la Capa de Presentación para cada uno de los casos de uso identificados en el Componente Registro de Estudiantes es similar, contando con las mismas páginas clientes y servidoras. Del mismo modo serán descritas algunas de las responsabilidades que realizarán las páginas servidoras que responden a la Lógica de Negocio. De esta manera se tendrá una comprensión mayor del funcionamiento del sistema en desarrollo.

Nombre: tb_escuela.	
Tipo de clase: Entidad.	
Atributo	Tipo
id_unidad.	Integer.

id_curso.	Integer.
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.1 Descripción de la clase entidad "tb_escuela"

Nombre: tb_escuela_planilla.	
Tipo de clase: Entidad.	
Atributo	Tipo
id_unidad.	Integer.
id_planilla.	Integer.
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.2 Descripción de la clase entidad "tb_escuela_planilla"

Nombre: tb_planilla	
Tipo de clase: Entidad.	
Atributo	Tipo
nombre_planilla.	String.
id_planilla.	Integer.
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.3 Descripción de la clase entidad "tb_planilla"

Nombre: tb_planilla_atributo.	
Tipo de clase: Entidad.	
Atributo	Tipo

id_planilla_atributo.	Integer.
id_atributo.	Integer.
id_planilla.	Integer.
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.4 Descripción de la clase entidad “tb_planilla_atributo”

Nombre: tb_brigada_docente.	
Tipo de clase: Entidad.	
Atributo	Tipo
id_brigada.	Integer.
nombre_brigada.	String.
matrícula_total.	Integer.
idioma.	String.
id_estudio.	Integer.
id_curso_escuela.	Integer.
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.5 Descripción de la clase entidad “tb_brigada_docente”

Nombre: tb_curso_escuela.	
Tipo de clase: Entidad.	
Atributo	Tipo
id_curso_escuela.	Integer.
id_unidad.	Integer.
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.6 Descripción de la clase entidad “tb_curso_escuela”

Nombre: tb_brigada_estudiante_matricula.	
Tipo de clase: Entidad.	
Atributo	Tipo
id_matricula.	Integer.
id_curso_escuela.	Integer.
id_ciudadano.	Integer.
id_estudiante.	Integer.
tipo_curso.	String.
fecha_matricula.	Date.
anno_académico.	Integer.
id_estudio.	Integer.
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.7 Descripción de la clase entidad "tb_brigada_estudiante_matricula"

Nombre: tb_estudiante.	
Tipo de clase: Entidad.	
Atributo	Tipo
id_estudiante.	Integer.
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.8 Descripción de la clase entidad "tb_estudiante"

Nombre: tb_estudiante_matricula.	
Tipo de clase: Entidad.	
Atributo	Tipo
id_matricula.	Integer.
id_curso_escuela.	Integer.
id_estudiante.	Integer.
tipo_curso.	String.
fecha_matricula.	Date.

anno_académico.	Integer.
id_estudio.	Integer.
anno_estudiante.	String.
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.9 Descripción de la clase entidad “tb_estudiante_matricula”

Nombre: tb_datos_estudiante.	
Tipo de clase: Entidad.	
Atributo	Tipo
id_atributo.	Integer.
id_estudiante.	Integer.
datos_estudiante.	String.
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.10 Descripción de la clase entidad “tb_datos_estudiante”

Nombre: tb_atributo.	
Tipo de clase: Entidad.	
Atributo	Tipo
id_atributo.	Integer.
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.11 Descripción de la clase entidad “tb_atributo”

Nombre: insertarEstudiante.	
Tipo de clase: Controladora.	
Atributo	Tipo
idEstudiante.	Integer.

idCiudadano.	Integer.
sNombre.	String.
sSegundoNombre.	String.
sApellido.	String.
sSegundoApellido.	String.
iCarnetIdentidad.	Integer.
sSexo.	String.
sColorPiel.	String.
dFechaNacimiento.	Date.
bEsCreyente.	Bolean.
sEstadoCivil.	String.
iNoHijos.	Integer.
idProcedenciaSocial.	String.
idReligion.	Integer.
idModeloFormacion.	Integer.
Responsabilidad:	
Nombre:	
Descripción:	Se encarga de insertar un nuevo estudiante cubano en el sistema, para lo cual se le pasa como parámetro los datos del estudiante.

Tabla 3.12 Descripción de la clase controladora “insertarEstudiante”

Nombre: configurarPlanillaControlador.	
Tipo de clase: Controladora.	
Atributo	Tipo
idPlanilla.	Integer.
idAtributo.	Integer.
sNombrePlanilla.	String.
Responsabilidad:	
Nombre:	ListarPlanilla()
Descripción:	Muestra una lista de las planillas que ya han sido configuradas por el sistema.
Responsabilidad:	
Nombre:	Editar Planilla (idPlanilla)
Descripción:	Da la posibilidad de cambiar los datos de la Planilla que se desea editar.

Nombre:	AdicionarPlanilla (idAtributo,sNombrePlanilla).
Descripción:	Crea una nueva planilla con los atributos seleccionados.
Nombre:	EliminarPlanilla(idPlanilla)
Descripción:	Se elimina la planilla seleccionada.

Tabla 3.13 Descripción de la clase controladora “configurarPlanillaControlador”

Nombre: brigadaDocenteControlador.	
Tipo de clase: Controladora.	
Atributo	Tipo
idBrigada	Integer
sNombreBrigada	String
iMatriculaTotal	Integer
sAnno	String
idEstudio	Integer
sModelo	String
Responsabilidad:	
Nombre:	Listar()
Descripción:	Muestra una lista de las brigadas que ya han sido creadas.
Nombre:	Adicionar(sNombreBrigada,iMatriculaTotal,sAnno,idio,idEstudio,sModelo)
Descripción:	Se adiciona una Brigada Docente.
Nombre:	Editar(idBrigada, sNombreBrigada,iMatriculaTotal,sAnno,idio,idEstudio,sModelo)
Descripción:	Se actualizan los datos de una brigada.
Nombre:	Eliminar (idBrigada)
Descripción:	Se elimina la brigada docente seleccionada.

Tabla 3.14 Descripción de la clase controladora “brigadaDocenteControlador”

Nombre: designarCentroControlador.	
Tipo de clase: Controladora	
Atributo	Tipo
Provincia	Integer
Municipio	Integer
idCentro	Integer
idCurso	Integer
Responsabilidad:	
Nombre:	Listar()
Descripción:	Muestra una lista de los centros que han sido designados para realizar matrícula.
Nombre: Editar(ID_Centro, ID_Curso)	
Descripción:	Se edita un centro cambiando la información de su curso docente.
Nombre: SelectMunicipio(Provincia)	
Descripción:	Muestra los municipios dada una provincia.
Nombre: SelectCentros(Municipio)	
Descripción:	Muestra los centros que tiene un municipio.
Nombre: Eliminar(ID_Centro)	
Descripción:	Elimina un centro seleccionado.

Tabla 3.15 Descripción de la clase controladora “designarCentroControlador”

Nombre: asignarPlanillaCentroControlador	
Tipo de clase: Controladora	
Atributo	Tipo
idPlanilla	Integer
idCentro	Integer
Responsabilidad:	
Nombre:	ListarCentrosconPlanilla()

Descripción:	Muestra una lista de los centros que ya le han asignado una planilla.
Nombre:	Editar(idPlanilla)
Descripción:	Se cambia la planilla al centro.
Nombre:	Eliminar(idCentro)
Descripción:	Se le quita la planilla al centro.

Tabla 3.16 Descripción de la clase controladora “asignarPlanillaCentroControlador”

Nombre: Asignar_Planilla_Centros	
Tipo de clase: Interfaz	
Atributo	Tipo
Centro	String
Planilla	String
Responsabilidad:	
Nombre:	asignarPlanillaCentros()
Descripción:	Indica el proceso de asignarle una planilla de matrícula a un centro.

Tabla 3.17 Descripción de la clase controladora “Asignar_Planilla_Centros”

Nombre: Designar_Centro_Matricula	
Tipo de clase: Interfaz	
Atributo	Tipo
Provincia	String
Municipio	String
Responsabilidad:	
Nombre:	designarCentro()
Descripción:	Indica el proceso de designar centros que realizarán matrícula.

Tabla 3.18 Descripción de la clase controladora “Designar_Centro_Matricula”

Nombre: Crear_Brigada_Docente	
Tipo de clase: Interfaz	
Atributo	Tipo
sNombreBrigada	String
sMatrículaTotal	Integer
sAnno	String
idioma	String
estudio	String
sModeloFormacion	String
Responsabilidad:	
Nombre:	adicionarBrigadaDocente()
Descripción:	Es el proceso de adicionar una brigada docente al centro.

Tabla 3.19 Descripción de la clase controladora "Crear_Brigada_Docente"

Nombre: InsertarDatosEstudiante	
Tipo de clase: Interfaz	
Atributo	Tipo
sNombre	String
sSegundoNombre	String
sApellido	String
sSegundoApellido	String
iCedula	Integer
iCarnetIdentidad	Integer
sSexo	String
sColorPiel	String
dFechaNacimiento	Date
sEsCreyente	String
sEstadoCivil	String
iNoHijos	Integer
iProcedenciaSocial	String
sReligion	String
sOrganizacionPolitica	String
sModeloFormacion	String

sDireccion	String
sPaisProcedencia	String
sEmail	String
Responsabilidad:	
Nombre:	insertarEstudiante()
Descripción:	Es el proceso de insertar un estudiante; matricularlo en un centro.

Tabla 3.20 Descripción de la clase controladora “InsertarDatosEstudiante”

3.4 Diseño de la BD de Registro de Estudiante

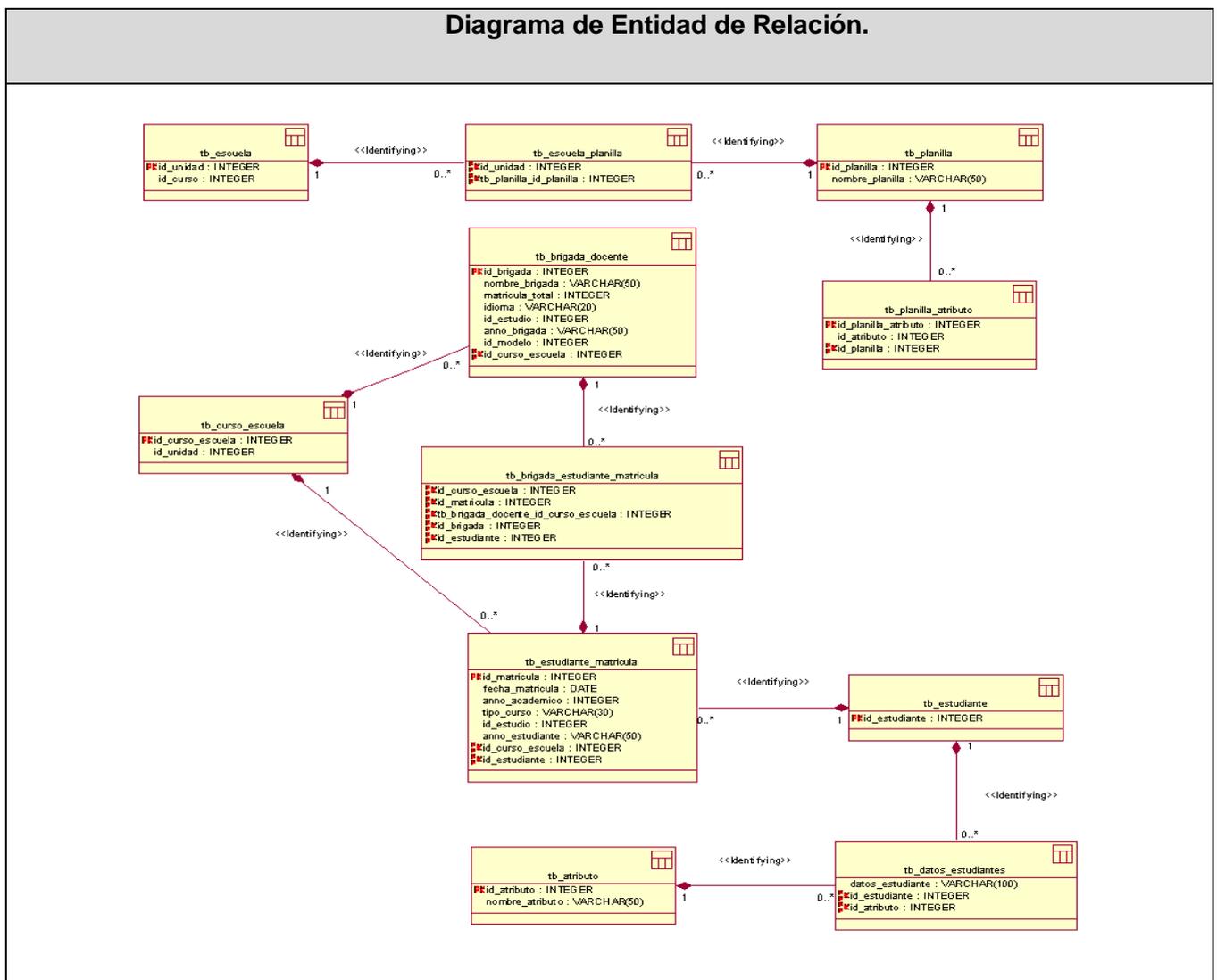


Fig. 3.18 Modelo de Datos “bd_RegistroEstudiantes”

3.4.1 Descripción de las tablas de la BD Registro de Estudiante

Nombre: tb_estudiante.		
Descripción: Esta tabla contiene la generalización de los datos de los estudiantes extranjeros y cubanos.		
Atributo	Tipo	Descripción
id_estudiante	Integer	Representa el identificador del estudiante.

Tabla 3.21 Descripción de la tabla “tb_estudiante”

Nombre: tb_estudiante_matricula.		
Descripción: Esta tabla contiene la información referente a la matrícula del estudiante.		
Atributo	Tipo	Descripción
id_estudiante	Integer	Representa el identificador del estudiante. Este atributo es llave foránea que la absorbe de la tabla tb_estudiante.
id_matricula	Integer	Representa el identificador de la matrícula.
fecha_matricula	Date	Representa la fecha de matrícula del estudiante.
anno_academico	Integer	Representa el año académico en que el estudiante matrícula.
tipo_curso	Varchar	Representa el tipo de curso que el estudiante matrícula. Este puede ser, Regular diurno, Por encuentro o Curso por trabajadores.
id_estudio	Integer	Representa el identificador de estudio del estudiante. Este atributo es absorbido de la tabla tb_nomen_estudio.
anno_estudiante	Varchar	Representa el año que está cursando el estudiante.
id_curso_escuela	Integer	Representa el identificador del curso. Este atributo es llave foránea que la absorbe de la tabla tb_curso_escuela.

Tabla 3.22 Descripción de la tabla “tb_estudiante_matricula”

Nombre: tb_brigada_docente.		
Descripción: Esta tabla contiene la información de las brigadas docentes donde se encuentra el estudiante.		
Atributo	Tipo	Descripción
id_brigada	Integer	Representa el identificador de la brigada.
id_curso_escuela	Integer	Representa el identificador de los cursos. Este atributo es llave foránea absorbida de la tabla tb_curso_escuela.
id_modelo	Integer	Representa el identificador del modelo de formación. Este atributo es absorbido de la tabla tb_nomen_modelo_formacion.
nombre_brigada	Varchar	Representa el nombre de la brigada.
matricula_total	Integer	Representa la matrícula total de la brigada.
idioma	Varchar	Representa el idioma que se habla en la brigada.
id_estudio	Integer	Representa el código de la carrera del estudiante. Este atributo es una llave foránea absorbida de la tabla tb_nomen_estudio.
anno_brigada	Varchar	Representa el año de la carrera del estudiante en la brigada.

Tabla 3.23 Descripción de la tabla “tb_brigada_docente”

Nombre: tb_curso_escuela.		
Descripción: Esta tabla contiene la información de los cursos de los estudiantes.		
Atributo	Tipo	Descripción
id_curso_escuela	Integer	Representa el identificador de los cursos del estudiante.
id_unidad	Integer	Representa el codificador de la unidad de salud de la escuela dentro del registro de unidad de salud. Este atributo es absorbido de la tabla tb_escuela.

Tabla 3.24 Descripción de la tabla “tb_curso_escuela”

Nombre: tb_escuela.		
Descripción: Esta tabla contiene la información de las escuelas de los estudiantes.		
Atributo	Tipo	Descripción
id_unidad	Integer	Representa el identificador de unidad de salud de la escuela.
id_curso	Integer	Representa el identificador de los cursos docentes. Este atributo es absorbido de tb_nomen_curso_docente.

Tabla 3.25 Descripción de la tabla “tb_escuela”

Nombre: tb_escuela_planilla.		
Descripción: Esta tabla contiene la información de las planillas que son asignadas a las escuelas.		
Atributo	Tipo	Descripción
id_unidad	Integer	Representa el identificador de la unidad de salud de la escuela. Este atributo es llave foránea absorbida de la tabla tb_escuela.
id_planilla	Integer	Representa el identificador de las planillas. Este atributo es llave foránea absorbida de la tabla tb_planilla.

Tabla 3.26 Descripción de la tabla “tb_escuela_planilla”

Nombre: tb_planilla.		
Descripción: Esta tabla contiene los nombres de las planillas de cada escuela.		
Atributo	Tipo	Descripción
id_planilla	Integer	Representa el identificador de la planilla.
nombre_planilla	Varchar	Representa el nombre de la planilla.

Tabla 3.27 Descripción de la tabla “tb_planilla”

Nombre: tb_planilla_atributo.		
Descripción: Esta tabla contiene los atributos de una planilla.		
Atributo	Tipo	Descripción
id_planilla_atributo	Integer	Representa el identificador del atributo de una planilla.
id_planilla	Integer	Representa el identificador de las planillas. Este atributo es llave foránea absorbida de la tabla tb_planilla.
id_atributo	Integer	Representa el identificador de los atributos. Este atributo es absorbido de la tabla tb_nomen_atributo.

Tabla 3.28 Descripción de la tabla “tb_planilla_atributo”

Nombre: tb_brigada_estudiante_matricula.		
Descripción: Esta tabla contiene la información de las brigadas con los estudiantes matriculados.		
Atributo	Tipo	Descripción
id_brigada	Integer	Representa el identificador de la brigada. Este atributo es llave foránea de la tabla tb_brigada_docente.
id_curso_escuela	Integer	Representa el identificador de los cursos del estudiante. Este atributo es llave foránea de la tabla tb_curso_escuela.
id_matricula	Integer	Representa el identificador de la matrícula. Este atributo es llave foránea de la tabla tb_estudiante_matricula.
id_estudiante	Integer	Representa el identificador del estudiante. Este atributo es llave foránea que la absorbe de la tabla tb_estudiante.

Tabla 3.29 Descripción de la tabla “tb_brigada_estudiante_matricula”

Nombre: tb_datos_estudiante.		
Descripción: Esta tabla contiene todos los datos referentes a los estudiantes.		
Atributo	Tipo	Descripción
id_atributo	Integer	Representa el identificador de los atributos. Este atributo es llave foránea absorbida de la tabla tb_nomen_atributo.
id_estudiante	Integer	Representa el identificador del estudiante. Este atributo es llave foránea que la absorbe de la tabla tb_estudiante.
datos_estudiante	Varchar	Representa el dato del estudiante.

Tabla 3.30 Descripción de la tabla "tb_datos_estudiante"

Nombre: tb_atributo.		
Descripción: Esta tabla contiene todo lo referido a los atributos.		
Atributo	Tipo	Descripción
id_atributo	Integer	Representa el identificador de los atributos.
nombre_atributo	Varchar	Representa el nombre del atributo.

Tabla 3.31 Descripción de la tabla "tb_atributo"

3.5 Diseño de la BD de Nomencladores

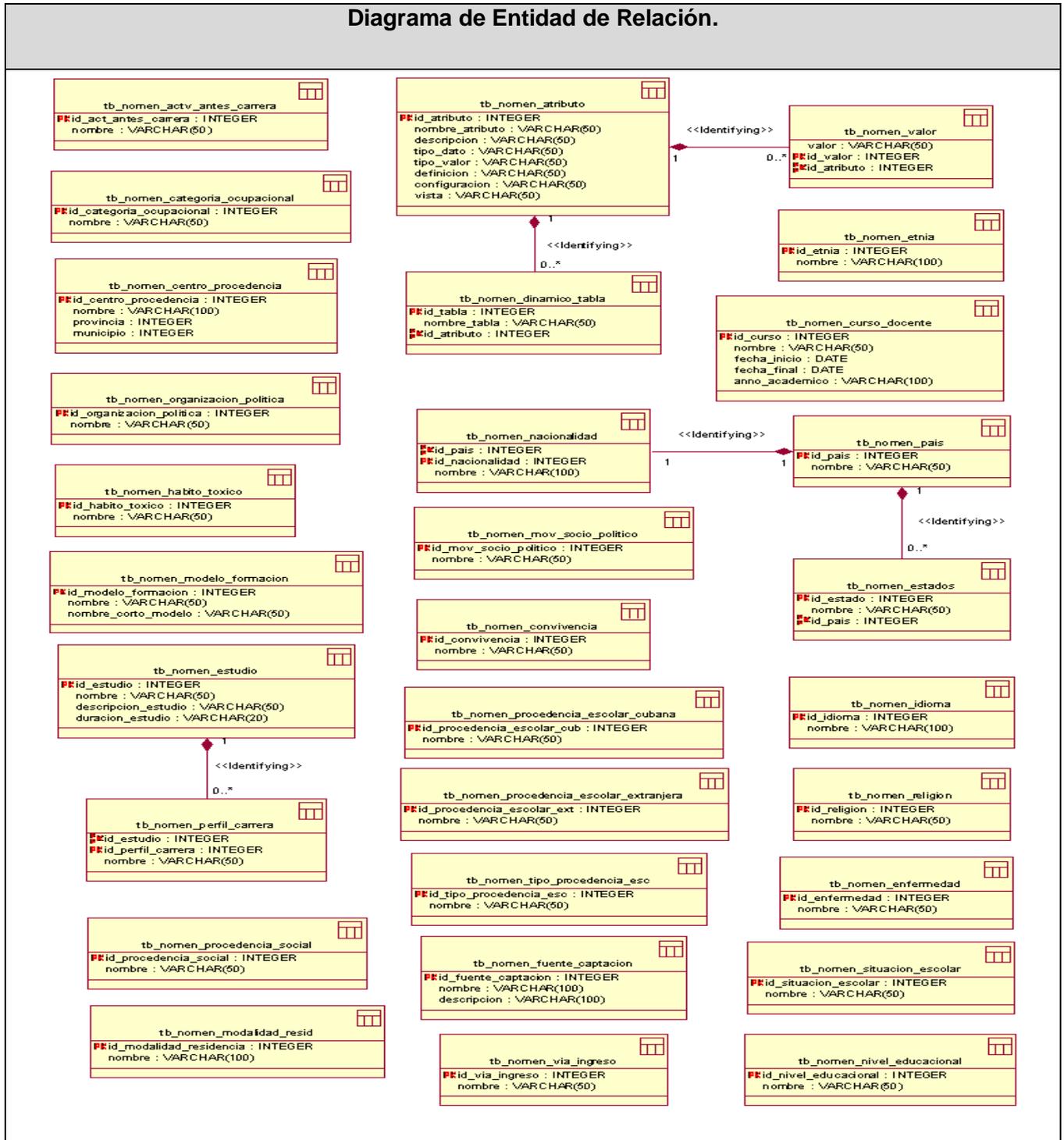


Fig. 3.19 Modelo de Datos “bd_Nomencladores”

3.5.1 Descripción de la BD Nomencladores

Nombre: tb_nomen_actv_antes_carrera.		
Descripción: Esta tabla contiene la información referente al codificador de actividades que puede realizar un estudiante antes de matricular en la carrera.		
Atributo	Tipo	Descripción
id_act_antes_carrera	Integer	Representa el identificador de la actividad antes de la carrera.
nombre	Varchar	Representa el nombre de la actividad de la actividad antes de la carrera.

Tabla 3.32 Descripción de la tabla “tb_nomen_actv_antes_carrera”

Nombre: tb_nomen_atributo.		
Descripción: Esta tabla representa toda la información referente a los atributos.		
Atributo	Tipo	Descripción
id_atributo	Integer	Representa el identificador de los atributos.
nombre_atributo	Varchar	Representa el nombre del atributo.
descripcion	Varchar	Representa la descripción del atributo.
tipo_dato	Varchar	Representa el tipo de dato del atributo.
tipo_valor	Varchar	Representa tipo de valor (dinámico o estático) o si se carga de un servicio web.
definicion	Varchar	Representa la definición del estudiante (nacional o extranjero).
configuracion	Varchar	Representa si el atributo va a ser fijo o configurable.
vista	Varchar	Representa la vista de los datos.

Tabla 3.33 Descripción de la tabla “tb_nomen_atributo”

Nombre: tb_nomen_categoria_ocupacional.		
Descripción: Esta tabla contiene la información referente al codificador de categoría ocupacional de los padres del estudiante.		
Atributo	Tipo	Descripción
id_categoria_ocupacional	Integer	Representa el identificador de la categoría ocupacional.
nombre	Varchar	Representa el nombre de la categoría ocupacional.

Tabla 3.34 Descripción de la tabla “tb_nomen_categoria_ocupacional”

Nombre: tb_nomen_centro_procedencia.		
Descripción: Esta tabla contiene la información referente al codificador de los centros de procedencia de los cuales un estudiante obtuvo la carrera.		
Atributo	Tipo	Descripción
id_centro_procedencia	Integer	Representa el identificador del centro de procedencia.
nombre	Varchar	Representa el nombre del centro de procedencia.
provincia	Integer	Representa la provincia del centro de procedencia.
municipio	Integer	Representa el municipio del centro de procedencia.

Tabla 3.35 Descripción de la tabla “tb_nomen_centro_procedencia”

Nombre: tb_nomen_convivencia.		
Descripción: Esta tabla contiene la información referente al codificador de convivencia.		
Atributo	Tipo	Descripción
id_convivencia	Integer	Representa el identificador de convivencia.
nombre	Varchar	Representa el nombre del centro de procedencia.

Tabla 3.36 Descripción de la tabla “tb_nomen_convivencia”

Nombre: tb_nomen_curso_docente.		
Descripción: Esta tabla contiene la información referente al codificador de los cursos docentes.		
Atributo	Tipo	Descripción
id_curso	Integer	Representa el identificador del curso docente.
nombre	Varchar	Representa el nombre del curso docente.
fecha_inicio	Date	Representa la fecha de inicio del curso.
fecha_final	Date	Representa la fecha de fin del curso.
anno_academico	Varchar	Representa el año académico que cursa.

Tabla 3.37 Descripción de la tabla "tb_nomen_curso_docente"

Nombre: tb_nomen_dinamico_tabla.		
Descripción: Esta tabla contiene la información de las tablas que se van a recoger para cargar una selección dinámica.		
Atributo	Tipo	Descripción
id_tabla	Integer	Representa el identificador de la tabla.
id_atributo	Varchar	Representa el identificador de los atributos. El cual es absorbida de la tb_nomen_atributo.
nombre_tabla	Varchar	Representa el nombre de la tabla.

Tabla 3.38 Descripción de la tabla "tb_nomen_dinamico_tabla"

Nombre: tb_nomen_enfermedad.		
Descripción: Esta tabla contiene la información referente al codificador de enfermedades.		
Atributo	Tipo	Descripción
id_enfermedad	Integer	Representa el identificador de la enfermedad.
nombre	Varchar	Representa el nombre de la enfermedad.

Tabla 3.39 Descripción de la tabla "tb_nomen_enfermedad"

Nombre: tb_nomen_estados.		
Descripción: Esta tabla contiene la información referente al codificador de estados.		
Atributo	Tipo	Descripción
id_estado	Integer	Representa el identificador del estado
id_pais	Integer	Representa el identificador del país. El cual es absorbido de la tb_nomen_pais.
nombre	Varchar	Representa el nombre del estado.

Tabla 3.40 Descripción de la tabla "tb_nomen_estados"

Nombre: tb_nomen_estudio.		
Descripción: Esta tabla contiene la información referente al codificador de las carreras que se estudian en Ciencias Médicas.		
Atributo	Tipo	Descripción
id_estudio	Integer	Representa el identificador de la carrera.
nombre	Varchar	Representa el nombre de los estudios de la carrera.
descripcion_estudio	Varchar	Representa una breve descripción de la carrera.
duracion_estudio	Varchar	Representa el tiempo de duración de los estudios de la carrera.

Tabla 3.41 Descripción de la tabla "tb_nomen_estudio"

Nombre: tb_nomen_etnia.		
Descripción: Esta tabla contiene la información referente al codificador de etnias.		
Atributo	Tipo	Descripción
id_etnia	Integer	Representa el identificador de la etnia.
nombre	Varchar	Representa el nombre de la etnia.

Tabla 3.42 Descripción de la tabla "tb_nomen_etnia"

Nombre: tb_nomen_fuente_captacion.		
Descripción: Esta tabla contiene la información referente al codificador de las fuentes de captación para los estudiantes extranjeros.		
Atributo	Tipo	Descripción
id_fuente_captacion	Integer	Representa el identificador de la fuente de captación.
nombre	Varchar	Representa el nombre de la fuente de captación.
descripcion	Varchar	Representa una breve descripción de la fuente de captación.

Tabla 3.43 Descripción de la tabla "tb_nomen_fuente_captacion"

Nombre: tb_nomen_habito_toxico.		
Descripción: Esta tabla contiene la información referente al codificador de hábitos tóxicos.		
Atributo	Tipo	Descripción
id_habito_toxico	Integer	Representa el identificador del hábito tóxico.
nombre	Varchar	Representa el nombre del hábito tóxico.

Tabla 3.44 Descripción de la tabla "tb_nomen_habito_toxico"

Nombre: tb_nomen_idioma.		
Descripción: Esta tabla contiene la información referente al codificador de idioma.		
Atributo	Tipo	Descripción
id_idioma	Integer	Representa el identificador del idioma.
nombre	Varchar	Representa el nombre del idioma.

Tabla 3.45 Descripción de la tabla "tb_nomen_idioma"

Nombre: tb_nomen_modalidad_resid.		
Descripción: Esta tabla contiene la información referente al codificador de modalidad de residencia.		
Atributo	Tipo	Descripción
id_modalidad_residencia	Integer	Representa el identificador de la modalidad de residencia.
nombre	Varchar	Representa el nombre de la modalidad de residencia.

Tabla 3.46 Descripción de la tabla “tb_nomen_modalidad_resid”

Nombre: tb_nomen_modelo_formacion.		
Descripción: Esta tabla contiene la información referente al codificador del modelo de formación.		
Atributo	Tipo	Descripción
id_modelo_formacion	Integer	Representa el identificador del modelo de formación.
nombre	Varchar	Representa el nombre del modelo de formación.
nombre_corto_modelo	Varchar	Representa el nombre corto del modelo de formación.

Tabla 3.47 Descripción de la tabla “tb_nomen_modelo_formacion”

Nombre: tb_nomen_mov_socio_politico.		
Descripción: Esta tabla contiene la información referente al codificador de movimientos socio-políticos del extranjero.		
Atributo	Tipo	Descripción
id_mov_socio_politico	Integer	Representa el identificador de los movimientos socio-políticos.
nombre	Varchar	Representa el nombre de los movimientos socio-políticos.

Tabla 3.48 Descripción de la tabla “tb_nomen_mov_socio_politico”

Nombre: tb_nomen_nivel_educacional.		
Descripción: Esta tabla contiene la información referente al codificador del nivel educacional del estudiante.		
Atributo	Tipo	Descripción
id_nivel_educacional	Integer	Representa el identificador del nivel educacional.
nombre	Varchar	Representa el nombre del nivel educacional.

Tabla 3.49 Descripción de la tabla “tb_nomen_nivel_educacional”

Nombre: tb_nomen_nacionalidad.		
Descripción: Esta tabla contiene la información referente al codificador de la nacionalidad de los estudiantes.		
Atributo	Tipo	Descripción
id_nacionalidad	Integer	Representa el identificador de la nacionalidad de los estudiantes.
id_pais	Integer	Representa el identificador del país. La cual es absorbida de la tb_nomen_pais.
nombre	Varchar	Representa el nombre de la nacionalidad.

Tabla 3.50 Descripción de la tabla “tb_nomen_nacionalidad”

Nombre: tb_nomen_organizacion_politica.		
Descripción: Esta tabla contiene la información referente al codificador de organizaciones políticas.		
Atributo	Tipo	Descripción
id_organizacion_politica	Integer	Representa el identificador de organizaciones políticas.
nombre	Varchar	Representa el nombre de organizaciones políticas.

Tabla 3.51 Descripción de la tabla “tb_nomen_organizacion_politica”

Nombre: tb_nomen_pais.		
Descripción: Esta tabla contiene la información referente al codificador de países.		
Atributo	Tipo	Descripción
id_pais	Integer	Representa el identificador del país.
nombre	Varchar	Representa el nombre del idioma del país.

Tabla 3.52 Descripción de la tabla “tb_nomen_pais”

Nombre: tb_nomen_perfil_carrera.		
Descripción: Esta tabla contiene la información referente al codificador del perfil de carreras.		
Atributo	Tipo	Descripción
id_perfil_carrera	Integer	Esta tabla contiene la información referente al codificador del perfil de las carreras de tecnologías de la salud.
id_estudio	Integer	Representa el código de la carrera a la cual pertenece el perfil. Este atributo es llave foránea que absorbe de la tabla tb_nomen_estudio.
nombre	Varchar	Representa el nombre del perfil.

Tabla 3.53 Descripción de la tabla “tb_nomen_perfil_carrera”

Nombre: tb_nomen_procedencia_escolar_cubana.		
Descripción: Esta tabla contiene la información referente al codificador de la procedencia escolar de los cubanos.		
Atributo	Tipo	Descripción
id_procedencia_escolar_cub	Integer	Esta tabla contiene la información referente al codificador de la procedencia escolar de los cubanos.
nombre	Varchar	Representa el nombre de la procedencia escolar.

Tabla 3.54 Descripción de la tabla “tb_nomen_procedencia_escolar_cubana”

Nombre: tb_nomen_procedencia_escolar_extranjera.		
Descripción: Esta tabla contiene la información referente al codificador de la procedencia escolar de los extranjeros.		
Atributo	Tipo	Descripción
id_procedencia_escolar_ext	Integer	Esta tabla contiene la información referente al codificador de la procedencia escolar de los extranjeros.
nombre	Varchar	Representa el nombre de la procedencia escolar.

Tabla 3.55 Descripción de la tabla “tb_nomen_procedencia_escolar_extranjera”

Nombre: tb_nomen_procedencia_social.		
Descripción: Esta tabla contiene la información referente al codificador de procedencia social.		
Atributo	Tipo	Descripción
id_procedencia_social	Integer	Representa el identificador de la procedencia social.
nombre	Varchar	Representa el nombre de la procedencia social.

Tabla 3.56 Descripción de la tabla “tb_nomen_procedencia_social”

Nombre: tb_nomen_religion.		
Descripción: Esta tabla contiene la información referente al codificador de la religión.		
Atributo	Tipo	Descripción
id_religion	Integer	Representa el identificador de la religión.
nombre	Varchar	Representa el nombre de la religión.

Tabla 3.57 Descripción de la tabla “tb_nomen_religion”

Nombre: tb_nomen_situacion_escolar.		
Descripción: Esta tabla contiene la información referente al codificador situación escolar del estudiante.		
Atributo	Tipo	Descripción
id_situacion_escolar	Integer	Representa el identificador de la situación escolar.
nombre	Varchar	Representa el nombre de la situación escolar.

Tabla 3.58 Descripción de la tabla “tb_nomen_situacion_escolar”

Nombre: tb_nomen_tipo_procedencia_esc.		
Descripción: Esta tabla contiene la información referente al codificador procedencia escolar de los estudiantes.		
Atributo	Tipo	Descripción
id_tipo_procedencia_esc	Integer	Representa el identificador del tipo de procedencia escolar del estudiante.
nombre	Varchar	Representa el nombre del tipo de procedencia escolar del estudiante.

Tabla 3.59 Descripción de la tabla “tb_nomen_tipo_procedencia_esc”

Nombre: tb_nomen_valor.		
Descripción: Esta tabla contiene los valores de los atributos.		
Atributo	Tipo	Descripción
id_valor	Integer	Representa el identificador del valor del atributo.
id_atributo	Integer	Representa el identificador del atributo.
valor	Varchar	Representa el valor del atributo.

Tabla 3.60 Descripción de la tabla “tb_nomen_valor”

Nombre: tb_nomen_via_ingreso.		
Descripción: Esta tabla contiene la información referente al codificador vías de ingreso a la carrera.		
Atributo	Tipo	Descripción
id_via_ingreso	Integer	Representa el identificador de la vía de ingreso.
nombre	Varchar	Representa el nombre de la vía de ingreso.

Tabla 3.61 Descripción de la tabla “tb_nomen_via_ingreso”

3.6 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 de diseño. A través de la misma se interactúa con los recursos que se ponen a disposición del usuario en el ordenador.

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.

- Lograr una autonomía, la aplicación y el entorno pertenecen al usuario, pero esto no significa que se abandonen las reglas ofreciéndole cierta amplitud de maniobrabilidad.
- Mantener un nivel alto de consistencia a través de toda la aplicación, reutilizando los criterios.
- 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 claridad.

Algunas interfaces del componente se muestran en el [Anexo 2](#).

3.7 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. Evento muy común en las páginas de error de las aplicaciones, que ofrecen información detallada acerca del error. Esta información puede ser aprovechada por el cliente para un uso malintencionado. En muchos casos las páginas de error también pueden dar información de versiones, paths de las aplicaciones e incluso la localización exacta de ficheros de cabecera (p.e, ficheros .inc) que suelen tener información sensible de la aplicación (usuario/contraseña de acceso a bases de datos, parte del código dinámico de las aplicaciones).

En la aplicación el tratamiento de errores está presente del lado del cliente. Se utiliza código JavaScript combinando el uso de expresiones regulares que brinda fortaleza y garantiza 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/o modificación de registros en la base de datos, se realicen de forma correcta. Lo antes mencionado establece mecanismos que visualizan la información para evitar, en la medida de lo posible, la introducción manual de los datos y minimizar los errores que pueda tener la aplicación. Lo cual convierte en válida toda la información.

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 por causas de infracción serán fácilmente configurables. Otros errores pueden ser generados por el Gestor de Base de Datos, estos se capturan antes de mostrarse al usuario y una vez tratados por el sistema, son mostrados al usuario de una manera entendible para él.

3.8 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. Para definir el estilo de codificación a seguir en la aplicación se utilizó la notación estándar establecida para aplicaciones desarrolladas en PHP.

Las etiquetas de apertura y cierre del lenguaje serán de la forma `<?php ?>`, pues son las que están configuradas por defecto en los servidores web como Apache.

Se harán uso de los arreglos predefinidos para el manejo de los valores enviados por el usuario `$_GET`, `$_POST`, `$_SESSION`, aprovechando las ventajas que esto trae.

Para nombrar las variables se escribirán los identificadores con letras minúsculas y en español. Se utiliza como separador para las palabras el carácter “_” tratando de usar nombres sugerentes a la acción de la variable.

Los nombres de las funcionalidades y clases deben ser lo más descriptivo posible y se debe evitar el uso de abreviaturas.

Durante el desarrollo del capítulo se ha logrado modelar todo el sistema. Se desarrollaron las etapas de análisis y diseño. Se obtuvieron los diagramas de clases del análisis, que reflejaban las relaciones existentes entre las clases interfaz, controladoras y entidades. En la etapa de diseño a partir de los artefactos del análisis, quedaron elaborados los diagramas de interacción que permitieron asignar responsabilidades a las clases. Además se obtuvieron los diagramas de clases del diseño donde se utilizaron estereotipos web, debido al tipo de aplicación que se desarrolló.

CAPÍTULO 4: IMPLEMENTACIÓN

En este capítulo se procede a implementar 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 de Análisis y Diseño. 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 Diagrama de despliegue

Un diagrama de despliegue muestra la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. A continuación se muestra una propuesta de despliegue que es como se distribuirá el sistema. La misma puede variar en dependencia de las posibilidades y recursos con que cuenta la institución cliente que en este caso es el Ministerio de Salud Pública (MINSAP).

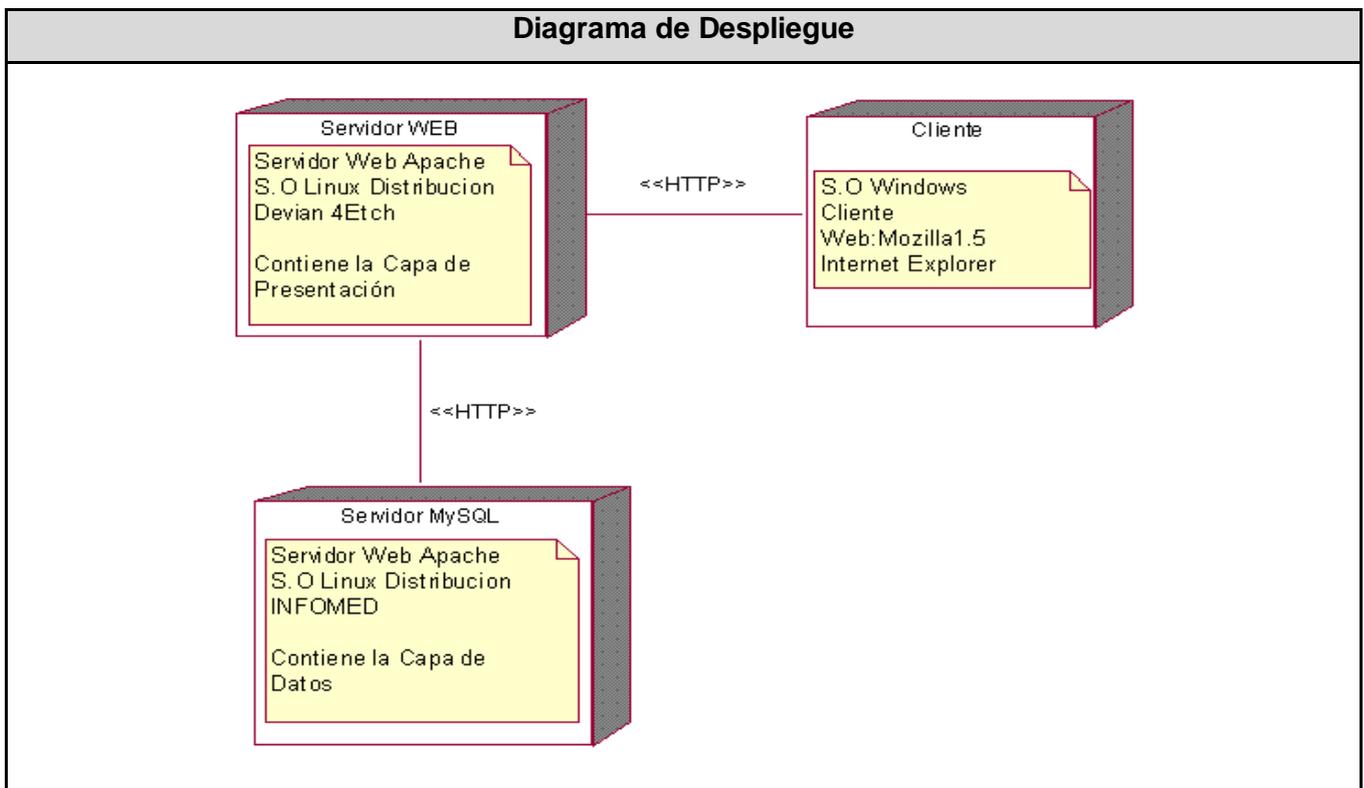


Fig. 4.1 Diagrama de despliegue

4.2 Diagrama de componentes

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación.

Con el objetivo de lograr un mayor entendimiento, se realizó un diagrama de componentes general de la aplicación y posteriormente se dividió en subsistemas de implementación. Estos proporcionan una forma de modelar los artefactos del modelo de implementación y de dividirlos en pequeñas partes que pueden ser integradas y probadas de forma separada, con el objetivo de lograr un mayor entendimiento.

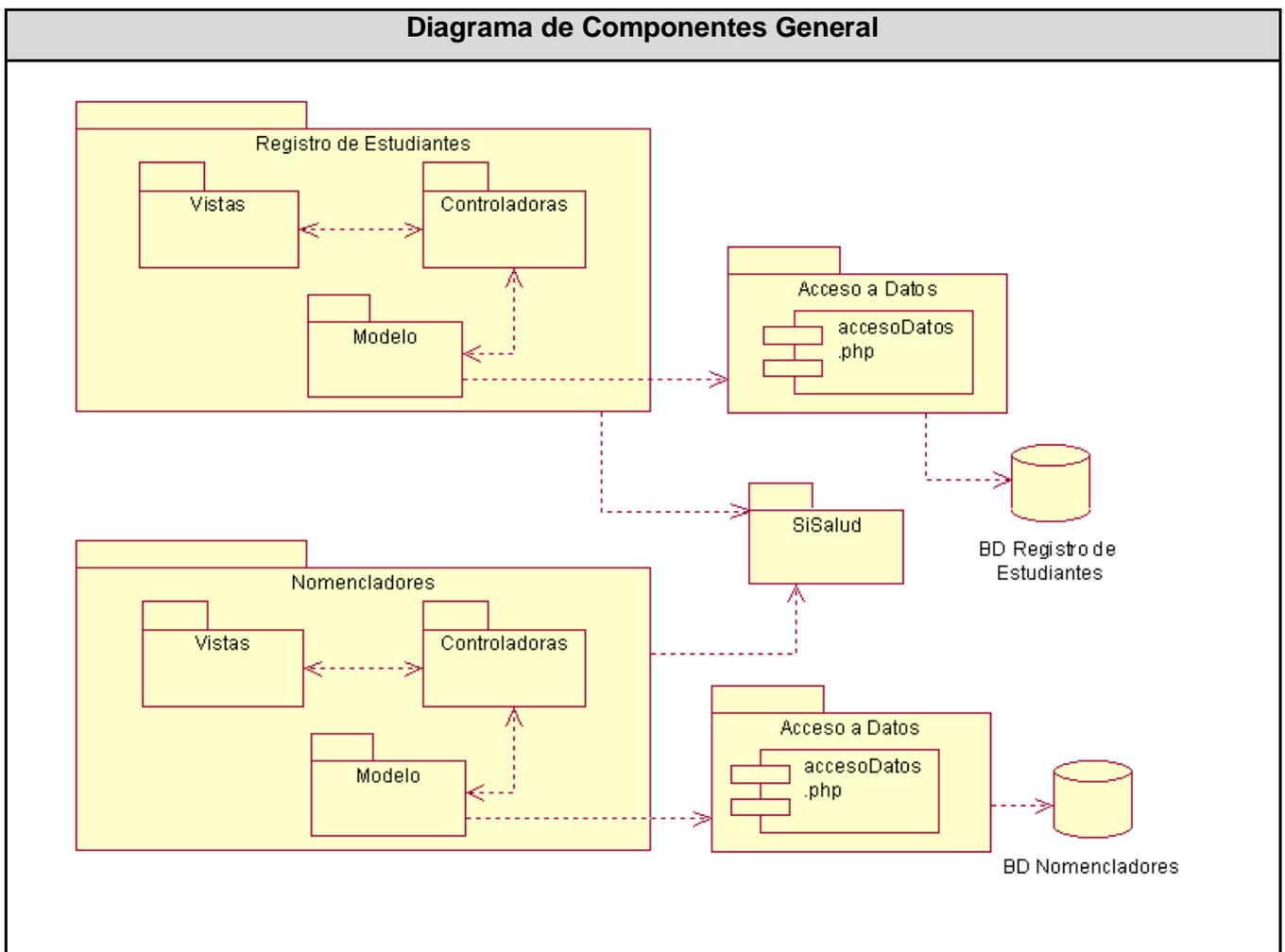


Fig. 4.2 Diagrama de componentes general

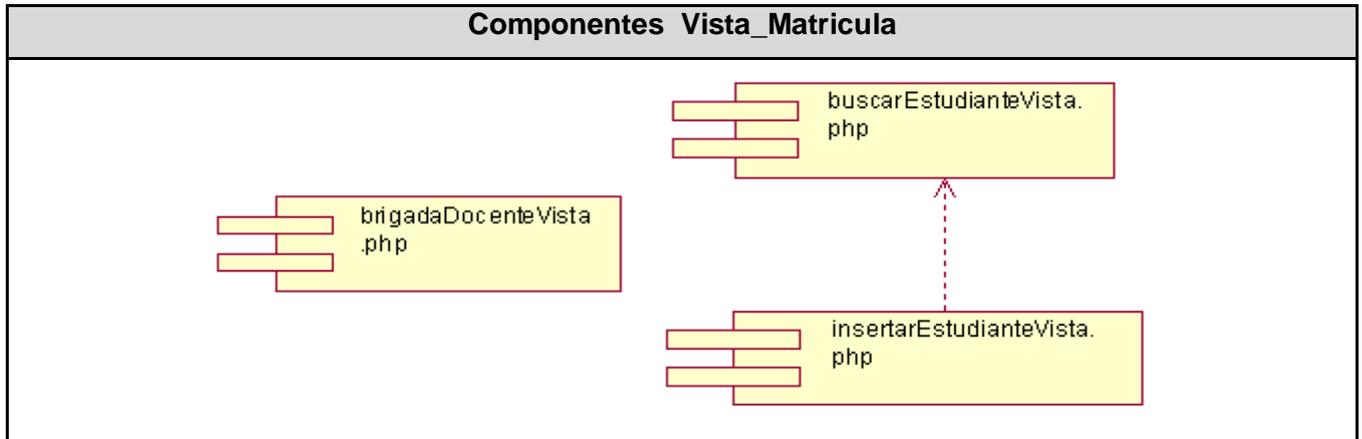


Fig. 4.3 Diagrama de componentes "Vista_Matricula"

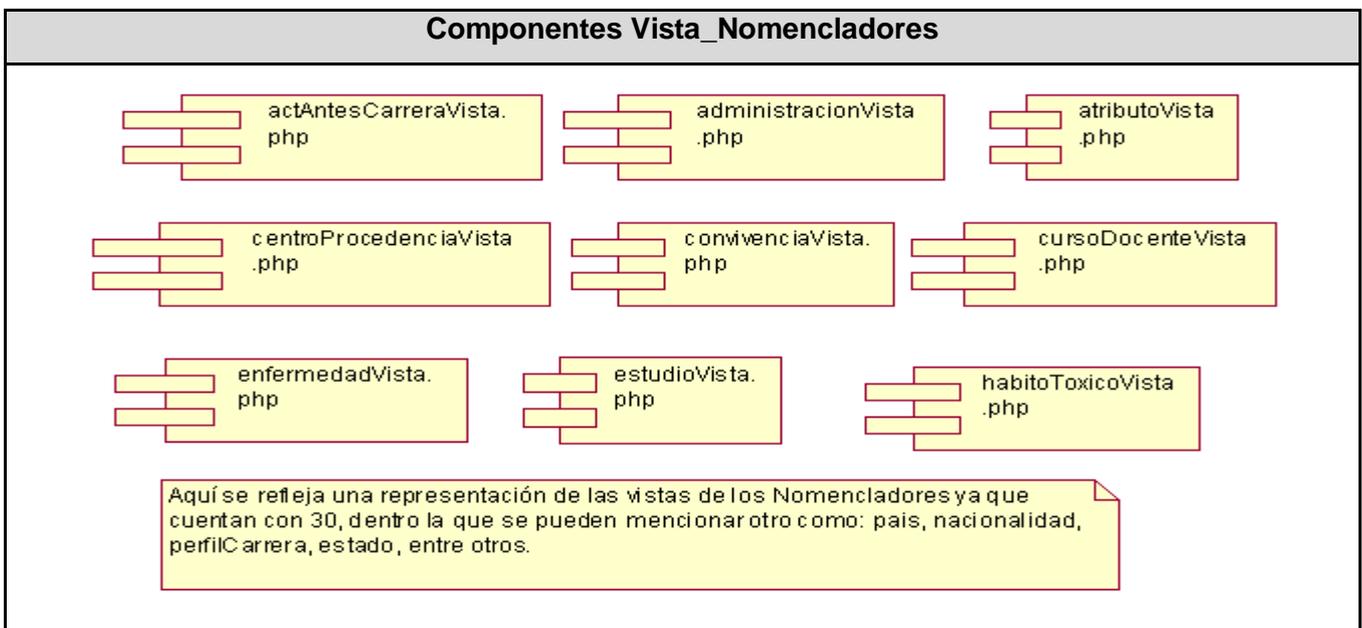


Fig. 4.4 Diagrama de componentes "Vista_Nomencladores"

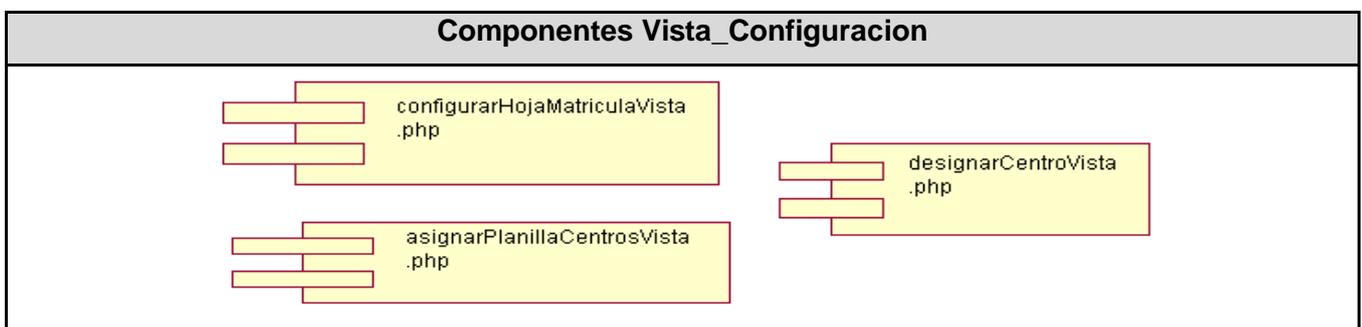


Fig. 4.5 Diagrama de componentes "Vista_Configuracion"

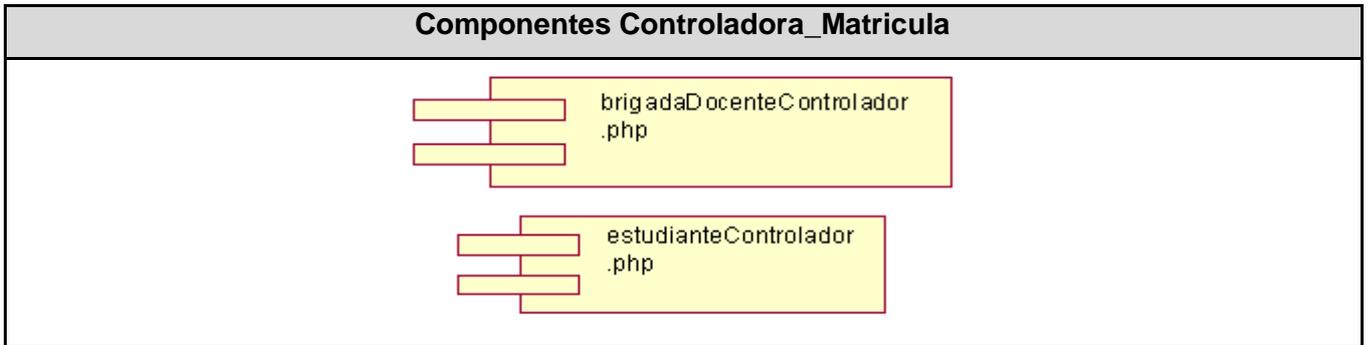


Fig. 4.6 Diagrama de componentes "Controladora_Matricula"

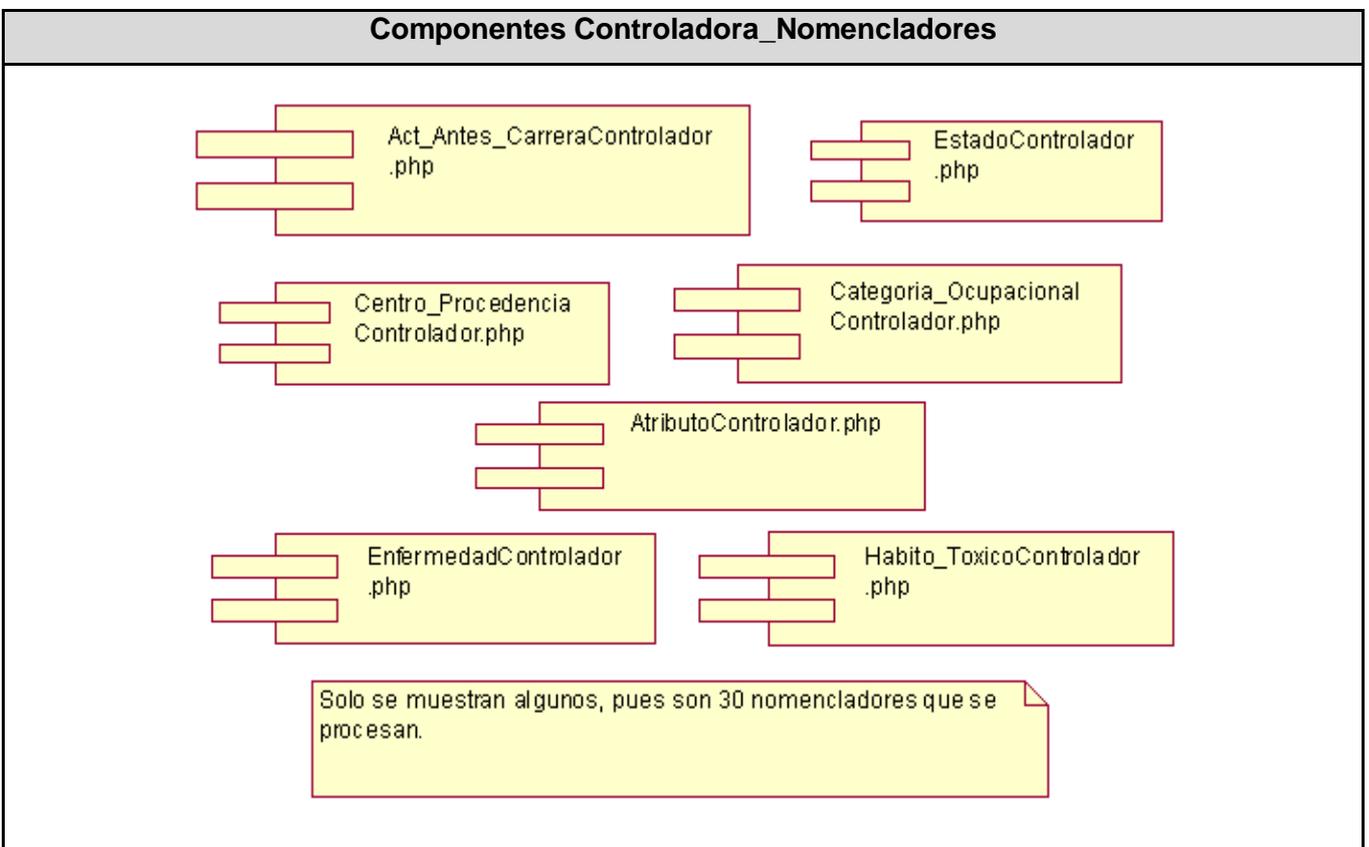


Fig. 4.7 Diagrama de componentes "Controladora_Nomencladores"

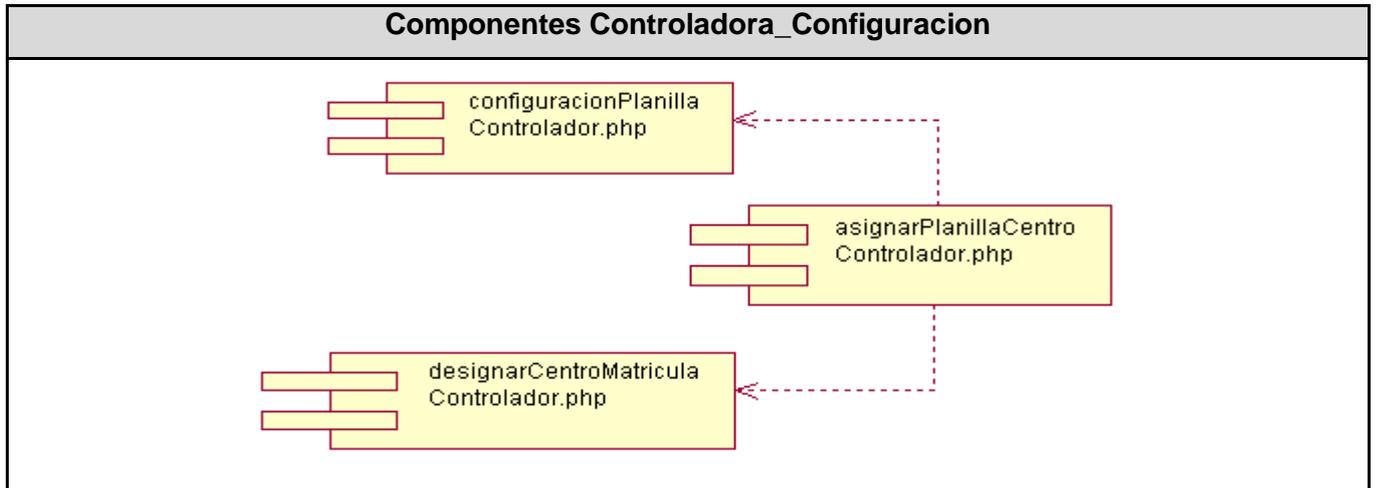


Fig. 4.8 Diagrama de componentes "Controladora_Configuracion"

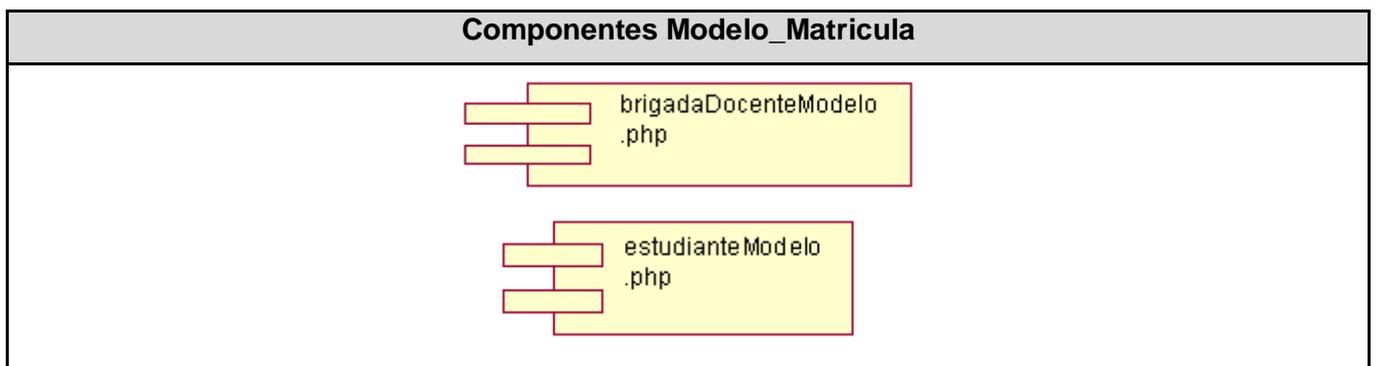


Fig. 4.9 Diagrama de componentes "Modelo_Matricula"

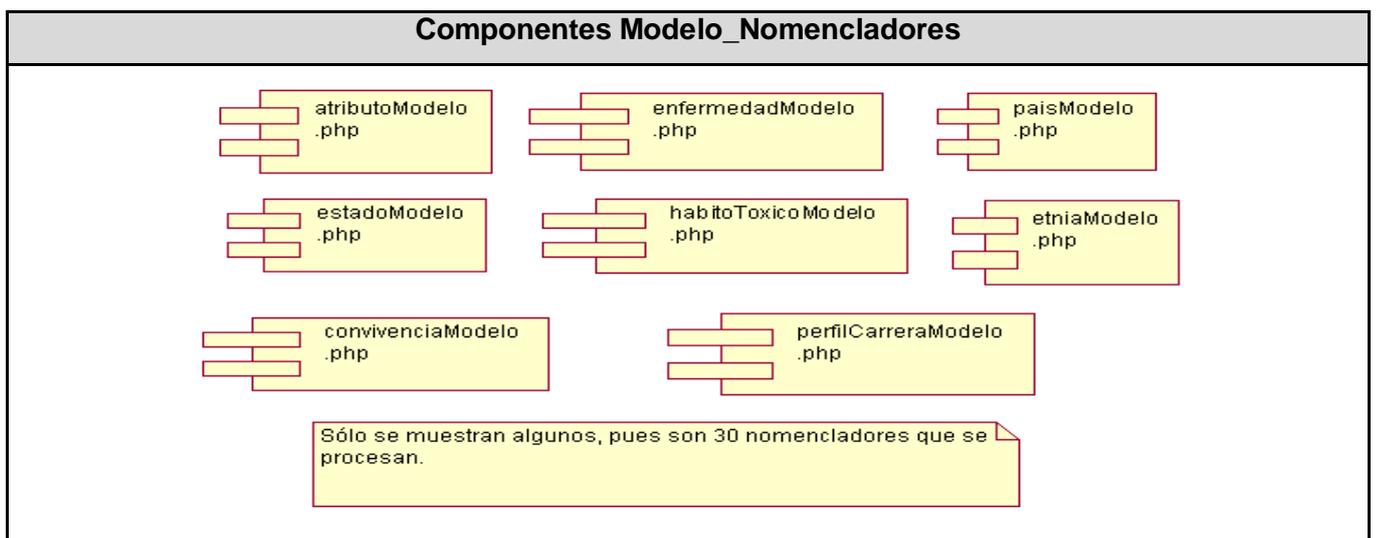


Fig. 4.10 Diagrama de componentes "Modelo_Nomencladores"

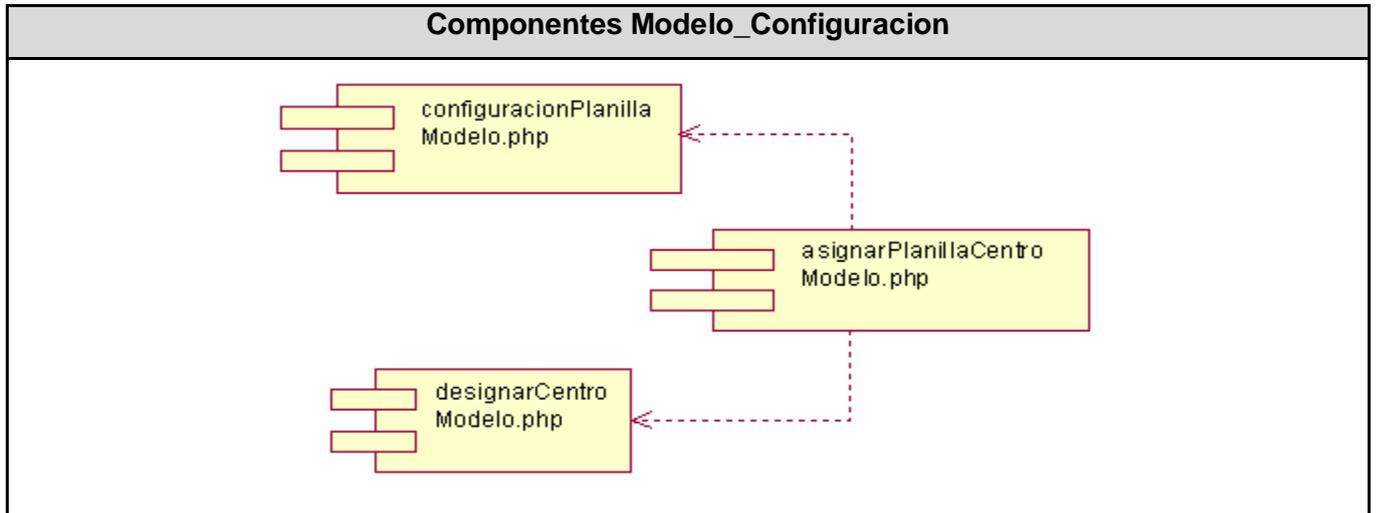


Fig. 4.11 Diagrama de componentes "Modelo_Configuracion"

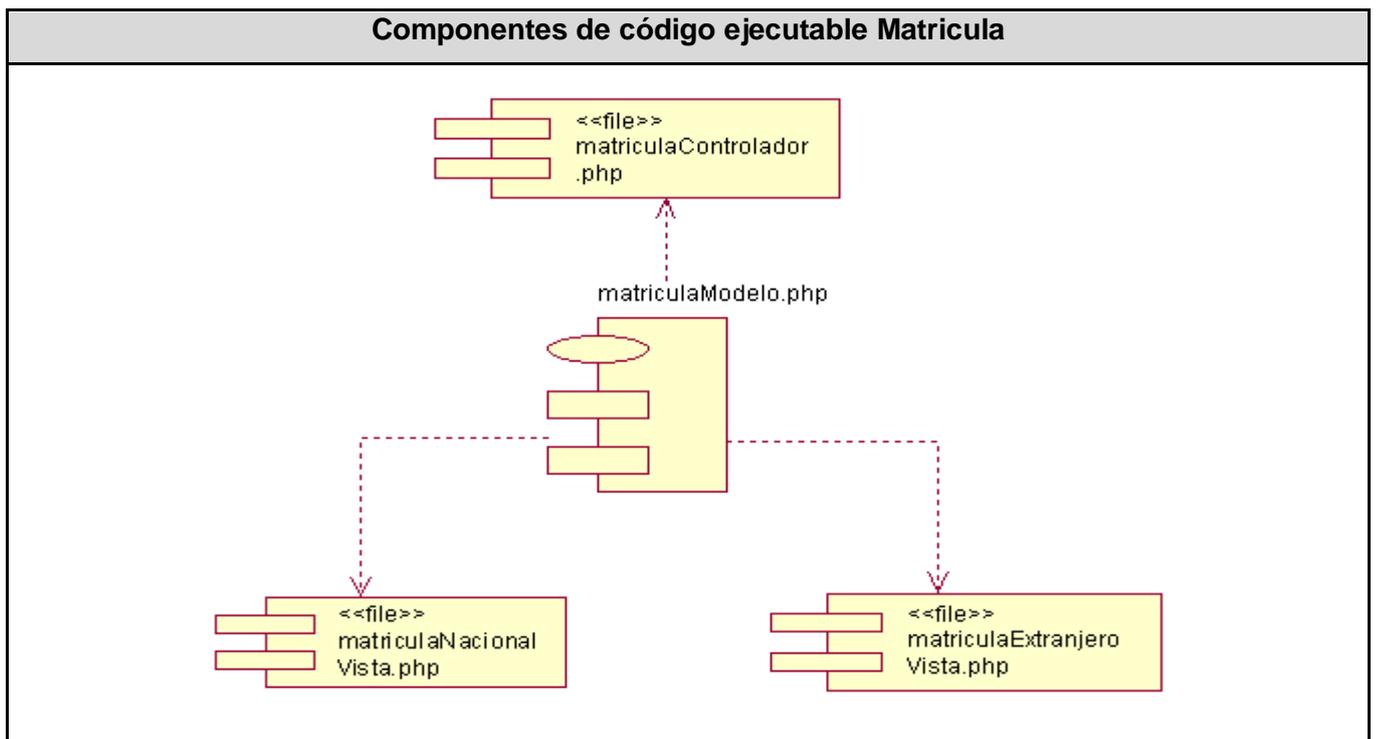


Fig. 4.12 Diagrama de componentes de código ejecutable "Matricula"

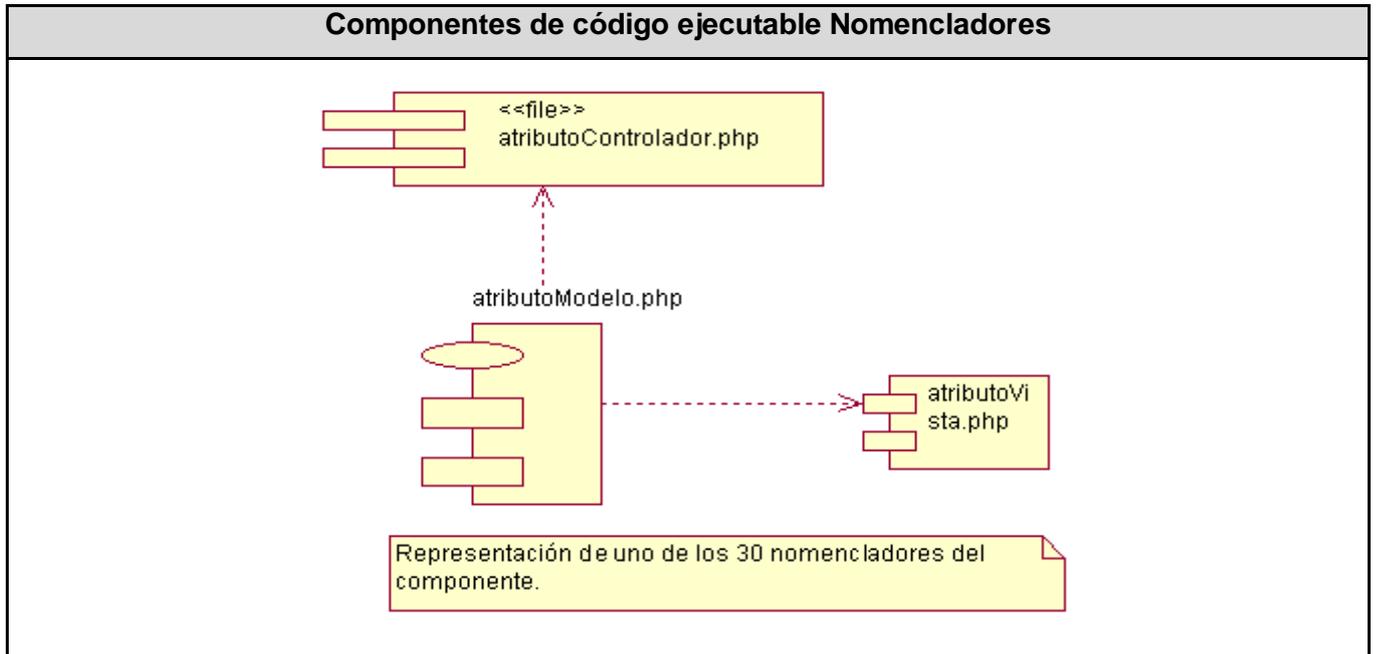


Fig. 4.13 Diagrama de componentes de código ejecutable “Nomencladores”

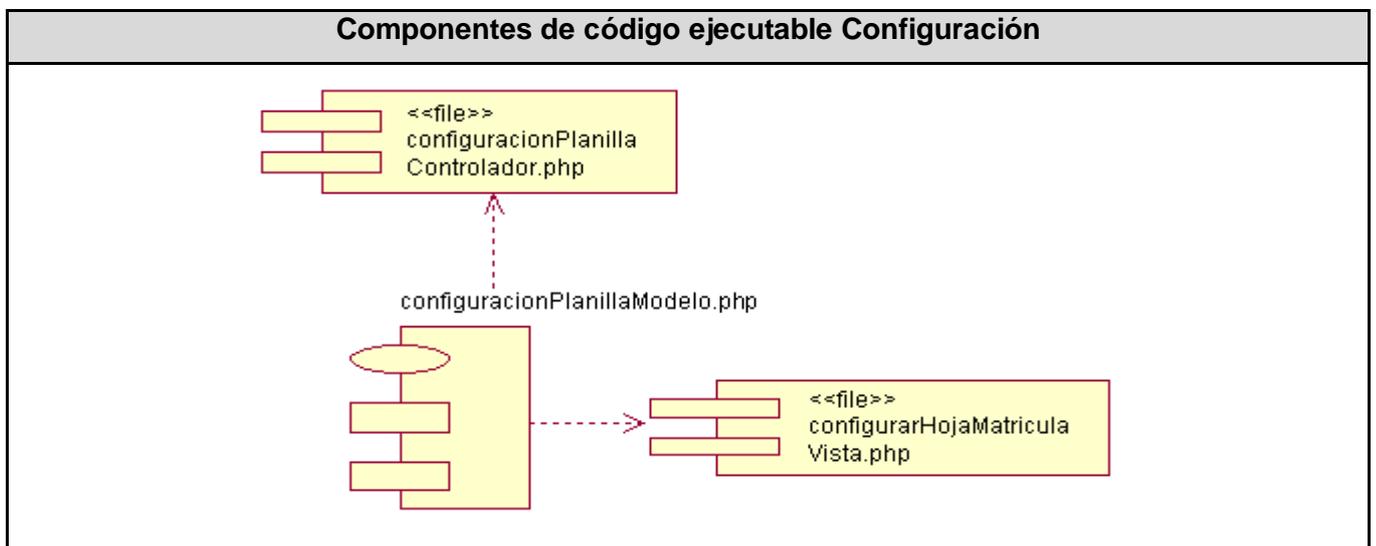


Fig. 4.14 Diagrama de componentes de código ejecutable “Configuración”

4.3 Publicaciones de Servicios Web

La implementación de servicios web es una de las fases más importantes del Componente Registro de Estudiantes. Cuenta con un Componente Nomencladores adicional que brinda la posibilidad a componentes externos, de consumir algunas de sus principales funcionalidades.

Para publicar un Servicio Web se agrupan las funcionalidades que se deseen brindar como servicios en una o varias clases contenedoras. Posteriormente se especifica la descripción para cada tipo de datos, tanto de entrada como de salida de cada función. Luego utilizando el IDE de Desarrollo para PHP, ZendStudio, se especifica la dirección del fichero donde se encuentran la clases contenedoras; a continuación se selecciona la clase que contiene las funciones que se desean publicar y finalmente se describe el servicio en un fichero .wsdl que es la interfaz entre el Servicio Web y los sistemas o clientes que necesitan consumir alguno de los servicios publicados.

4.3.1 Servicios Web

El Componente Nomencladores cuenta con la gestión de 30 de ellos, que brindan Servicios Web como Listar y Buscar. Debido a que su estructura es similar, a continuación se describen sólo algunos de los servicios brindados por el componente. También se especifican los parámetros de entrada y salida, con un ejemplo sobre la forma de implementación de cómo consumirlos.

Nombre del Servicio Web: listarActividad.	
Descripción General: Este servicio es el encargado de devolver el listado de las actividades del nomenclador tb_nomen_actv_antes_carrera el cual es consumido por el Componente Registro de Estudiantes.	
PARÁMETROS DE ENTRADA	
No tiene parámetros de entrada.	
PARÁMETROS DE SALIDA	
id_act_antes_carrera	String
nombre	String

Tabla 4.1 Descripción del Servicio Web “listarActividad”

Ejemplo de Implementación
<pre> <? try { \$cliente = new SoapClient("http://10.31.15.66:5800/ComponenteNomencladores/modelo/wsdl/actividad.wsdl"); //-----listar \$arg = array(); \$r = \$cliente->__call("listarActividad",\$arg); for (\$i = 0; \$i < sizeof(\$r); \$i++) { \$id = \$r[\$i]['idActividad']; \$id = utf8_decode(\$id); \$nomb = \$r[\$i]['nombreActividad']; \$nomb = utf8_decode(\$nomb); \$result[] = array(\$id,\$nomb); } print_r(\$result); } catch (SoapFault \$f) { print_r(\$f); } ?? </pre>

Fig. 4.15 Ejemplo de Implementación para Consumir el Servicio Web “listarActividad”

Nombre del Servicio Web: buscarActividad.	
Descripción General: Este servicio es el encargado de devolver una búsqueda de las actividades del nomenclador tb_nomen_actv_antes_carrera el cual es consumido por el Componente Registro de Estudiantes.	
PARÁMETROS DE ENTRADA	
id_actividad	String
PARÁMETROS DE SALIDA	
id_actividad	String
nombre	String

Tabla 4.2 Descripción del Servicio Web “buscarActividad”

Ejemplo de Implementación

```

<?
try
{
    $cliente = new SoapClient("http://10.31.15.66:5800/ComponenteNomencladores/modelo/wsdl/actividad.wsdl");

    //----- buscar
    $arg = array('id_actividad'=>'3');
    $r= $cliente->__call("buscarActividad",$arg);

    print_r($result);
}
catch (SoapFault $f)
{
    print_r($f);
}
?>

```

Fig. 4.16 Ejemplo de Implementación para Consumir el Servicio Web “buscarActividad”

Nombre del Servicio Web: listarAtributo	
Descripción General: Este servicio es el encargado de devolver el listado de los atributos del nomenclador tb_nomen_atributo, el cual es consumido por el Componente Registro de Estudiantes.	
PARÁMETROS DE ENTRADA	
No tiene parámetros de entrada.	
PARÁMETROS DE SALIDA	
id_atributo	String
nombre_atributo	String
descripcion	String
tipo_dato	String
tipo_valor	String
definicion	String
configuracion	String
vista	String

Tabla 4.3 Descripción del Servicio Web “listarAtributo”

Ejemplo de Implementación
<pre> <> try { \$cliente = new SoapClient("http://10.31.15.66:5800/ComponenteNomencladores/modelo/wsd/atributo.wsd"); //-----listar \$arg = array(); \$r = \$cliente->_call("listarAtributo",\$arg); for (\$i = 0; \$i < sizeof(\$r); \$i++) { \$id = \$r[\$i]['idAtributo']; \$id = utf8_decode(\$id); \$nomb = \$r[\$i]['nombreAtributo']; \$nomb = utf8_decode(\$nomb); \$desc = \$r[\$i]['descripcion']; \$desc = utf8_decode(\$desc); \$dato = \$r[\$i]['tipoDato']; \$dato = utf8_decode(\$dato); \$valor = \$r[\$i]['tipoValor']; \$valor = utf8_decode(\$valor); \$def = \$r[\$i]['definicion']; \$def = utf8_decode(\$def); \$conf = \$r[\$i]['configuracion']; \$conf = utf8_decode(\$conf); \$vist = \$r[\$i]['vista']; \$vist = utf8_decode(\$vist); \$result[] = array(\$id,\$nomb,\$desc,\$dato,\$valor,\$def,\$conf,\$vist); } print_r(\$result); } catch (SoapFault \$f) { print_r(\$f); } ?> </pre>

Fig. 4.17 Ejemplo de Implementación para Consumir el Servicio Web “listarAtributo”

Nombre del Servicio Web: buscarAtributo.	
Descripción General: Este servicio es el encargado de devolver la búsqueda de los atributos del nomenclador tb_nomen_atributo el cual es consumido por el Componente Registro de Estudiantes.	
PARÁMETROS DE ENTRADA	
id_atributo	String
PARÁMETROS DE SALIDA	
id_atributo	String
nombre_atributo	String
descripcion	String
tipo_dato	String
tipo_valor	String
definicion	String
configuracion	String
vista	String

Tabla 4.4 Descripción del Servicio Web “buscarAtributo”

Ejemplo de Implementación
<pre style="font-family: monospace; font-size: 0.9em; color: #000000;"> <? try { \$cliente = new SoapClient("http://10.31.15.66:5800/ComponenteNomencladores/modelo/wsd1/atributo.wsd1"); //----- buscar \$arg = array('id_atributo'=>'38'); \$r= \$cliente->__call("buscarAtributo",\$arg); print_r(\$result); } catch (SoapFault \$f) { print_r(\$f); } ?></pre>

Fig. 4.18 Ejemplo de Implementación para Consumir el Servicio Web “buscarAtributo”

El Componente Registro de Estudiantes brinda el Servicio Web Listar Estudiante. A continuación se describe y se especifican los parámetros de entrada y salida. Se muestra un ejemplo sobre la forma de implementación de cómo consumirlo.

Nombre del Servicio Web: listarEstudiante.	
Descripción General: Este servicio es el encargado de devolver el listado de los estudiantes de la tabla tb_datos_estudiante.	
PARAMETROS DE ENTRADA	
No tiene parámetros de entrada.	
PARÁMETROS DE SALIDA	
id_estudiante	Integer
Id_atributo	Integer
datos_estudiante	String

Tabla 4.5 Descripción del Servicio Web “listarEstudiante”

Ejemplo de Implementación
<pre> <? try { \$cliente = new SoapClient("http://10.31.15.66:5800/ComponenteRegistro/modelo/wsd/estudiante.wsdl"); \$arg = array(); \$r = \$cliente->__call("listarEstudiante",\$arg); for (\$i = 0; \$i < sizeof(\$r); \$i++) { \$idEstudiante = \$r[\$i]['id_estudiante']; \$idEstudiante = utf8_decode(\$idEstudiante); \$idAtributo = \$r[\$i]['id_atributo']; \$idAtributo = utf8_decode(\$idAtributo); \$sDatoEstudiante = \$r[\$i]['datos_estudiante']; \$sDatoEstudiante = utf8_decode(\$sDatoEstudiante); \$result[] = array(\$idEstudiante,\$idAtributo,\$sDatoEstudiante); } print_r(\$result); } catch (SoapFault \$f) { print_r(\$f); } ?> </pre>

Fig. 4.19 Ejemplo de Implementación para Consumir el Servicio Web “listarEstudiante”

4.4 Integración con otros sistemas

Disponer de una infraestructura de hardware, software y comunicaciones, fiable y correctamente dimensionada para soportar las aplicaciones, es fundamental para obtener un óptimo rendimiento. Esto logra una parcial o total integración con otros sistemas. El componente logra adquirir dicha integración a estos sistemas para el uso común de informaciones y funcionalidades necesarias.

4.4.1 SAAA

SAAA (Single Authentication, Authorization and Accounting). 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 Componente Registro de Estudiantes se gestiona a través del SAAA, para garantizar el acceso de los diferentes usuarios que se identificarán con la aplicación una vez en funcionamiento.

4.4.2 RC (Registro de Ciudadano)

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 realiza con el fin de homologar este sistema externo con el Componente Registro de Estudiantes.

4.4.3 RUS (Registro de Unidad de Salud)

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` que pertenece a este componente, se obtienen las unidades de salud que permitirán al Editor Nacional seleccionar la deseada.

4.4.4 RU (Registro de Ubicación)

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 `ListarProvincias` se obtiene un listado de todas las provincias con sus correspondientes municipios, el cual será utilizado para el proceso de ubicación de los usuarios.

En este capítulo se desarrolló el flujo de trabajo de implementación. Se muestran todos los diagramas de componentes, de despliegue y la utilización de servicios web. Se expone además la integración que logra el componente con otros sistemas externos del área de la salud.

CONCLUSIONES

Luego de haber realizado un estudio de las principales tendencias y tecnologías actuales para el desarrollo del Componente Registro de Estudiantes, se concluye lo siguiente:

- Se desarrollaron los flujos de trabajo del Proceso Unificado de Desarrollo “Modelado del Negocio”, “Requerimientos”, “Análisis y Diseño” e “Implementación”.
- Se logró un sistema con flexibilidad de configuración y perdurabilidad, desarrollado con tecnologías de software libre y la arquitectura definida por el Área Temática Sistemas de Apoyo a la Salud (SAS).
- Se desarrollaron los XML Web Services públicos, con sus respectivas descripciones (WSDL), los que podrán ser consumidos por componentes desarrollados por el Área Temática SAS, y otros del Sistema de Información para la Salud que necesiten información de los estudiantes.
- Se realizó un Componente Nomencladores que mejoró la gestión de la información en el Componente Registro de Estudiantes. Además, de brindar Servicios Web públicos a este y a otros componentes y sistemas desarrollados por el Área Temática Sistemas de Apoyo a la Salud.
- El diseño e implementación de un sistema robusto permite controlar la gestión de la información relacionada con la matrícula de estudiantes en el proceso de formación de recursos humanos en el Ministerio de Salud Pública.

RECOMENDACIONES

Después de haber dado cumplimiento a los objetivos y tareas del trabajo se recomienda:

- Integrar el Componente Registro de Estudiantes al nuevo Componente de Seguridad realizado en el Área Temática Sistemas de Apoyo a la Salud.
- Desarrollar una segunda versión del componente, utilizando el framework Symfony por las ventajas que brinda, ya que al inicio de esta versión se realizó un estudio del mismo, pero al contar con poco tiempo para el desarrollo del componente, se dificultó su asimilación.
- Completar los manuales de usuarios de los componentes desarrollados, e incorporarles una ayuda a los mismos, para garantizar al usuario mejor comprensión de sus funcionalidades.

REFERENCIAS BIBLIOGRÁFICAS

1. González, Reinier Alonso. *Tesis Reinier Alonso.pdf*. Ciudad de la Habana : s.n., mayo 2007.
2. Información Aplicada a la Gestión Docente. [En línea] [Citado el: 17 de enero de 2008.]
http://www.bvs.sld.cu/revistas/san/vol5_3_01/san12301.htm.
3. Sistema de Gestión Universitaria. [En línea] [Citado el: 18 de enero de 2008.]
<http://www.cecam.sld.cu/pages/desarrollo/gestionuniv.htm>.
4. Beneficio de las Aplicaciones basadas en Web. [En línea] [Citado el: 18 de enero de 2008.]
http://www.masternewmedia.org/es/aplicaciones_web/temas_de_aplicaciones_web/Beneficios_De_Las_Aplicaciones_Basadas_En%20Web_Y_El_Anuncio_De_Microsoft_De_La_Era_En_Vivo.html.
5. Las Aplicaciones Web. [En línea] [Citado el: 19 de enero de 2008.]
http://www.dimagin.net/es/contenido.php?t_id=6.
6. Principales definiciones de los términos más usados en Internet. [En línea] [Citado el: 20 de enero de 2008.] <http://www.informaticamilenium.com.mx/paginas/espanol/sitioweb>.
7. Internet y la Web. [En línea] [Citado el: 20 de enero de 2008.]
http://www.portalplanetasedna.com.ar/web_internet.htm.
8. Funcionamiento de la Web. [En línea] [Citado el: 21 de enero de 2008.]
http://www.wikilearning.com/monografia/historia_y_funcionamiento_de_internet-funcionamiento_de_la_web/3443-6.htm.
9. Web 2.0. [En línea] [Citado el: 21 de enero de 2008.] <http://www.maestrosdelweb.com/editorial/web2>.
10. Carro, Elena Frau. SOA Arquitectura Orientada a Servicios. [En línea] [Citado el: 22 de enero de 2008.] [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).
11. Guía Breve de Servicio Web. [En línea] [Citado el: 24 de enero de 2008.]
<http://www.w3c.es/Divulgacion/Guiasbreves/ServiciosWeb>.
12. Carlos Reynoso, Nicolás Kicillof. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. Universidad de Buenos Aires : s.n., 2004.
13. Modelo Vista Controlador. [En línea] [Citado el: 24 de enero de 2008.]
http://es.wikipedia.org/wiki/Modelo_Vista_Controlador.
14. Arquitectura por Capa. [En línea] [Citado el: 25 de enero de 2008.]
<http://www.utpl.edu.ec/blog/jlbautista/2008/02/04/10>.
15. XML. [En línea] [Citado el: 26 de enero de 2008.]
http://es.wikipedia.org/wiki/XML#Ventajas_del_XML.
16. SOAP (Simple Object Access Protocol). [En línea] [Citado el: 26 de enero de 2008.]
<http://www.desarrolloweb.com/articulos/1557.php>.

17. WSDL. [En línea] [Citado el: 27 de enero de 2008.] <http://es.wikipedia.org/wiki/WSDL>.
18. Servicio Web. [En línea] [Citado el: 27 de enero de 2008.]
http://es.wikipedia.org/wiki/Servicio_Web#Ventajas_de_los_servicios_Web .
19. UDDI. [En línea] [Citado el: 27 de enero de 2008.] <http://es.wikipedia.org/wiki/UDDI>.
20. Lenguaje del lado del cliente. [En línea] [Citado el: 27 de enero de 2008.]
http://www.adelat.org/media/docum/nuke_publico/lenguajes.
21. Ajax. [En línea] [Citado el: 29 de enero de 2008.] <http://es.wikipedia.org/wiki/AJAX>.
22. HTML. [En línea] [Citado el: 29 de enero de 2008.] <http://es.wikipedia.org/wiki/HTML>.
23. XML. [En línea] [Citado el: 29 de enero de 2008.] <http://es.wikipedia.org/wiki/XML>.
24. XHTML. [En línea] [Citado el: 29 de enero de 2008.] <http://es.wikipedia.org/wiki/XHTML>.
25. PHP. [En línea] [Citado el: 30 de enero de 2008.] <http://es.wikipedia.org/wiki/.php>.
26. Sistema Gestor de Base de Datos. [En línea] [Citado el: 30 de enero de 2008.]
http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php .
27. Conceptos básicos del servidor web. [En línea] [Citado el: 30 de enero de 2008.]
http://www.cibernetia.com/manuales/instalacion_servidor.
28. Una Introducción a APACHE. [En línea] [Citado el: 31 de enero de 2008.]
http://linux.ciberaula.com/articulo/linux_apache_intro.
29. Ivar Jacobson, Grady Booch, James Rumbaugh. *Proceso Unificado de Desarrollo de Software*. Madrid : s.n., 2000.
30. James Rumbaugh, Ivar Jacobson, Grady Booch. *Lenguaje Unificado de Modelado* .
31. The Yahoo! User Interface Library (YUI). . [En línea] [Citado el: 31 de enero de 2008.]
<http://developer.yahoo.com/yui>.

BIBLIOGRAFÍA

- ¿Qué es la Web 2.0?* (s.f.). Recuperado el 21 de Enero de 2008, de <http://www.maestrosdelweb.com/editorial/web2/>
- Annia Arencibia Morales, K. G. (2007). *Centro de Control para el Sistema de Información para la Salud*. Ciudad de la Habana, Cuba.
- Aplicaciones Web*. (s.f.). Recuperado el 19 de Enero de 2008, de http://www.dimagin.net/es/contenido.php?t_id=6
- Arquitectura por capas*. (s.f.). Recuperado el 25 de Enero de 2008, de <HTTP://WWW.UTPL.EDU.EC/BLOG/JLBAUTISTA/>
- Beneficios de las Aplicaciones basada en Web*. (s.f.). Recuperado el 18 de Enero de 2008, de http://www.masternewmedia.org/es/aplicaciones_web/temas_de_aplicaciones_web/Beneficios_De_Las_Aplicaciones_Basadas_En%20Web_Y_El_Anuncio_De_Microsoft_De_La_Era_En_Vivo.htm
- Computación Distribuida*. (s.f.). Recuperado el 23 de Enero de 2008, de http://es.wikipedia.org/wiki/Computaci%C3%B3n_distribuida (computación distribuida)
- Gestión Docente*. (s.f.). Recuperado el 17 de Enero de 2008, de http://www.bvs.sld.cu/revistas/san/vol5_3_01/san12301.htm
- González, R. A. (2007). *Sistema de Gestión de Información en el Proceso de Formación de Recursos Humanos en Salud*. Ciudad de la Habana.
- Historia y Funcionamiento de Internet*. (s.f.). Recuperado el 21 de Enero de 2008, de http://www.wikilearning.com/funcionamiento_de_la_web-wkccp-3443-6.htm
- HTML*. (s.f.). Recuperado el 29 de Enero de 2008, de <http://es.wikipedia.org/wiki/HTML>
- Internet y la Web*. (s.f.). Recuperado el 19 de Enero de 2008, de http://www.portalplanetasedna.com.ar/web_internet.htm
- Ivar Jacobson, J. R. (2000). *El Proceso Unificado de Desarrollo del Software*. Madrid.
- James Rumbaugh, I. J. *El Lenguaje Manual Unificado de Modelado* (versión 1.3 ed.).
- La web del programador*. (s.f.). Obtenido de <http://www.lawebdelprogramador.com/>
- Larman, C. (1999). *UML y Patrones*. México.
- Lenguaje del Servidor*. (s.f.). Recuperado el 26 de Enero de 2008, de http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html
- Maestros de Web*. (s.f.). Obtenido de <http://www.maestrosdelweb.com>
- Metodología AJAX*. (s.f.). Recuperado el 29 de Enero de 2008, de <http://es.wikipedia.org/wiki/AJAX>
- Pere Crespo Molina, J. A. *Tecnologías de la información al servicio de la historia clínica electrónica.pdf*.
- PHP*. (s.f.). Recuperado el 30 de Enero de 2008, de <http://es.wikipedia.org/wiki/.php>

- Pressman, R. S. *Ingeniería del Software. Un enfoque práctico* (Quinta edición ed.).
- Principales definiciones de Internet.* (s.f.). Recuperado el 19 de Enero de 2008, de <http://www.informaticamilenium.com.mx/paginas/espanol/sitioweb.htm>
- SAIDO.* (s.f.). Recuperado el 18 de Enero de 2008, de <http://www.cecam.sld.cu/pages/desarrollo/gestionuniv.htm>
- Servicios Web.* (s.f.). Recuperado el 24 de Enero de 2008, de <http://www.w3c.es/Divulgacion/Guiasbreves/ServiciosWeb>
- Servidor Web.* (s.f.). Recuperado el 1 de Febrero de 2008, de http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php
- Sistema Gestor de Base de Datos.* (s.f.). Recuperado el 30 de Enero de 2008, de <http://es.wikipedia.org/wiki/DBMS>
- Sistema Gestores de Base de Datos.* (s.f.). Recuperado el 30 de Enero de 2008, de http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php
- Tecnología Cliente-Servidor.* (s.f.). Recuperado el 24 de Enero de 2008, de <http://www.inei.gov.pe/biblioineipub/bancopub/inf/lib5038/indice.HTM> (cliente-servidor)
- Una introducción a Apache.* (s.f.). Recuperado el 1 de Febrero de 2008, de http://linux.ciberaula.com/articulo/linux_apache_intro/
- Ventajas de los Servicios Web.* (s.f.). Recuperado el 24 de Enero de 2008, de http://es.wikipedia.org/wiki/Servicio_Web#Ventajas_de_los_servicios_Web
- Ventajas de SOAP.* (s.f.). Recuperado el 25 de Enero de 2008, de <http://www.desarrolloweb.com/articulos/1557.php>
- VideoTutoriales.* (s.f.). Obtenido de <http://www.videotutoriales.com/>
- XHTML.* (s.f.). Recuperado el 29 de Enero de 2008, de <http://es.wikipedia.org/wiki/XHTML>
- XML.* (s.f.). Recuperado el 29 de Enero de 2008, de <http://es.wikipedia.org/wiki/XML>
- YUI.* (s.f.). Recuperado el 3 de Febrero de 2008, de http://es.wikipedia.org/wiki/Yahoo!_UI_Library

ANEXO 1: MODELO DE DOMINIO

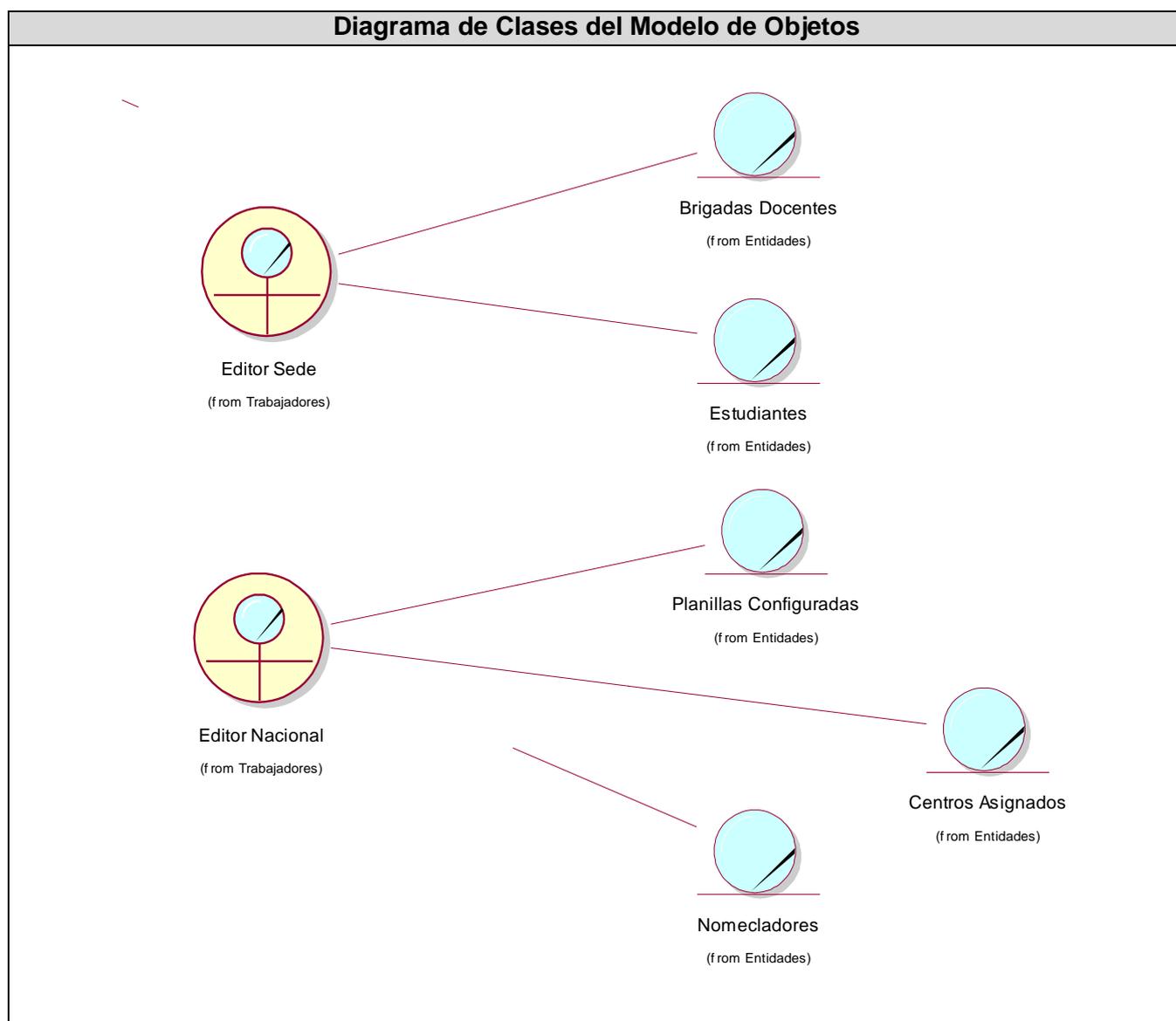


Fig. 1 Diagrama de Clases del Modelo de Objetos

ANEXO 2: INTERFACES DEL COMPONENTE

The screenshot displays the 'Configurar planilla de matrícula' interface within the 'Componente Registro de Estudiantes' system. The interface is structured as follows:

- Header:** Shows the date and time 'Lunes, 7 de Julio de 2008 18:11:24' on the left, and user information 'Usuario: docnace01', 'Derecho: Editor', and 'Nivel: Nacional' on the right.
- Navigation:** A breadcrumb trail reads '>> Inicio >> Planilla de Matrícula >> Configurar'. Below this are tabs for 'Datos de la Carrera Pregrado', 'Datos Generales', 'Datos de la Dirección', 'Datos Personales 1', 'Datos Personales 2', 'Datos Familiares', and 'Referente a la Carrera'.
- Left Menu:** A vertical menu under 'Administración' includes options like 'Designar centros con matrícula', 'Configurar planilla de matrícula', 'Gestionar planilla', 'Asignar planilla a los centros', 'Sugerencias', 'Portal de inicio', 'Ayuda', and 'Cerrar Sesión'.
- Main Content Area:** Titled 'Configurar los datos referentes a la carrera', it features two columns: 'Atributos configurables' (listing 'Clase de estudiante', 'Situación escolar', 'Procedencia escolar', 'Vía de ingreso', 'Concepto de ingreso', 'Situación de defensa', 'Índice académico', and 'Valor del escalafón') and 'Atributos escogidos *'. Between these columns are navigation buttons (>, >>, <, <<). Below the columns is a text input field for 'Nombre Planilla *'.
- Footer:** A bar at the bottom contains the text '(*) Entrada obligatoria' and three buttons: '+ Salvar', 'Vista Previa', and 'Nueva'.
- Page Footer:** The bottom-most bar reads 'Componente Registro de Estudiantes. © Copyright 2008 Área Temática SAS'.

Fig. 2 Interfaz "Configurar planilla de matrícula"

Lunes, 7 de Julio de 2008 18:07:53

Usuario: docuse01
Derecho: Editor
Nivel: Unidad de Salud

Componente Registro de Estudiantes

>> Inicio >> Matrícula >> Estudiantes Extranjeros

Datos de la Carrera Pregrado | Datos Generales | Datos de la Dirección
Datos Personales 1 | Datos Personales 2 | Datos Familiares | Referente a la Carrera

Datos de la carrera de pregrado

Matrícula Estudiantes Extranjeros

Curso docente << Seleccione >> (*)

Escuela de Ciencias Médicas Policlínico Heroes de Girc (*)

Tipo de estudiante Extranjero (*)

Tipo de modelo << Seleccione >> (*)

Carrera << Seleccione >> (*)

Perfiles << Seleccione >> (*)

Tipo de curso << Seleccione >> (*)

Año académico << Seleccione >> (*)

Brigada docente << Seleccione >> (*)

(*) Entrada obligatoria

Componente Registro de Estudiantes. © Copyright 2008 Área Temática SAS

Fig. 3 Interfaz "Nuevo estudiante"

GLOSARIO DE TÉRMINOS

Capa de Datos: Es donde residen los datos. Está formado por uno o más Gestores de Base de Datos que realiza todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Capa de Negocio: Es donde residen los programas que son ejecutados mediante peticiones del usuario y envían la respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que se deben cumplir. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

Capa de Presentación: Es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso (realiza un filtrado previo para comprobar de que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio.

DOM: Siglas de Document Object Model. Es esencialmente un modelo computacional a través de la cual los programas y scripts pueden acceder y modificar dinámicamente el contenido, estructura y estilo de los documentos HTML y XML. Su objetivo es ofrecer un modelo orientado a objetos para el tratamiento y manipulación en tiempo real -o en forma dinámica- a la vez que de manera estática de páginas de internet.

HTTP: Siglas de HyperText Transfer Protocol. Es un protocolo de transferencia de hipertexto, define la sintaxis y la semántica que utilizan los elementos software de la arquitectura Web (clientes, servidores, proxies) para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.

Iframe: Es un elemento HTML que permite insertar o incrustar un documento HTML dentro de un documento HTML principal. Insertar un iframe entre una sección o bloque es semejante a insertar un elemento *object*. Esto permite que se pueda insertar un documento HTML dentro de otro, alineado de acuerdo a sus límites.

INFOMED: Red del Ministerio de Salud Pública (MINSAP) de Cuba.

JSON: Es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML. La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX.

TCP: Siglas de Transmission Control Protocol. Es uno de los protocolos fundamentales en Internet. Muchos programas dentro de una red de datos compuesta por ordenadores pueden usarlo para crear conexiones entre ellos a través de las cuales puede enviarse un flujo de datos. El protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron.

Unidad de Salud: Centro de trabajo del Ministerio de Salud Pública (MINSAP).