

**Universidad de las Ciencias Informáticas
Facultad 7**



**Título: Sistema de Gestión de Información en el
Proceso de Formación de Recursos Humanos en la
Salud. Componente de Gestión Académica**

**Trabajo de Diploma para optar por el título de
Ingeniero Informático**

Autores: Renier Ramos Oliva

Tania Sachyn González Andino

Tutor: Ing.Reinier Alonso González

Asesor: Lic.Julio Antonio Tejera Castillo

Ciudad de La Habana, Junio 2008

“Año 50 de la Revolución”

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los 20 días del mes de Junio del año 2008.

Tania Sachyn González Andino

Renier Ramos Oliva

Firma de la Autora

Firma del Autor

Ing. Reinier Alonso González

Firma del Tutor

DATOS DE CONTACTO

Tutor: 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 de para la Salud de la Facultad 7.

“Nunca consideres el estudio como una obligación, sino como una oportunidad de penetrar en el bello y maravilloso mundo del saber”

Albert Einstein

AGRADECIMIENTOS.

A la Revolución por darnos la oportunidad de estudiar en esta universidad de excelencia y al Comandante Fidel Castro Ruz.

A todos los compañeros de trabajo del área temática SAS por su disposición y ayuda incondicional cuando los hemos necesitado. A la dirección y el claustro de profesores de la Facultad 7 por su imprescindible contribución en nuestra preparación profesional.

A la profesora Yenisel Valido y nuestro tutor Reinier Alonso, por la paciencia y dedicación que tuvieron a lo largo de estos meses de intenso trabajo.

A la profesora Pura por su ayuda incondicional.

A nuestras familias que siempre han estado apoyándonos y alentándonos en todo momento.

A todas aquellas personas que de una forma u otra contribuyeron con nuestra formación personal y profesional.

En particular:

Tania:

A mis compañeros de aula por compartir estos cinco de años de carrera universitaria conmigo.

A mi compañero de tesis, Reinier.

Reinier:

A mis amigos que siempre estuvieron presentes cuando hizo falta, Reinier Tula, Alfredo Guzmán, Alexis Turruella, Danisbel Rojas y Héctor Solís.

A mi hermana y a Jesús por su continua preocupación y apoyo.

A mi compañera de tesis, Tania.

DEDICATORIA.

A mis padres, Norma y Carlos que son el tesoro más grande que tengo, por haberme apoyado en todas mi decisiones.

A mis hermanos Cristyan y Alberto por ser incondicionales y regalarme tanta la alegría.

A todos mis compañeros de aula por haber compartido juntos tantos momentos buenos y haberme ayudado cuando lo necesitaba.

Tania S González Andino

*A las personas que más amo en esta vida: mis padres José y Teresa, mi hermana Yarima, mis abuelas Adolfina y Pastora, por entregarme de forma incondicional su amor y dedicación, por haberme encaminado de forma correcta en la vida.
A la memoria de mi abuelo Agustín.*

Renier Ramos Oliva

RESUMEN

El presente trabajo, surge por la necesidad de controlar toda la información que se genera durante el proceso docente de pregrado en las universidades médicas del país. Tiene como objetivo construir un componente que permita la gestión académica en los centros clínicos docentes integrados al Sistema de Gestión de Información en el Proceso de Formación de los Recursos Humanos en Salud.

La aplicación se desarrolla en un ambiente Web. Para la misma, se definieron usuarios a diferentes niveles de acceso de información, lo cual garantiza, entre otras ventajas, la realización de funciones delimitadas según el rol. Está soportada sobre un servidor LAMP (*Linux, Apache, MySQL, PHP*), utilizando herramientas de software libre. Además, se utilizaron: el lenguaje Javascript, la metodología Ajax y como motor de plantillas Smarty.

La implementación del componente mejora la calidad de las prestaciones al cliente, garantizando efectividad y rapidez en los servicios. Al mismo tiempo, optimiza los procesos de flujo de información Cliente-Servidor; al viajar la menor cantidad de datos por la red, la menor cantidad de veces posibles. También, mejora la gestión interna por el uso de una base de datos eficiente que responde a las necesidades actuales, asegurando la flexibilidad del componente.

PALABRAS CLAVES

Gestión Académica, Informatizar, Centros Clínicos Docentes.

ÍNDICE

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
1.1 ¿Qué es la Gestión Académica?.....	4
1.2 Algunos Sistemas de Gestión Académica utilizados en Cuba.....	4
1.3 Antecedentes.....	5
1.4 Tendencias y tecnologías actuales.	6
1.5 Metodología utilizada para el desarrollo de software.....	12
1.6 Sistema gestor de base de datos.....	14
1.7 Lenguajes de programación utilizados.	16
1.8 ¿Por qué usar PHP y MySql?	20
1.9 Herramientas utilizadas en el desarrollo.	20
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	24
2.1 Conceptos asociados al dominio del problema.	24
2.2 Objeto de estudio.....	25
2.3 Propuesta del sistema.	27
2.4 Modelo de negocio.....	28
2.5 Especificación de los requisitos de software.	28
2.6 Definición de los casos de uso del sistema.	32
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	34
3.1 Análisis	34
3.2 Diseño.	38
3.3 Descripción de las clases.....	46
3.4 Diseño de la base de datos.....	61
CAPÍTULO 4: IMPLEMENTACIÓN	68
4.1 Diagrama de componentes	68
4.2 Diagrama de despliegue.	73
CONCLUSIONES	75
RECOMENDACIONES	76
REFERENCIAS BIBLIOGRÁFICAS.....	77
BIBLIOGRAFÍA.....	79
ANEXOS #2: MODELO DE SISTEMA.	89
ANEXO # 3: MODELO DE DISEÑO.....	104

INTRODUCCIÓN

Los avances en los sectores de la informática y las comunicaciones han sacudido al mundo durante las últimas décadas. La rapidez con que cambian las tecnologías establece una dinámica diferente en las sociedades, imponiendo nuevos retos a alcanzar, para no quedar rezagados en el camino dominante de la informatización.

En Cuba, se trabaja intensamente con el objetivo de utilizar las tecnologías de la información y la comunicación para apoyar la salud pública. Las acciones que se han emprendido en este sentido parten de reconocer la importancia crucial de la revolución científico-técnica que se vive, pero se han caracterizado sobre todo, por priorizar el factor humano y adecuar estos avances a los problemas reales de la nación.

El crecimiento de la matrícula en las universidades médicas cubanas como consecuencia del surgimiento de nuevos programas de formación de profesionales de la salud. Así como, los constantes cambios en los planes de estudio de las diferentes especialidades, ha ocasionado que el control de la información que se maneja en las secretarías docentes resulte cada vez más difícil. Específicamente la gestión académica, es uno de los procesos más importantes que se lleva a cabo en cualquier institución de la Educación Superior. Esta gestión, a diario encuentra trabas porque no se cuenta con un sistema de servicios acorde con las necesidades actuales de estas instituciones.

Hasta hace algunos años, estas tareas se desarrollaban mediante procedimientos manuales y la información era registrada en formato duro. Obviamente, se dedicaban largas horas de trabajo manipular los archivos y acceso a los datos. Como resultado, era deficiente la capacidad de emitir reportes reales en un corto período de tiempo. Durante el curso 1989/1990, el Estado dedica un primer presupuesto a equipar los centros de educación médica con tecnología actualizada. Sin embargo, el propósito de introducir masivamente la informática en los centros docentes de salud, no fue materializada en toda su extensión debido a las dificultades económicas de los primeros años de la década del 90, situación que comenzó a revertirse a partir de 1996.

La introducción de los ordenadores, permitió comenzar a registrar de los datos en soporte digital. Se concentran los esfuerzos para informatizar todo el proceso de gestión académica, con el fin de mejorar la eficiencia, la calidad y la productividad de los servicios.

Con este fin, hasta el año 1999 se utilizó (y en algunos lugares se ha continuado utilizando) el sistema SAIDO. Este fue desarrollado por el Centro de Cibernética Aplicada a la Medicina, para el manejo de los

datos del proceso docente en los departamentos de secretaría a los diferentes niveles. Aunque esta aplicación posee una serie de servicios que disminuyen el trabajo, ha presentado problemas por ser poco flexible a las transformaciones actuales de la enseñanza superior en el sector de la salud.

Como resultado de los problemas antes expuestos, la situación actual para controlar la información docente en las facultades médicas es crítica. Todavía se utilizan aplicaciones implementadas en múltiples lenguajes y sistemas gestores de bases de datos, siempre condicionados por las posibilidades de desarrollo locales que existían en cada uno de estos centros. Provocando que no haya homogeneidad en los datos y por tanto, no se pueden integrar estas soluciones a nivel central. Imposibilitando así, agrupar toda la información en un sólo sistema.

Uno de los mayores inconvenientes, es la lentitud del proceso provocada por la inexistencia de un sistema bien estructurado, altamente configurable y que permita gestionar de forma adecuada un gran volumen de información. Así como, incorporar mecanismos de intercambio y cooperación, la extracción de información y el acceso a datos de forma totalmente integrada.

Por la necesidad del Estado de controlar de forma rápida y efectiva la información referente a la Formación de Recursos Humanos en Salud, el Ministerio de Salud Pública (MINSAP) inició un proyecto en el año 2005, con sede en la Universidad de Ciencias Informáticas. El sistema cuenta con dos módulos liberados en la Red Telemática de INFOMED, implementados en un primer ciclo de desarrollo: Matrícula y Movimientos. Como continuación del mismo, se pretende ampliar el software con reportes a varios niveles, crear un módulo de Pre-Médico y funcionalidades para la gestión académica.

Dada la situación anterior el **problema** consiste en ¿Cómo facilitar el proceso de gestión académica en las secretarías de las instituciones clínico docentes del MINSAP?

Por consiguiente el **objeto de estudio** es el proceso de la gestión docente del curso de pregrado en las instituciones clínico docentes del Ministerio de Salud. El **campo de acción** se centra en el proceso de gestión académica del curso de pregrado en las secretarías de las instituciones clínico docentes del MINSAP.

Para dar solución al problema se propone como **objetivo general**: Desarrollar un componente que permita realizar la gestión académica para el proyecto Sistema de Gestión de Información en el Proceso de Formación de Recursos Humanos en la Salud.

En función de cumplir con los objetivos trazados se definen una serie de **tareas** a realizar:

- Analizar la gestión de planes de estudio en diferentes centros educacionales.

- Analizar el proceso de gestión académica en las secretarías docentes de las instituciones clínicas docentes del MINSAP.
- Asimilar las tecnologías y arquitectura definidas para el sistema al que pertenece el componente de gestión académica.

En virtud de desarrollar con éxito el Componente de Gestión Académica, se han realizado diversas entrevistas con especialistas del Ministerio de la Salud Pública con el objetivo de crear una vinculación estrecha y obtener datos significativos del proceso. Con estos, se definieron todas las funcionalidades a informatizar, los datos de entrada y salida y las restricciones existentes.

El documento está organizado en cuatro capítulos, en los que se fundamenta la solución propuesta. Se hace referencia al estudio de las tecnologías, las herramientas utilizadas para el desarrollo de la aplicación. Así como, el modelado del negocio, el análisis y diseño del sistema y la implementación.

Capítulo 1 “Fundamentación teórica”: se realiza un estudio del estado del arte del tema tratado y de las tecnologías actuales que facilitan el desarrollo del trabajo, describiéndose los principales aspectos de las herramientas y tecnologías a utilizar.

Capítulo 2 “Características del sistema”: se plantea el objeto de estudio, objetivos propuestos y se describe el modelo del negocio, los requisitos y casos de uso del sistema; así como la expansión de los casos de uso.

Capítulo 3 “Análisis y Diseño del sistema”: se realiza los diagramas de clases del análisis y del diseño, así como diagramas de interacción, que facilitan la comprensión del comportamiento entre páginas y objetos. Además se muestra el diagrama de la base de datos del sistema con la descripción de sus tablas y campos.

Capítulo 4 “Implementación”: se desarrolla la misma en términos de componentes, además se describe cómo se organiza el sistema de acuerdo a los nodos especificados en el modelo de despliegue.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se hace referencia a las tendencias actuales de las tecnologías, se da una panorámica de las herramientas de desarrollo definidas, la arquitectura que soporta el componente y las metodologías de desarrollo utilizadas. Se tiene en cuenta la integración del componente a desarrollar con el Sistema de Gestión de Información en el Proceso de Formación de Recursos Humanos en Salud ya que utilizan las mismas tecnologías.

1.1 ¿Qué es la Gestión Académica?

La Gestión Académica tiene su núcleo en las secretarías docentes de los centros de educación universitaria. Estas constituyen la unidad administrativa con características muy específicas, pues su misión no es otra que la de servir de hilo conductor de la gestión administrativa del alumno desde el inicio hasta la finalización de los estudios universitarios gestionando los procesos académicos, coordinando y planificando sus actuaciones.

Su importancia radica en contribuir a definir el rumbo de la universidad, realiza el diseño, implantación y operación de las decisiones académicas, se evalúan los programas académicos y se asegura su calidad.

1.2 Algunos Sistemas de Gestión Académica utilizados en Cuba.

Las tareas de gestión académica en los centros docentes del país se han desarrollado a lo largo de muchos años de forma manual y el registro de la información era archivada en formato duro. Gracias al vertiginoso avance de las tecnologías y su vinculación progresiva con todos los sectores, surgen numerosos intentos para hacer de la gestión docente un proceso más eficiente, dinámico y organizado. A continuación se presentan algunos sistemas utilizados en el país y que tienen la gestión académica como una de sus funcionalidades.

1.2.1 GESTACAD. Sistema para la gestión académica.

Sistema creado por un grupo de jóvenes desarrolladores en un acercamiento a la solución del problema de la gestión de la información docente en las instituciones de la educación superior cubana. La versión en ejecución incluye un software cliente elaborado sobre el lenguaje de programación Borland Delphi, que

tiene acceso a una base de datos construida sobre InterBase Firebird Server. Sus datos se muestran en la INTRANET de la Universidad de Matanzas en formato Web. Utiliza el lenguaje Server Side PHP para la elaboración de las consultas Web a la Base de Datos.

El sistema brinda funcionalidades que responden a la búsqueda de un alumno, proporciona listados de estudiantes por grupo, reportes dinámicos de la información existente, reporte de notas por asignatura y grupo, tabla con los resultados docentes de un grupo en un semestre y de un estudiante en toda su carrera. Además, genera actas de exámenes de las diferentes asignaturas, un registro de características de un grupo de estudiantes y permite dar Baja a un estudiante.

Aunque el sistema resuelve parte de los problemas que existen en las universidades del país no está desarrollado específicamente para un perfil médico. No se adapta a las exigencias presentadas por las facultades médicas, las que tienen características específicas en cada una de sus especialidades.

1.2.2 AKADEMOS

Es un sistema Web desarrollado en la Universidad de las Ciencias Informáticas. Brinda una interfaz común para todos sus usuarios. Además, ofrece la información actualizada en tiempo real de las facultades de la universidad. Akademos realiza la gestión de toda la información referente a la formación de pregrado de un estudiante.

En este caso, a pesar de ser desarrollado con tecnologías web, no sería factible utilizar este sistema. Debido a que la información que se necesita registrar tiene particularidades (distintos modelos de formación) que no se adaptan a las previstas por Akademos. Otra cuestión fundamental, es que ya posee una arquitectura definida que no coincide con la precisada para el Sistema de Gestión de Información en el Proceso de Formación de Recursos Humanos en la Salud.

1.3 Antecedentes

SAIDO

En el sector docente del Ministerio de Salud Pública, hasta el año 1999 se utilizó, y en algunos centros se continúa utilizando, el sistema llamado SAIDO, que contiene la gestión académica entre una de sus funcionalidades. Su base de datos contiene y procesa información sobre los estudiantes de ciencias médicas, e incluye módulos de Matrícula, Actualización y Notas. Además, procesa reportes de Salidas para información del Instituto y el MINSAP y permite obtener por selección listados tablas y totales sobre los datos almacenados. Este software fue creado por el Centro de Cibernética Aplicada a la Medicina, sin

embargo, dada la inserción de nuevos programas de formación de profesionales, acompañados de cambios en los modelos de enseñanza en el Sistema de Salud Cubano, es poco flexible a las necesidades actuales. Cada institución ha implementado una solución al problema con ayuda de su personal, lo que ha conllevado a la no estandarización del sistema.

Todas estas soluciones han sido desarrolladas sobre la filosofía de entorno de escritorio, implementadas mayormente en Visual Basic, Delphi y usando como gestor de base de datos Microsoft Access, aunque algunas han incorporado SQL Server. Las aplicaciones de este tipo limitan las funcionalidades y flexibilidad ante los cambios, producto a que las futuras actualizaciones y mejoras que se le pudieran realizar al sistema tendrían la necesidad de actualizarse en cada una de las secretarías docentes del país. Además limita el acceso a la información de forma real y actualizada, a funcionarios de niveles superiores de dirección.

1.4 Tendencias y tecnologías actuales.

Es imprescindible para los diseñadores de software hoy en día, estar al tanto de todo lo que ocurre a nivel mundial referente a las soluciones informáticas acordes con el desarrollo actual. La calidad de los servicios se ve limitada a la utilización de tecnologías de punta debido a la globalización del mundo en que se vive. Por esta razón, se hace necesario conocer las ventajas y posibilidades que brindan las tecnologías a utilizar en la realización del producto.

En este momento la tendencia hacia la Web es marcada, no solo por los costos reducidos para hacerlo, sino por el alto nivel de flexibilidad y agilidad que permite en la implementación de la estrategia trazada. El acceso universal a una página con estos servicios es independiente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica y capacidades de los usuarios. Por esta razón, en vez de crear clientes para Windows, Mac OS X, GNU/Linux, y otros sistemas operativos, la aplicación es escrita una vez y mostrada a todos los clientes a través del acceso a un servidor. (1)

Esto trae como ventaja que las actualizaciones del sistema sólo se harían en el servidor siendo transparentes para los clientes, evitando posibles errores de actualización provocados por usuarios inexpertos.

En las siguientes secciones se describen los principales conceptos, tecnologías y herramientas a utilizar en la construcción del componente.

1.4.1 Internet y su funcionamiento.

Físicamente, Internet (*Interconnected Networks*) es una red compuesta por miles de redes independientes pertenecientes a instituciones públicas, centros de investigación, empresas privadas y otras organizaciones. Para que se pueda realizar la comunicación efectiva entre todas las máquinas de diversos tipos conectadas entre sí, debe existir un camino físico que las interconecte (líneas telefónicas, conmutadas, redes digitales, enlaces satelitales, microondas, fibra óptica, cable coaxial, etc.) (2)

Para asegurar el intercambio de información entre estas computadoras deben compartir normas que son los denominadas protocolos Internet. En ocasiones se la denomina *conjunto de protocolos TCP/IP*, en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP), que fueron los dos primeros en definirse, y que son los más utilizados de la familia.

Existen diversos protocolos en función del tipo de comunicación que se establece entre las máquinas: FTP (para la transferencia de ficheros), POP y SMTP (para el envío y recepción de correo electrónico), TELNET (para la conexión con terminales remotos), GOPHER, WAIS y HTTP (para el acceso a servidores de información). El protocolo HTTP (HyperText Transfer Protocol) es la base de la World Wide Web o telaraña mundial, abreviada WWW y más conocida como Web (3).

1.4.2 La Web.

La Web es un sistema que facilita el acceso a los distintos recursos que ofrece Internet a través de una interfaz común basada en el hipertexto. Esta se ha hecho enormemente popular con la aparición de programas dotados de interfaces gráficas amigables que permiten a cualquier persona acceder a los documentos de la red conocidos como examinadores (browsers), visualizadores o, más comúnmente, navegadores. Los más conocidos son el Explorer de Microsoft, Mozilla Firefox y el Netscape. (4)

El elemento básico de la Web es la página web, un documento que contiene enlaces de hipertexto a otros documentos relacionados que pueden estar localizados en la propia máquina o en máquinas remotas. De esta forma, siguiendo los enlaces se puede navegar por la red visitando páginas ubicadas en distintos lugares como si formaran una unidad.

1.4.3 Elementos de Arquitectura.

Para poder comprender mejor los conceptos que se tratarán a continuación, primero se hace necesario responder la siguiente pregunta ¿Qué es una arquitectura?

La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema, programa o aplicación y aporta una visión abstracta de alto nivel, posponiendo el detalle de cada uno de los módulos definidos a pasos posteriores del diseño. Tiene la responsabilidad de definir los módulos principales, las responsabilidades de los mismos, la interacción que existirá entre dichos módulos: como el control y flujo de datos, la secuenciación de la información, los protocolos de interacción y comunicación y la ubicación en el hardware (5).

Una definición un poco más aclaratoria desde el punto de vista de ingeniería de software es: "La arquitectura de software de un programa o sistema de cómputo es la estructura o estructuras del sistema que comprenden elementos de software, las propiedades visibles externamente de esos elementos y las relaciones entre ellos" (6).

1.4.4 Definiciones de estilos de arquitectura.

Existen diferentes estilos arquitectónicos como por ejemplo arquitectura en tres capas, cliente-servidor, u orientada a servicios. Estos estilos son una generalización y abstracción de los patrones de diseño, caracterizan una familia de sistemas que están relacionados por compartir propiedades estructurales y funcionales. También puede definirse como "La descripción de los tipos de componentes y de los patrones de interacción entre ellos" (7).

1.4.5 Arquitectura en Capas.

La arquitectura en capas es una de las más utilizadas hoy en día. En general constituyen la organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior" (8).

La ventaja fundamental de una arquitectura n-capas es que el estilo soporta un diseño basado en niveles de abstracción crecientes, lo cual a su vez permite a los implementadores la partición de un problema complejo en una secuencia de pasos incrementales. En segundo lugar, el estilo admite muy naturalmente optimizaciones y refinamientos. En tercer lugar, proporciona amplia reutilización. Al igual que los tipos de datos abstractos, se pueden utilizar diferentes implementaciones o versiones de una misma capa en la medida que soporten las mismas interfaces de cara a las capas adyacentes. Esto conduce a la posibilidad

de definir interfaces de capa estándar, a partir de las cuales se pueden construir extensiones o prestaciones específicas. (9)

1.4.6 Arquitecturas Cliente Servidor.

Esta arquitectura consiste básicamente en que un programa -el cliente- realiza peticiones a otro programa -el servidor- que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma. (10)

Principales ventajas de la Arquitectura Cliente Servidor:

- Centralización del control: Los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos.
- Escalabilidad: Se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- Fácil mantenimiento: Al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.
- Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de Cliente/Servidor que aseguran la seguridad en las transacciones, la amigabilidad del interfaz, y la facilidad de empleo.

1.4.7 Servidor Web.

Un servidor web es un programa que implementa el *protocolo HTTP* (hypertext transfer protocol). Este protocolo está diseñado para transferir los hipertextos, páginas web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. (11)

Es un programa que ejecuta de forma continua en un ordenador (también se utiliza el término para referirse al ordenador que lo ejecuta), manteniéndose a la espera de peticiones por parte de un cliente (un navegador de internet) y que contesta a estas peticiones de forma adecuada, sirviendo una página web que será mostrada en el navegador o mostrando el mensaje correspondiente si se detectó algún error.

Algunos de los servidores web más importantes y conocidos son:

- Apache.
- Internet Information Services (IIS).
- Cherokee.

El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Tiene amplia aceptación en la red desde 1996 por lo que actualmente es uno de los servidores web más usados por ser gratuito, poseer gran fiabilidad y extensibilidad. Este servidor es el utilizado en su versión 2.0.55 en el sistema al que pertenece el componente de gestión académica que se pretende desarrollar.

Dentro de sus puntos fuertes se encuentran:

- Tiene interfaz con todos los sistemas de autenticación.
- Facilita la integración como "plug-ins" de los lenguajes de programación de páginas Web dinámicas más comunes.
- Tiene integración en estándar del protocolo de seguridad SSL.
- Provee interfaz a todas las bases de datos.
- Posee Virtual Host.

Las principales características de Apache son:

- Funcionalidad en múltiples plataformas.
- Elaborado índice de directorios.
- Sencilla administración basada en la configuración de un único archivo.
- Soporte para CGI (Common Gateway Interface) y FastCGI

1.4.8 Estilos de arquitecturas y patrones.

Un patrón es un modelo a seguir para realizar algo, son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Ayudan a construir la experiencia colectiva de Ingeniería de Software ya que constituyen son una abstracción de "problema – solución" Se ocupan de problemas recurrentes, identifican y especifican abstracciones de niveles más altos que componentes o clases individuales y proporcionan vocabulario y entendimiento común.

Existen diversas clasificaciones de patrones como resultado de los múltiples temas a los que se quiere dar solución en la práctica de la arquitectura de software. Entre ellos radican los patrones de análisis, arquitectura, diseño y lenguaje de programación (*idioms*).

Un patrón de arquitectura de software describe un problema particular y recurrente del diseño, que surge en un contexto específico, y presenta un esquema genérico y probado de su solución. (13) El patrón utilizado para desarrollar el sistema al cual pertenece el módulo de gestión académica es el Modelo Vista Controlador. Este logra separar el modelo de negocios de la presentación usada de esta forma la información podrá ser vista en múltiples formatos. Esta separación permite diversas vistas para una misma lógica de negocios.

Elementos del patrón:

- **Modelo:** Contendrá una lista de usuarios autorizados junto con sus contraseñas, la base de datos sobre la que efectuar las consultas y los módulos (*clases o librerías de funciones*) que se encarguen de realizar las operaciones (*comprobar la información del usuario, construir la consulta, efectuar la consulta*).
- **Vista:** Muestra la información del modelo al usuario y al necesitar poco o nada de código, puede ser desarrollada por un equipo de diseñadores independiente del de programadores.
- **Controlador:** Gestiona las entradas del usuario y serviría de "pegamento", recogería las órdenes y datos que el usuario enviase desde la vista, las traduciría en operaciones del modelo de negocio y, en base a los resultados, mostrará los resultados en uno u otro formulario de la vista.

1.4.9 Modelo Cliente Servidor de tres Capas (Three Tier).

La arquitectura Cliente-Servidor es una forma de dividir y especializar programas y funciones, a fin de que la tarea que realiza cada uno, lo haga de la forma más eficiente posible, garantizando así un fácil

mantenimiento y actualización del sistema. Además provee de mayor seguridad y posibilita la reutilización del código(14).

Cada uno de los componentes de la aplicación en una arquitectura de tres capas se separa en una sola entidad. Esto te permite implementar componentes de una manera más flexible, es decir, la aplicación tiene que estar preparada para los posibles cambios que el cliente pueda pedir sin tener que reescribir totalmente la aplicación.

1.5 Metodología utilizada para el desarrollo de software.

La calidad en el desarrollo y mantenimiento del software se ha convertido hoy en día en uno de los principales objetivos estratégicos de las organizaciones, debido a que cada vez más, los procesos principales dependen de los sistemas informáticos para su buen funcionamiento. En los últimos años se han publicado diversos estudios y estándares en los que se exponen los principios que se deben seguir para la mejora de los procesos de software.

Una metodología para el desarrollo de un proceso de software es un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de Sistemas Informáticos. La metodología que va a guiar el proceso de desarrollo del sistema es un paso tan importante. En la actualidad existen diversas metodologías de desarrollo con particularidades específicas, entre las más conocidas están: RUP, XP y MSF. Por las particularidades de la solución que se desarrolla el sistema utiliza RUP.

1.5.1 Lenguaje Unificado de Modelado (UML).

En la totalidad de los casos, las metodologías bases son el resultado de experiencias sumadas, anotaciones, y conceptos. UML no es la excepción, por lo que se ha convertido en un lenguaje de modelado orientado a objetos estándar para definir, organizar y visualizar los elementos que configuran la arquitectura de una aplicación.

Es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. Proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto elementos conceptuales, tales como procesos del negocio y funciones del sistema, como elementos concretos, como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes software reutilizables.

Entre sus principales características se encuentran las de constituir un lenguaje unificado para la modelación de sistemas, contiene corrección de errores viables en todas las etapas, es aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor y los modelos permiten la comunicación con el cliente en todas las etapas.

1.5.2 El Proceso Unificado de Modelado (RUP).

El objetivo final de cualquier aplicación, es un software robusto, flexible y escalable, por lo que es necesario tanto un lenguaje como un proceso para poder obtenerlo.

El *Proceso Unificado de Rational* (RUP), es un proceso de ingeniería de software planteado por Kruchten (1996) cuyo objetivo es producir software de alta calidad, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecido.

RUP toma en cuenta las mejores prácticas en el modelo de desarrollo de software en particular las siguientes:

- Desarrollo de software en forma iterativa (repite una acción).
- Manejo de requerimientos.
- Utiliza arquitectura basada en componentes.
- Modela el software visualmente (modela con UML).
- Verifica la calidad del software.
- Controla los cambios.

Este proceso está basado en componentes y utiliza el UML para diseñar todos los esquemas de un sistema de software. Los aspectos más importantes que definen al proceso unificado lo constituyen tres elementos claves: dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental.

El Proceso Racional Unificado, es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas.

1.5.3 Herramientas CASE

¿Por qué se debería usar herramientas CASE de modelado con UML?

A medida que los sistemas que hoy se construyen se tornan más y más complejos, las herramientas de modelado con UML ofrecen muchos beneficios para todos los involucrados en un proyecto, por ejemplo,

administrador del proyecto, analistas, arquitectos, desarrolladores y otros. Las herramientas CASE de modelado con UML permite aplicar la metodología de análisis y diseño orientados a objetos y abstraerse del código fuente, en un nivel donde la arquitectura y el diseño se tornan más obvios y más fáciles de entender y modificar. Cuanto más grande es un proyecto, es más importante utilizar una herramienta CASE.

La herramienta CASE utilizada fue el Rational Rose Enterprise Edition que ofrece un entorno de creación de diagramas para UML. Así como, un diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad; el uso de un lenguaje estándar común a todo el equipo de desarrollo. Lo que facilita la comunicación, disponibilidad de múltiples versiones y disponibilidad de integrarse en los principales IDEs. Además, fue la herramienta definida para utilizar en el desarrollo del sistema al cual pertenece el módulo que se va a implementar.

1.6 Sistema gestor de base de datos.

Los sistemas gestores de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Tiene como propósito el de manejar de manera clara, sencilla y ordenada un conjunto de información que por lo general suele ser bastante grande. Algunos de los sistemas más conocidos son: PostgreSQL, MySQL, Microsoft Access, Microsoft SQL Server ,Oracle , etc. Existen distintos objetivos que deben cumplir los sistemas gestores de base de datos como son: (15)

- **Abstracción de la información.** Ahorrar a los usuarios detalles acerca del almacenamiento físico de los datos.
- **Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema (*físico o lógico*) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Redundancia mínima.** Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante.
- **Seguridad.** La información almacenada en una base de datos puede llegar a tener un gran valor por lo que estos sistemas deben garantizar que esta información se encuentra asegurada frente a

usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario.

- **Tiempo de respuesta.** Lógicamente, es deseable minimizar el tiempo que el sistema gestor de base de datos tarda en dar la información solicitada y en almacenar los cambios realizados. (16)

Dentro de los sistemas gestores de base de datos inicialmente mencionados, para el desarrollo de la solución propuesta se decidió utilizar MySQL.

1.6.1 ¿Por que MySQL?

“MySQL es un sistema gestor de base de datos, multihilo y multiusuario con más de seis millones de instalaciones. Desarrollado como software libre, es muy utilizado en aplicaciones web desarrolladas en plataformas Linux/Windows – Apache – MySQL - PHP/Perl/Python. Su popularidad en aplicaciones web está muy ligada a PHP. Es un sistema muy rápido en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a este gestor de Base de datos, ideal para este tipo de aplicaciones.

Es basado en Open Source (*Código abierto*) diseñado para los sistemas Unix formando parte de la tecnología LAMP (*Linux, Apache, MySQL y PHP*), aunque existen versiones para Windows. Actualmente está en su versión 5.0.22 incluyendo procedimientos almacenados (*stored procedures*), disparadores (*triggers*), vistas (*views*) y muchas otras características.

Ventajas: (17)

- Diseñado con el objetivo de aumentar la velocidad.
- Consume muy pocos recursos de CPU y memoria lo que proporciona un alto rendimiento.
- Tamaño del registro sin límite.
- Buena integración con PHP.
- Utilidades de administración (PhpMyAdmin).
- Buen control de acceso usuarios-tablas-permisos”

Desventajas.

- Es gratis para aplicaciones de código abierto, de lo contrario hay que pagar licencia comercial.

Además de todo lo anteriormente expuesto, una de las principales motivaciones al uso de MySQL como sistema gestor de base de datos en la solución propuesta está condicionado porque el personal encargado de darle soporte a la futura aplicación tiene una basta experiencia con el trabajo en este gestor de base de datos no siendo así con Microsoft SQL Server, Oracle y otros. Debido a que el grueso de las aplicaciones que se administran en el área, están desarrolladas sobre MySQL.

1.7 Lenguajes de programación utilizados.

Los lenguajes de programación son creados para poder comunicarse con las computadoras. Un lenguaje de programación es un conjunto limitado de palabras y de símbolos que representan procedimientos, cálculos, decisiones y otras operaciones que pueden ejecutar una computadora.

La programación Web, parte de las siglas WWW, que significan World Wide Web o telaraña mundial, donde una de las cualidades de Internet de los restantes medio de comunicación es la personalización de la información del usuario mediante los diversos lenguajes de programación, estos lenguajes se clasifican en dos grupos: Lenguajes del lado del cliente y lenguajes del lado del servidor.

Dentro del grupo de lenguajes del lado del servidor, los más usados a nivel mundial son ASP, PHP, Java, JSP, PERL, etc. A través de ellos los desarrolladores implementan la lógica de negocio dentro del servidor, además de los accesos a los distintos sistemas gestores de bases de datos.

Dentro del grupo de lenguajes del lado del cliente, algunos de los más usados a nivel mundial son JavaScript, XSLT y el Visual Basic Script, que al combinarse con el HTML forman lo que se conoce como DHTML, es decir, salida estándar dinámica o HTML dinámico.

Esta distinción entre los lenguajes ha sido necesaria debido a que el protocolo http es un protocolo sin estado (*state less*), no guarda información sobre conexiones anteriores y al finalizar la transacción los datos se pierden, cada petición/respuesta es una operación distinta, por lo que la web trabaja en modo desconectado; o sea, un usuario a través de un navegador hace una petición de una página web a un Servidor Web (*Request*), el servidor obtiene la petición, la procesa y le envía la respuesta al cliente (*Response*), este hace la recepción y se desconecta.

1.7.1 Lenguajes utilizados del lado del cliente:

HTML.

“El HTML, acrónimo inglés de HyperText Markup Language (*lenguaje de marcas hipertextuales*), lenguaje de etiquetas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores del tipo Internet Explorer, Opera, Firefox o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos, es una aplicación de SGML conforme al estándar internacional ISO 8879.

JavaScript.

Es un lenguaje interpretado, es decir, que no requiere compilación, orientado a las páginas web. Dirigido por eventos, por lo que estará listo para actuar en cuanto un evento (*un click en un botón, por ejemplo*) sea ejecutado, implementa una sencilla interfaz de objetos/propiedades/métodos. Se integra dentro del código HTML de las páginas Web, se ejecuta en el navegador al mismo tiempo que las sentencias van descargándose junto con el código HTML. Brinda rapidez a la aplicación web a la hora de las validaciones de los formularios. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

AJAX.

Acrónimo de Asynchronous JavaScript And XML (*JavaScript y XML asíncronos, donde XML es un acrónimo de eXtensible Markup Language*). Es una técnica de desarrollo web para crear aplicaciones interactivas que se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

AJAX es una combinación de tres tecnologías ya existentes: (18)

- **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.”

1.7.2 Lenguajes utilizados del lado del servidor:

PHP

“Es un lenguaje de programación usado generalmente para la creación de contenido para sitios web. Las siglas significan "PHP Hypertext Pre-processor" (*inicialmente PHP Tools, o, Personal Home Page Tools*), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, Microsoft SQL Server, etc ; lo cual permite la creación de aplicaciones web muy robustas. Tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (*y de ese tipo, como Linux*), Windows y Mac OS X, y puede interactuar con los servidores de web más populares como Apache.

Es gratuito e independiente de plataforma, muy rápido, con una gran librería de funciones y mucha documentación. En el caso de estar montado sobre un servidor Linux o Unix, es uno de los más rápidos, dado que se ejecuta en un único espacio de memoria.

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

- Cubre más del 75% de las necesidades del mercado.
- Es multiplataforma, es decir, puede ser utilizado en cualquiera de los principales sistemas operativos del mercado actual y es soportado por la mayoría de los servidores Web.
- Es software libre, lo que implica menos costos y servidores más baratos, por lo que se puede utilizar en proyectos sin tener que pagar por su licencia.

- El tiempo de aprendizaje de PHP es muy corto gracias a su simplicidad
- Es muy rápido y su integración con el gestor de base de datos MySQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado.
- Su librería estándar es realmente amplia, lo que permite reducir los llamados "costos ocultos", uno de los principales defectos de ASP.

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

1.7.3 Motor de plantilla utilizado.

El mecanismo de plantillas sirve para separar el código de presentación del resto del código de una aplicación web. Consiste en codificar todo lo que tenga que ver con la presentación en una serie de plantillas de código HTML (*u otro lenguaje de presentación*), con expresiones sencillas intercaladas para comunicarse con el resto de la aplicación y poder mostrar datos dinámicos. Un motor de plantillas es el que se encarga de hacer la traducción a HTML.

De esta forma, las vistas simplemente se ocupan de extraer la información necesaria del modelo y comunicársela a las plantillas, y estas simplemente de dar un formato visual a esa información, añadiendo la información estática pertinente.

El motor de plantilla utilizado en el desarrollo fue **Smarty**, se encuentra bajo la licencia GPL, por lo que puede ser usado libremente.

Algunas de las características de Smarty: (19)

- Es extremadamente rápido.
- Es eficiente ya que puede interpretar el trabajo más sucio.
- No analiza gramaticalmente desde arriba el template, solo compila una vez.
- Está atento para solo recompilar los archivos de plantilla que fueron cambiados.
- Se puede crear funciones habituales y modificadores de variables personalizadas, de modo que el lenguaje de la platilla es altamente extensible.
- Los constructores `if/elseif/else/endif` son pasados por el interpretador de PHP, así la sintaxis de la expresión `{if ...}` puede ser compleja o simple de la forma que usted quiera.
- Permite un anidamiento ilimitado de sections, ifs, etc.
- Es posible incrustar directamente código PHP en los archivos de plantilla.

- Funciones habituales de manipulación de cache
- Arquitectura de Plugin”

1.8 ¿Por qué usar PHP y MySql?

No todos los sistemas operativos pueden correr todo tipo de hardware; no todos los lenguajes de programación se conectan a todas las bases de datos, etc. Una de las ventajas de PHP y MySQL es que pueden correr tanto en Unix como en Windows. PHP permite las técnicas de programación orientada a objetos y lo fundamental es que brinda una gran cantidad de funciones internas que facilitan el trabajo con MySQL y es muy rápido tanto en la velocidad de ejecución como en la velocidad para desarrollar el código.

Basado en algunas experiencia de desarrolladores en otros lenguajes de programación una de las mejores combinaciones entre lenguaje de programación y sistema gestor de base de datos está dada por PHP y MySql, además ambos son libres. Actualmente existe una comunidad de millones de programadores de todo el mundo donde se puede encontrar ayuda y cualquier tipo de documentación. Lo cual es muy importante ya que nadie es experto en todo y siempre se necesitan consejos de gente que sabe más o que tiene experiencia en otras áreas.

1.9 Herramientas utilizadas en el desarrollo.

Las herramientas de desarrollo de software ayudan al equipo de desarrollo a diseñar y construir la aplicación. Estas son de vital importancia para el buen desempeño de la aplicación, la selección de las mismas está determinado por las tecnologías en que se desarrolle la aplicación en este caso las herramientas usadas fueron:

1.9.1 Dreamweaver 8.

Dreamweaver 8 es la herramienta de desarrollo Web líder del mercado, permite diseñar, desarrollar y mantener de forma eficaz sitios y aplicaciones Web basados en normas, lo abarca todo, desde la creación y mantenimiento de sitios Web básicos hasta aplicaciones avanzadas compatibles con las mejores prácticas y las tecnologías más recientes.

1.9.2 Rational Suite 2003.

Rational Rose es la herramienta CASE desarrollada por los creadores de UML (*Booch, Rumbaugh y Jacobson*), que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables.

Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros de equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de software (UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto.

Rational Rose permite completar una gran parte de las disciplinas (flujos fundamentales) del proceso unificado de Rational (RUP), en concreto:

- Modelado del negocio.
- Captura de requisitos.
- Análisis y diseño.
- Implementación.
- Control de cambios y gestión de configuración.

Características Principales:

- Admite como notaciones: UML, COM, OMT y Booch.
- Realiza Chequeo semántico de los modelos.
- Ingeniería “de ida y vuelta”: Rose permite generar código a partir de modelos y viceversa.
- Desarrollo multiusuario.
- Integración con modelado de datos.
- Generación de documentación.
- Tiene un lenguaje de script para poder ampliar su funcionalidad.

1.9.3 EMS 2005 y MySql Administrator.

Estas son herramientas para el trabajo con el sistema gestor de base de datos, que facilita toda la gestión de usuarios y configuración, además para la realización de consultas, procedimientos almacenados, vistas etc.

1.9.4 ZendStudio 5.2.0

Se trata de un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones web en lenguaje PHP. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

1.9.5 Sistema de control de versión utilizado.

Se llama control de versiones a la gestión de versiones (*revisiones*) de todos los elementos de configuración que forman la línea base de un producto o una configuración del mismo. Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado junto a las posibles especializaciones realizadas para algún cliente específico. En la industria informática el control de versiones se realiza principalmente para controlar las distintas versiones del código fuente creadas por los desarrolladores.

Aunque un sistema de control de versiones puede realizarse de forma manual, es muy aconsejable disponer de herramientas que faciliten esta gestión, algunas de las más usadas y conocidas se encuentran CVS, Subversion, SourceSafe, ClearCase, Darcs, Plastic SCM, etc. Para el entorno de desarrollo de la solución propuesta se utiliza el Subversion que surge diseñado específicamente para reemplazar al popular CVS, el cual posee varias deficiencias. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como svn por ser ese el nombre de la herramienta de línea de comandos que utiliza. Una característica importante de Subversion es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo.

Ventajas

- Se sigue la historia de los archivos y directorios a través de copias y renombrados.
- Las modificaciones (*incluyendo cambios a varios archivos*) son atómicas.

- El creado de ramas y etiquetas es una operación más eficiente; Tiene costo de complejidad constante ($O(1)$) y no lineal ($O(n)$) como en CVS.
- Se envían sólo las diferencias en ambas direcciones (*en CVS siempre se envían al servidor archivos completos*).
- Puede ser servido, mediante Apache, sobre WebDAV/DeltaV. Esto permite que clientes WebDAV utilicen Subversion en forma transparente.
- Maneja eficientemente archivos binarios (*a diferencia de CVS que los trata internamente como si fueran de texto*).
- Permite selectivamente el bloqueo de archivos. Se usa en archivos binarios que, al no poder fusionarse fácilmente, conviene que no sean editados por más de una persona a la vez.
- Cuando se usa integrado a Apache permite utilizar todas las opciones que este servidor provee a la hora de autenticar archivos (*SQL, LDAP, PAM, etc*).

Clientes

Existen varias interfaces a Subversion, ya sea programas individuales como interfaces que lo integran en entornos de desarrollo. El utilizado en este caso fue **TortoiseSVN** debido a que es un cliente Subversion, implementado como una extensión al shell de Windows, y además es software libre liberado según la licencia GNU GPL.” (19)

Conclusiones:

Este capítulo, se estudiaron algunos sistemas existentes relacionados con la gestión académica a nivel nacional. Se definieron los principales conceptos relacionados con el sistema y se caracterizaron los lenguajes, tecnologías y metodologías utilizadas para el desarrollo del componente.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En el presente capítulo se hace la descripción de la propuesta de solución, para ello se detallan los procesos del negocio que tienen que ver con el objeto de estudio, que dan paso a representar posteriormente el problema en términos informáticos.

Se enumeran además los requisitos funcionales y no funcionales que debe tener el sistema, lo cual permite hacer una concepción general de lo que se quiere desarrollar. Con los requisitos definidos, se realiza el Modelo de Casos de Uso, se identifican los actores y sus relaciones con el sistema representándose en el Diagrama de Casos.

2.1 Conceptos asociados al dominio del problema.

Secretaría Docente: Oficina encargada de garantizar, tramitar, y legalizar todos los procesos académicos que se generan desde la entrada del estudiante al centro docente hasta su salida.

Plan de Estudio: Documento elaborado por especialistas del Ministerio de Salud Pública que contienen un grupo de asignaturas que se van a impartir a estudiantes con un perfil determinado. Este documento es tramitado a nivel ministerial a los centros docentes donde se pondrá en vigor este plan de estudio.

Rematrícula: Proceso que se lleva a cabo cuando un estudiante culmina un año académico y va a continuar en la institución cursando estudios el próximo semestre.

Adelantar Asignatura: Cursar una asignatura que se encuentra en un semestre posterior al que está en vigor junto a las demás que corresponden al estudiante según el plan de estudio asignado.

Elemento que aporta bonificación: Se refiere a un evento, prueba o evaluación que se define a nivel nacional por los metodólogos del MINSAP, que van a aportar una bonificación al promedio de un estudiante.

2.2 Objeto de estudio.

2.2.1 Situación Problemática.

En las instituciones de la Educación Superior pertenecientes al sector de salud, el proceso de gestión académica que se realiza en las secretarías docentes se encuentra en estado deficiente. El registro y control de las evaluaciones de los estudiantes por año, la reafirmación de la matrícula y la asignación de planes de estudio, encuentra a diario obstáculos ya que no se cuenta con un sistema estándar que responda con eficiencia y calidad a las necesidades actuales del MINSAP.

En la actualidad existe un proyecto con sede en la Universidad de las Ciencias Informáticas que intenta dar solución a los problemas existentes en el proceso de Formación de Recursos Humanos en Salud. Este sistema se encuentra en fase de construcción y cuenta en la actualidad con 2 módulos liberados: Matrícula y Movimientos, correspondientes a un primer ciclo de desarrollo del producto.

Aunque esta aplicación posee una serie de servicios que amenizan el trabajo en las secretarías, todavía no se ha desarrollado en su totalidad, por lo que carece de funcionalidades que respondan a las insuficiencias vigentes en el proceso de gestión académica.

2.2.2 Descripción de los procesos del negocio.

- **Proceso de confección del Plan de Estudio.**

Los metodólogos nacionales que trabajan para el Ministerio de Salud Pública son los encargados de confeccionar un plan de estudio. Este pertenece a una carrera en específico y está conformado por períodos que contienen asignaturas propias de una especialidad. Cada período contiene diferentes tipos de evaluaciones como son: cortes evaluativos, evaluaciones finales, extraordinarias y mundiales. Además se planifican exámenes especiales que pueden formar parte de un período.

- **Proceso de asignar un Plan de Estudio para cada estudiante.**

Los planes de estudio, como se define en el proceso anterior, recogen todas las asignaturas que va a recibir el estudiante, agrupadas en períodos y con tipos de evaluaciones determinados. Cuando un alumno va a ingresar en una facultad médica para recibir sus estudios superiores, se le asigna un plan de estudio por el que se va a regir durante el transcurso de su carrera. Este plan de estudio puede cambiar en

dependencia de los intereses de la dirección del centro donde radica o específicamente del MINSAP. Esta decisión no depende solo del centro de estudios sino que se define a nivel nacional.

- **Proceso de control y registro de las Evaluaciones.**

Esta es una de las tareas más importante que se realizan en una secretaría docente, y sólo el personal que trabaja en ella podrá reflejar en el expediente académico del estudiante las calificaciones alcanzadas por este. Existen distintos momentos en el curso para reflejar los resultados docentes alcanzados por el estudiante en el expediente los cuales son: convocatorias ordinarias y extraordinarias de fin de semestre y convocatorias ordinarias y extraordinarias de fin de curso. Se producen además cortes evaluativos en cada semestre cuyos resultados no se reflejan en el expediente pero sí es de importancia procesarlos para conocer los resultados.

Es de vital importancia conocer de forma rápida y precisa el estado académico de uno o varios estudiantes después de la convocatoria extraordinaria de fin de curso o en cualquier otro momento para estudiar parámetros y trazar medidas para mejorar los resultados docentes de forma general y de esta forma mejorar cada vez más la formación del profesional que necesita la Revolución.

- **Proceso de Rematricular.**

El proceso de rematrícula se realiza cuando en el centro docente culmina un curso académico. Esta acción corresponde a la decisión de un estudiante de continuar sus estudios en esta carrera. Para esto es necesario conocer si el alumno tiene aprobadas todas las asignaturas del plan de estudio correspondientes y en caso contrario, la dirección de la facultad decide si se le otorga la posibilidad de continuar sus estudios en el centro. Si la decisión es a favor del estudiante, cursaría el próximo año solo con la(s) asignatura(s) desaprobadas.

- **Proceso de Adelantar Asignaturas.**

El plan de estudio está concebido para impartir un grupo determinado de asignaturas en un semestre o período. Si un estudiante desea cursar una o más asignaturas extras, puede hacerlo presentándose en la secretaria docente y pedir Adelantar. Para esto se debe tener en cuenta que esta asignatura no posea precedencia, o sea, que no dependa de otra para poder cursarla, si se da el caso, se debe tener aprobada

la que antecede. Cuando se está adelantando una asignatura, esta influye en el período tanto como las demás, por lo que el estudiante tiene la responsabilidad de aprobarla.

2.3 Propuesta del sistema.

El Componente de Gestión Académica forma parte del Sistema de Gestión de Información en el Proceso de Formación de los Recursos Humanos en Salud. Se implementa como parte de un segundo ciclo de desarrollo del proyecto inicial y contiene características propias referentes al proceso de gestión académica que se lleva a cabo en todos los centros clínico docentes del país.

El sistema, en su totalidad, se encontrará brindando servicios sobre la red telemática de salud cubana INFOMED a través de la cual todos los nodos (*secretarías docentes, puestos de trabajos relacionados con el sistema*) accederán al sistema. Se contará con un servidor centralizado al que se conectarán las instituciones de todo el país. A continuación se muestra un esquema sobre cómo estará distribuida la aplicación.



Fig: 2.1 Distribución de la aplicación.

Después de identificar cuáles son los procesos que se quieren automatizar, se procede a modelar el negocio, especificar los requerimientos que va a cumplir la aplicación en respuesta las necesidades existentes y definir los casos de uso que va a tener el sistema.

2.4 Modelo de negocio.

Para informatizar los procesos correspondientes al problema en cuestión, es necesario conocer cómo se desarrollan los mismos a nivel de entidad. Con este objetivo, el Modelo de negocio permite tener una visión clara de cómo fluye toda la información en estas organizaciones y por lo tanto definir procesos, roles y responsabilidades que serán reflejadas en los modelos de casos de uso del negocio y de objetos plasmados en el **Anexo1**.

2.5 Especificación de los requisitos de software.

2.5.1 Requisitos Funcionales.

Una vez conocidos los conceptos que rodean al objeto de estudio, se debe analizar: ¿Qué debe hacer el sistema para que se cumplan los objetivos planteados al inicio de este trabajo?, para ello se enumeran, a través de requerimientos funcionales, las acciones que el sistema deberá ser capaz de realizar.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Para este sistema se definen los siguientes:

Nivel Nacional

R1: Gestionar plan de estudio.

- 1.1 Crear un nuevo plan de estudio.
- 1.2 Editar el nombre de un plan de estudio existente.
- 1.3 Eliminar un plan de estudio existente.
- 1.4 Adicionar un nuevo Estudio.
- 1.5 Editar un Estudio existente.
- 1.6 Eliminar un Estudio existente.
- 1.7 Adicionar un nuevo Período.
- 1.8 Editar el nombre de un Período existente.
- 1.9 Eliminar un Período existente.
- 1.10 Adicionar una Asignatura.
- 1.11 Editar el nombre de una Asignatura existente.
- 1.12 Eliminar una Asignatura existente.
- 1.13 Adicionar un nuevo Tipo de Evaluación.

- 1.14 Editar el nombre de un Tipo de Evaluación existente.
- 1.15 Eliminar un Tipo de Evaluación existente.
- 1.16 Adicionar un nuevo Examen Especial.
- 1.17 Editar un Examen Especial existente.
- 1.18 Eliminar un Examen Especial existente.
- 1.19 Conformar un Plan de Estudio con todos sus elementos.
- 1.20 Distribuir un Plan de Estudio a las sedes universitarias médicas.
- 1.21 Mostrar un Plan de Estudio existente con todos sus elementos.

R2: Gestionar elementos que aportan bonificación.

- 2.1 Adicionar un elemento que aporte bonificación al promedio final del estudiante.
- 2.2 Editar un elemento que aporte bonificación existente.
- 2.3 Eliminar un elemento que aporte bonificación
- 2.4 Mostrar todos los elementos que aportan bonificación
- 2.5 Nivel Sede Universitaria

R3: Asignar un plan de estudio.

- 3.1 Mostrar de un plan de estudio seleccionado con todos sus elementos.
- 3.2 Mostrar las brigadas que no poseen un plan de estudio vigente y por lo tanto se encuentran disponibles para que se le asigne uno especificado.
- 3.3 Mostrar las brigadas que pertenecientes a un plan de estudio especificado.

R4: Gestionar evaluaciones del curso académico.

- 4.1 Registrar y modificar notas (cualitativas, cuantitativas, de exámenes especiales y elementos que aportan bonificación), según el tipo de evaluación, de las asignaturas de cada período de la carrera para un estudiante buscado.
- 4.2 Registrar y modificar notas (cualitativas, cuantitativas y de exámenes especiales), según el tipo de evaluación, de las asignaturas de cada período de la carrera para los estudiantes de un grupo buscado.

R5: Mostrar resultados docentes de un estudiante.

- 5.1 Calcular índice académico acumulado de un estudiante durante los años académicos que ha cursado.
- 5.2 Mostrar las bonificaciones al promedio pertenecientes a un estudiante.
- 5.3 Generar reporte de todas las evaluaciones de un estudiante buscado durante su carrera separadas por años académicos y por períodos.

R6: Generar reportes de las evaluaciones de los estudiantes.

- 6.1 Generar reporte de las evaluaciones cualitativas de un corte evaluativo de un grupo buscado.
- 6.2 Generar reporte de las evaluaciones cuantitativas finales de un grupo buscado.
- 6.3 Generar reporte de todas las evaluaciones de corte evaluativo de un grupo buscado.
- 6.4 Generar reporte de todas las evaluaciones finales de un grupo buscado

R7: Adelantar una asignatura.

R8: Rematricular un estudiante.

- 8.1 Mostrar asignaturas aprobadas y pendientes de un estudiante buscado.

2.5.2 Requisitos No Funcionales.

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener.

2.5.2.1 Usabilidad.

El sistema podrá ser utilizado por cualquier persona, 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 ante las solicitudes 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 aquellas que por cuestiones de seguridad,

o de acceso a los datos lo requieran. Lográndose 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 del mismo a través de versiones posteriores.

2.5.2.4 Apariencia o Interfaz externa.

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.

2.5.2.5 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 código JavaScript.

2.5.2.6 Seguridad.

Teniendo en cuenta que la aplicación almacenará información sensible referente a todo el proceso de formación de estudiantes, de la que dependerá la toma de decisiones del Estado:

La información estará protegida contra accesos no autorizados utilizando mecanismos de autenticación y autorización que puedan garantizar el cumplimiento de esto: usuario, contraseña y nivel de acceso dado por la asignación de roles (nacional y sede).

Los datos de entrada al componente se validan tanto en el cliente como en el servidor.

Se guardan además, las trazas de adición, edición y eliminación en las tablas de la base de datos, donde se recoge el nombre de la tabla, la operación realizada sobre esta, el usuario que la ejecutó y la fecha.

No obstante los usuarios accederán de manera rápida y operativa al sistema sin que los requerimientos de seguridad se conviertan en un retardo para ellos.

2.5.2.7 Requerimientos de Software.

- MySQL versión 5.0.18 o cualquier otra que soporte trabajo con procedimientos almacenados, vistas.
- Servidor Apache versión 2.0.55 o compatible.
- PHP versión 5 que soporte el trabajo orientado a objetos.

2.5.2.8 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)

2.6 Definición de los casos de uso del sistema.

Utilizando las facilidades que brinda el UML, se representarán los requisitos funcionales del sistema mediante un diagrama de casos de uso. Para ello hay que definir de acuerdo a lo planteado en los epígrafes anteriores, cuales serían los actores que van a interactuar con el sistema y los casos de uso que van a representar las funcionalidades.

Un caso de uso es un artefacto narrativo que describe la secuencia de un actor (agente externo) que utiliza un sistema para completar un proceso. Un actor no es parte del sistema, sino un rol que se juega dentro del sistema, que puede intercambiar información o puede ser un recipiente pasivo de información y representa a un ser humano, a un software o a una máquina que interactúa con el sistema.

En el **Anexo2** se encuentra descrito el modelado del sistema donde se explican y justifican los actores del sistema, los casos de uso priorizados en ciclos de desarrollo, descritos y representados en diagramas de caso de usos del sistema.

Conclusiones:

En este capítulo se comenzó a desarrollar la propuesta de solución por lo que se obtiene una perspectiva de lo que se desea construir. En consecuencia se generaron todos los artefactos que propone RUP a partir del análisis de los procesos del negocio para el flujo de trabajo de Modelación del Negocio y Requerimientos.

Por lo definido hasta el momento, se puede comenzar con la construcción del componente asegurando que se cumplan todos los requerimientos y las funcionalidades que han sido consideradas necesarias en este capítulo.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Definidas las funcionalidades que va a tener el sistema el siguiente paso corresponde a concretar cómo se desarrollará el componente. En consecuencia en este capítulo se especifican el análisis y diseño, paso muy importante en el proceso de construcción de la aplicación ya que constituyen la vista lógica de la arquitectura del software. Para realizar los modelos necesarios de esta etapa se utiliza el lenguaje UML, a través del cual se brindara una vista práctica que mejorara el entendimiento de lo que se quiere y la forma en que se desea implementar el componente.

Para comprender el análisis realizado, se definieron tres tipos de clases: interfaz, controladora y entidad. En todos los casos se define su vínculo con cada caso de uso y las relaciones que tienen entre ellas. El diseño es un refinamiento del análisis, por lo que se especifica con mayor claridad la interacción entre los actores y el sistema. Se describen además las clases que se representan en el diseño, al igual que se obtiene el Diagrama Entidad-Relación de la base de datos y la descripción de sus tablas.

3.1 Análisis

A la hora de desarrollar una solución software a medida, es imprescindible analizar convenientemente los procesos. Mediante un buen análisis se garantizará un trabajo con éxito ya que es la base de todo lo que se realizará posteriormente.

A partir de esta etapa, se hace un refinamiento de los casos de uso brindando una mayor comprensión y claridad de su funcionamiento en el sistema. Seguidamente se representarán los diagramas de clases del análisis según el caso de uso que corresponda dividiéndolos por paquetes adecuados a los dos niveles de acceso al sistema.

3.1.1 Paquete Administración Académica.

El paquete de administración cuenta con dos casos de uso: Gestionar Plan de Estudio y Gestionar Elementos que Aportan Bonificación. En el primer caso, este se compone por once nomencladores correspondientes a los elementos que constituyen un plan de estudio por lo que a continuación se representa solo el diagrama de clases del análisis del nomenclador nombre de plan de estudio por poseer los demás características similares.

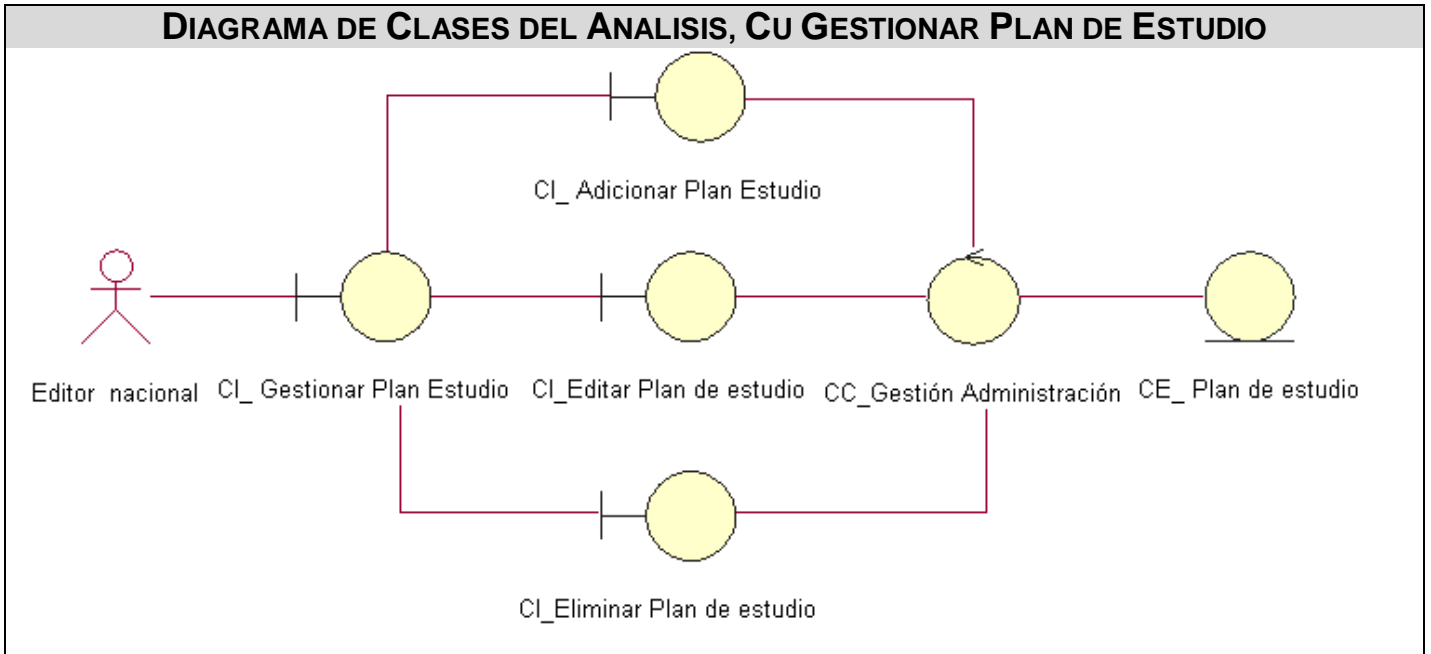


Fig. 3.1 Diagrama de clases del análisis del nomenclador nombre de plan de estudio perteneciente al caso de uso Gestionar Plan de Estudio.

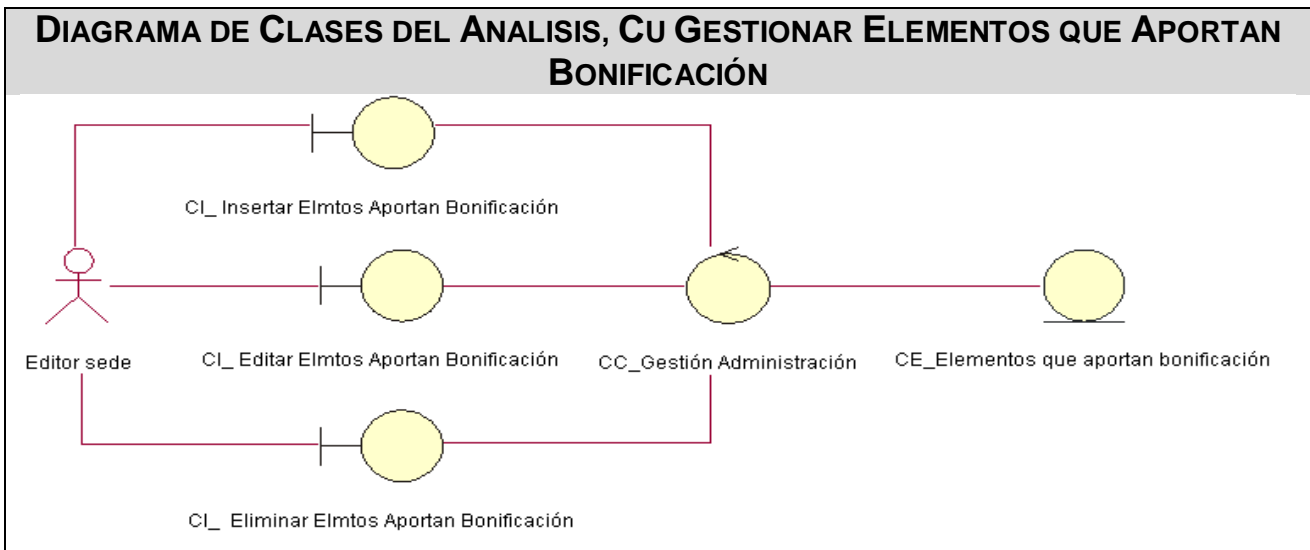


Fig. 3.2 Diagrama de clases del análisis del caso de uso Gestionar Elemento que Aportan Bonificación.

3.1.2 Paquete de Gestión Académica.

Este paquete cuenta con 5 casos de uso que darán solución a los procesos que se llevan a cabo en las propias sedes universitarias médicas.

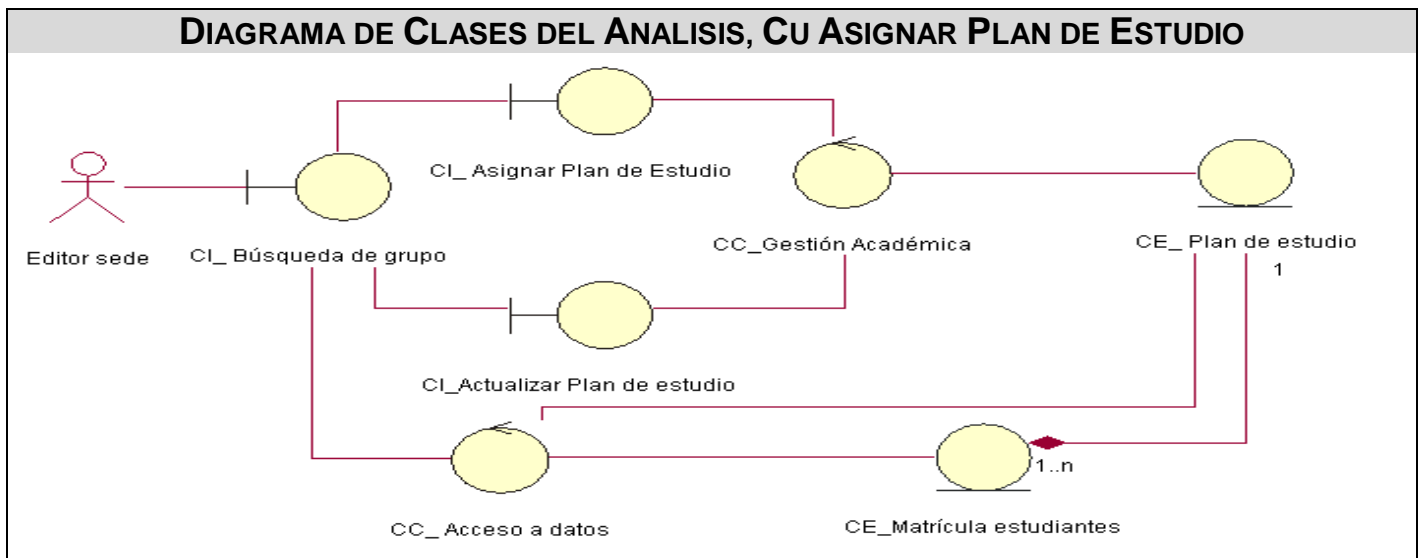


Fig: 3.3 Diagrama de clases del análisis del caso de uso Asignar Plan de Estudio.

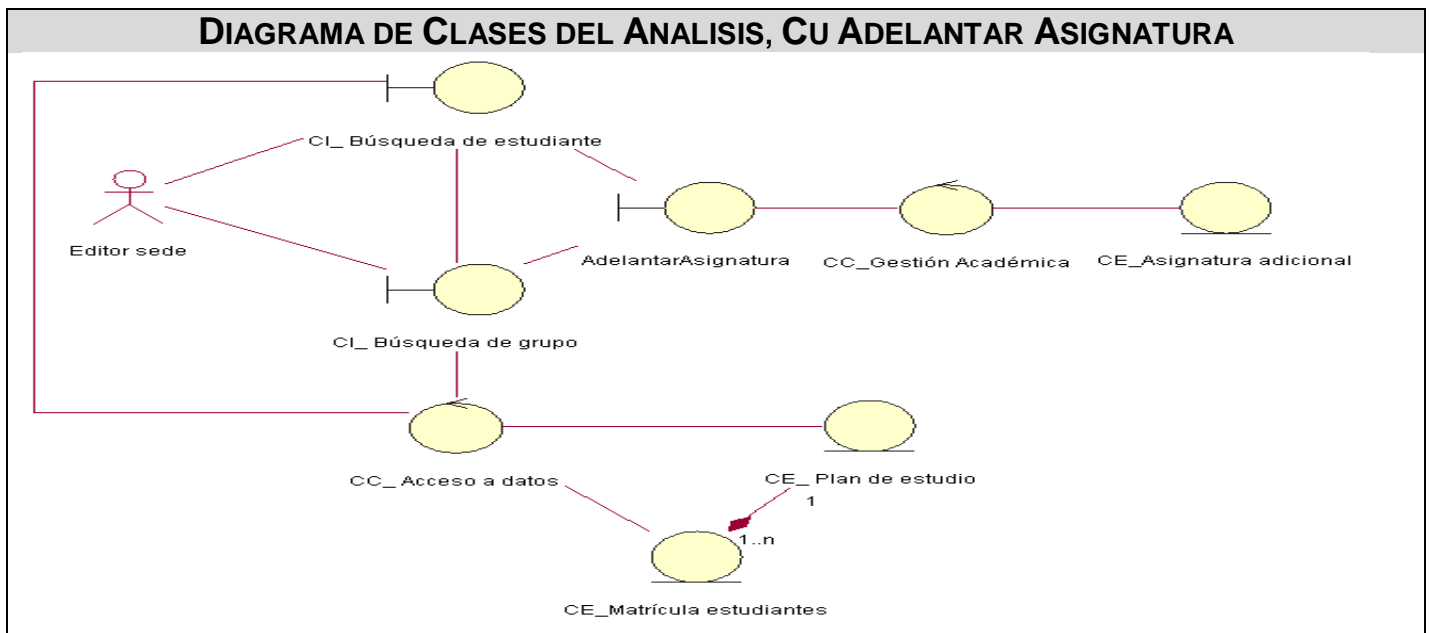


Fig: 3.4 Diagrama de clases del análisis del caso de uso Adelantar Asignatura.

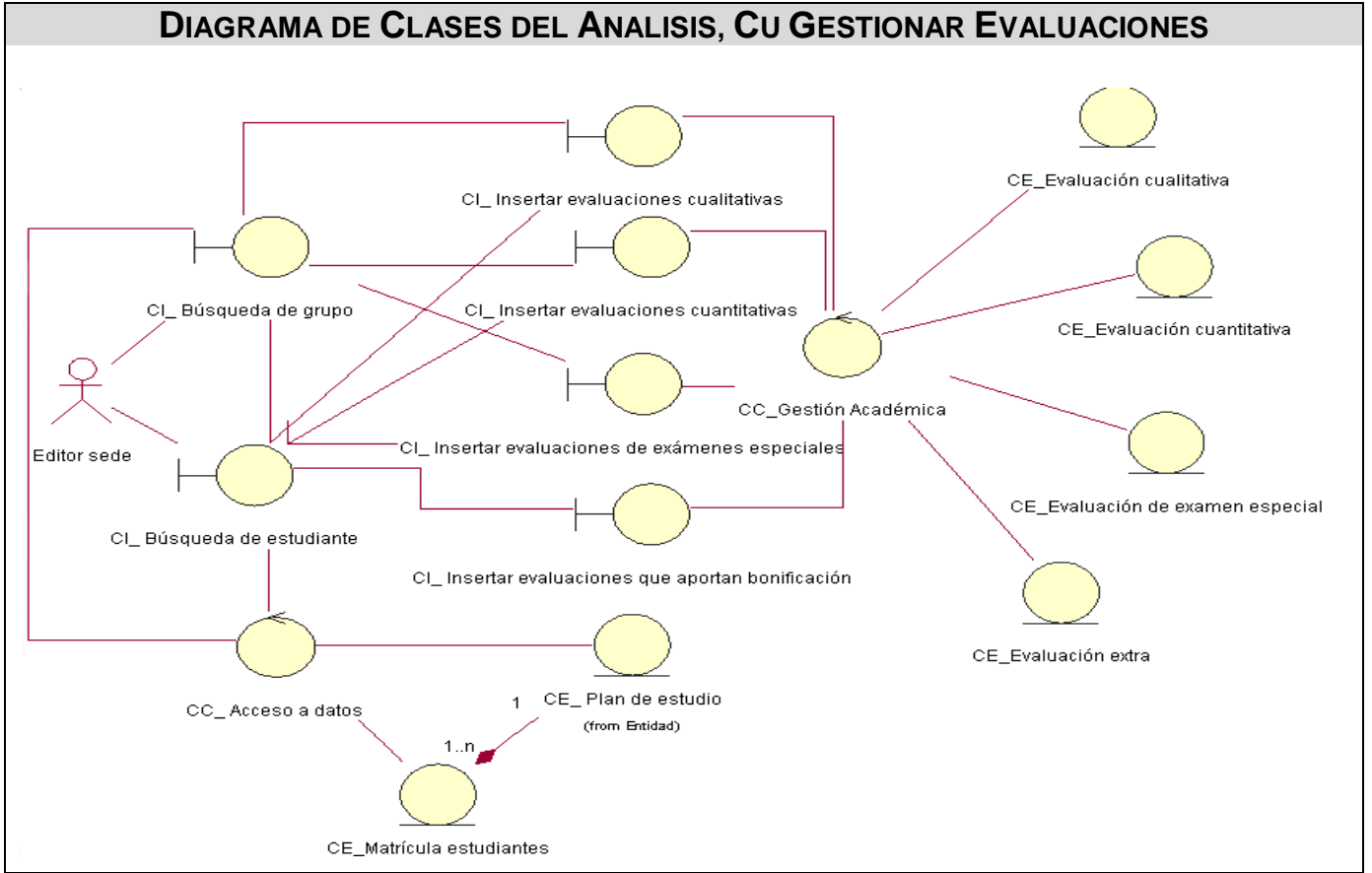


Fig: 3.5 Diagrama de clases del análisis del caso de uso Gestionar Evaluaciones.

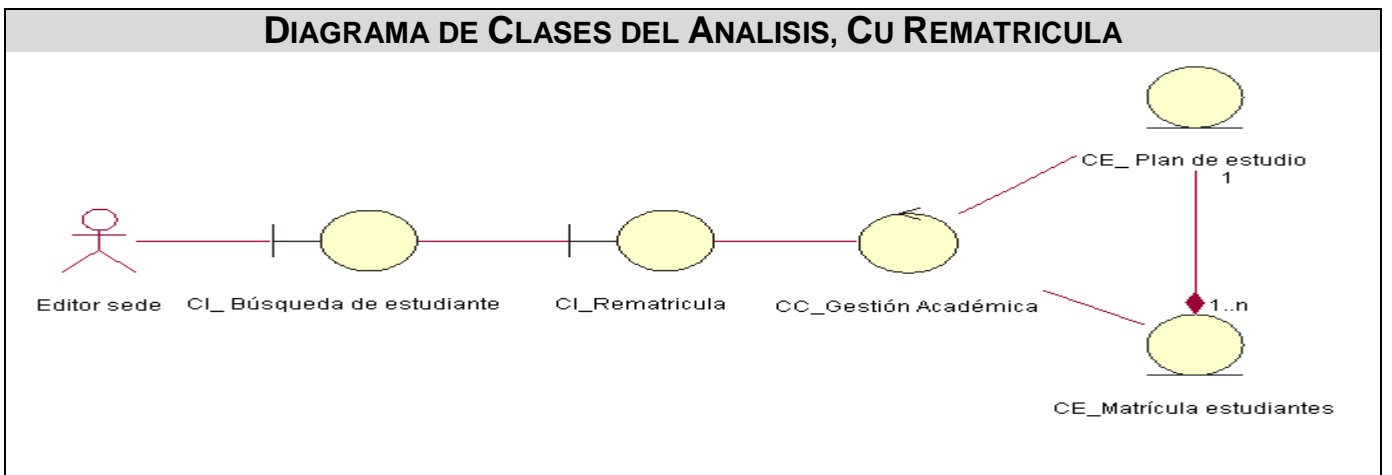


Fig: 3.6 Diagrama de clases del análisis del caso de uso Rematricula.

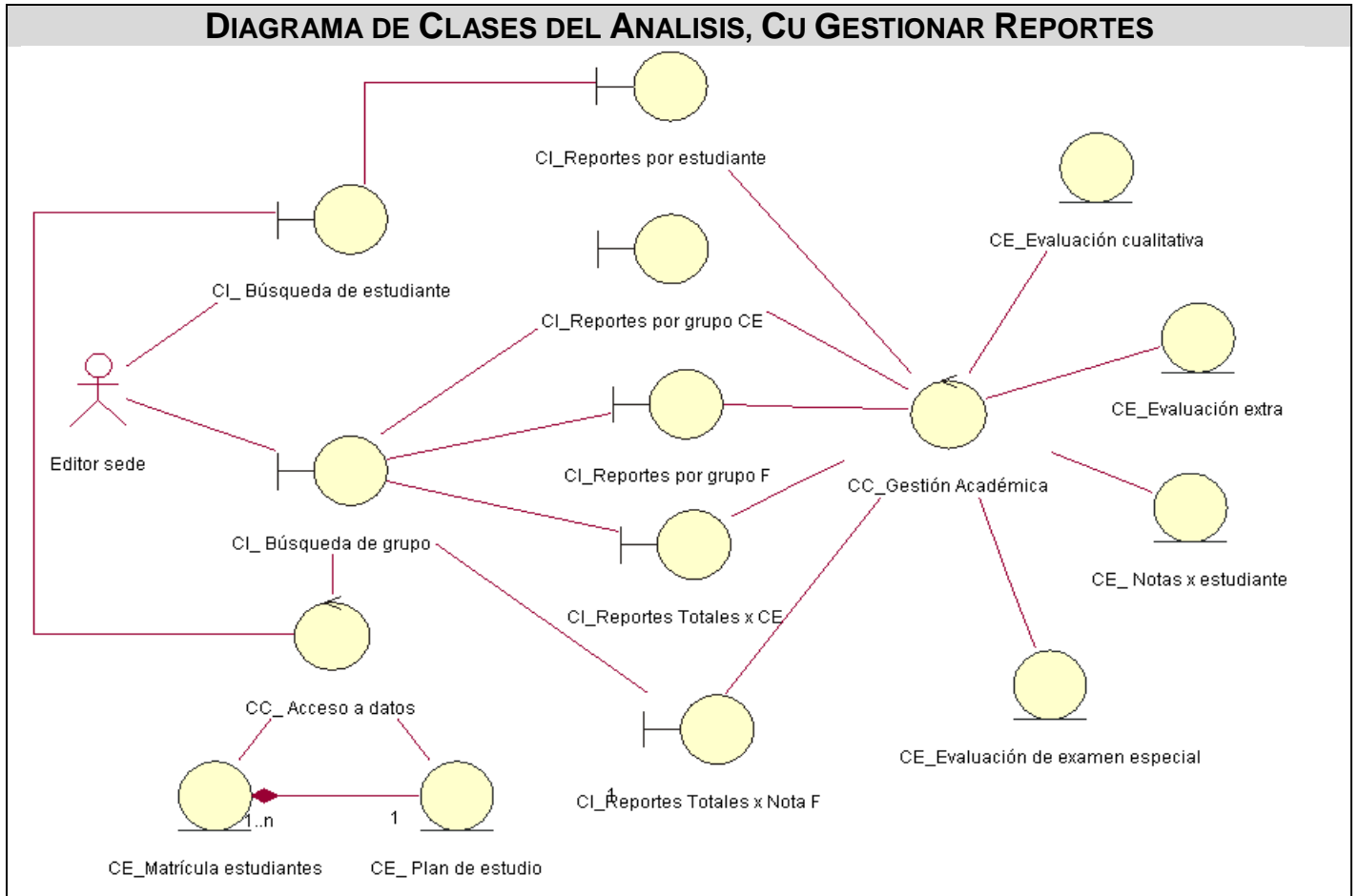


Fig: 3.7 Diagrama de clases del análisis del caso de uso Gestionar Reportes.

3.2 Diseño.

Durante el ciclo de desarrollo iterativo, es posible pasar a la fase de diseño una vez culminado el análisis. Se logra una solución lógica que se fundamente en el paradigma orientado a objetos. A medida que se avanza en esta etapa, se modela con más precisión cómo va a estar estructurado el sistema para dar soporte a los requisitos funcionales y no funcionales existentes. Además de constituir un punto de partida para realizar la implementación. Entre los artefactos principales que se dan a conocer en el presente

capítulo, se encuentra el Modelo de Diseño que contiene los Diagramas de Clases del Diseño y los Diagramas de Interacción, mediante los cuales se muestra la realización física de los casos de uso.

3.2.1 Diagramas de Clases del Diseño.

Los diagramas de clases del diseño resumen la definición de las clases (e interfaces) que se implementan en el software. En los diagramas siguientes se muestra la interacción entre las clases del diseño definidas para cada caso de uso en particular.

3.2.1.1 Paquete Administración Académica.

En este caso, como en el diagrama de clases del análisis del caso de uso Gestionar Plan de Estudio, se muestra solo el diagrama de clases del diseño del nomenclador Nombre de Plan de Estudio por presentar características similares con los demás nomencladores que se definen para este caso de uso.

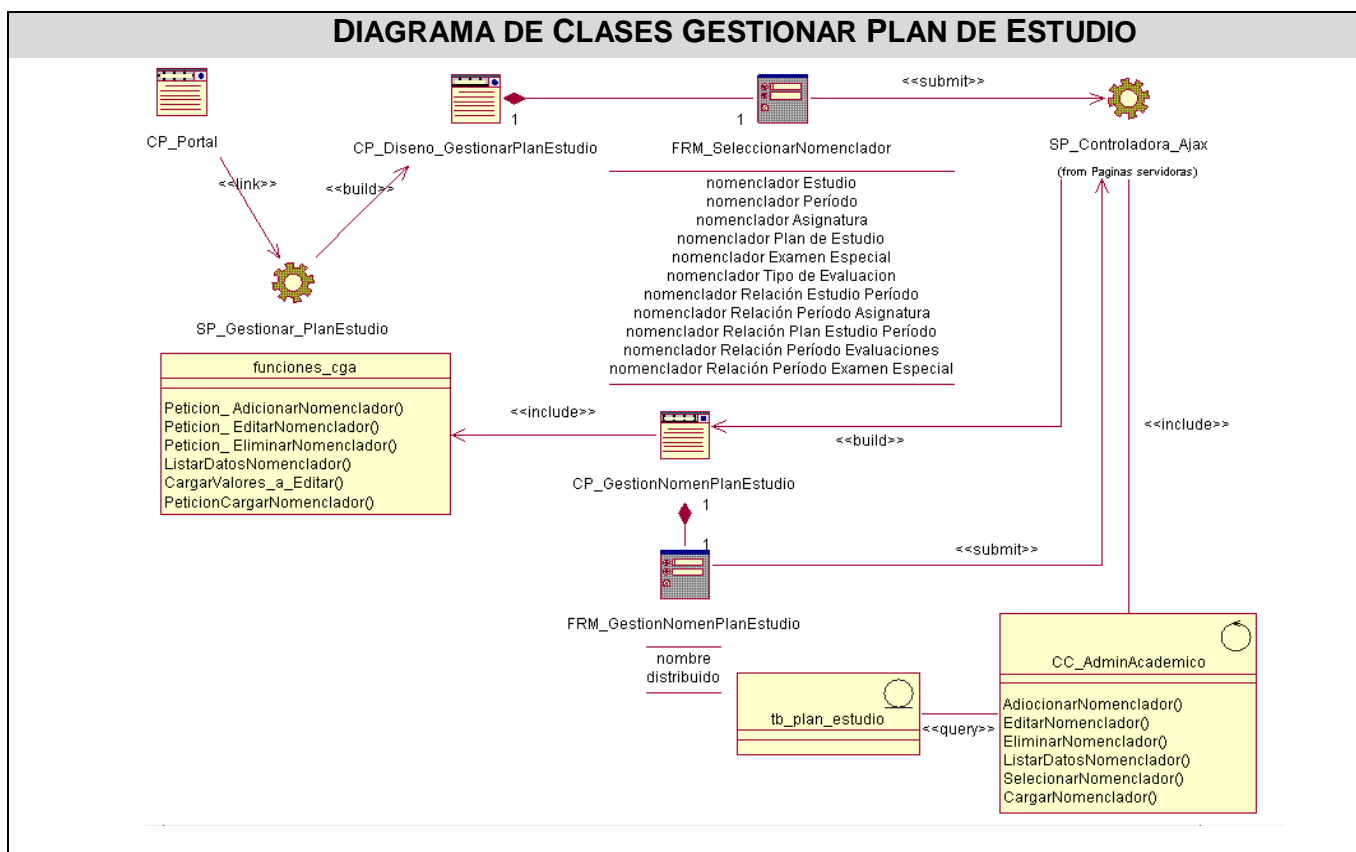


Fig 3.8 Diagrama de clases del diseño del nomenclador Nombre de Plan de estudio perteneciente al caso de uso Gestionar Plan de Estudio.

DIAGRAMA DE CLASES DE GESTIONAR ELEMENTOS QUE APORTAN BONIFICACIÓN

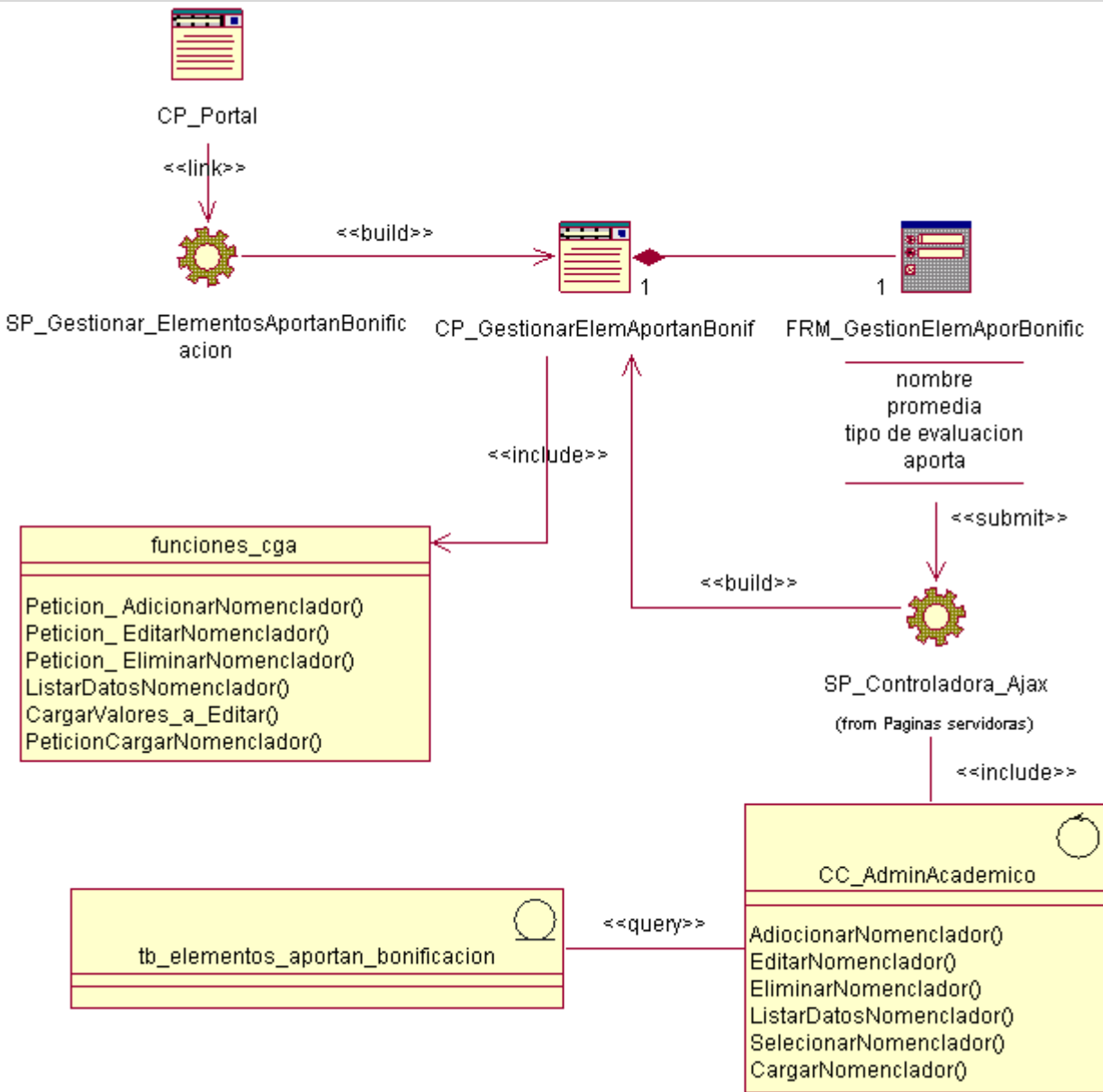


Fig: 3.9 Diagrama de clases del diseño del caso de uso Gestionar Elemento que Aportan Bonificación.

3.2.1.2 Paquete de Gestión Académica.

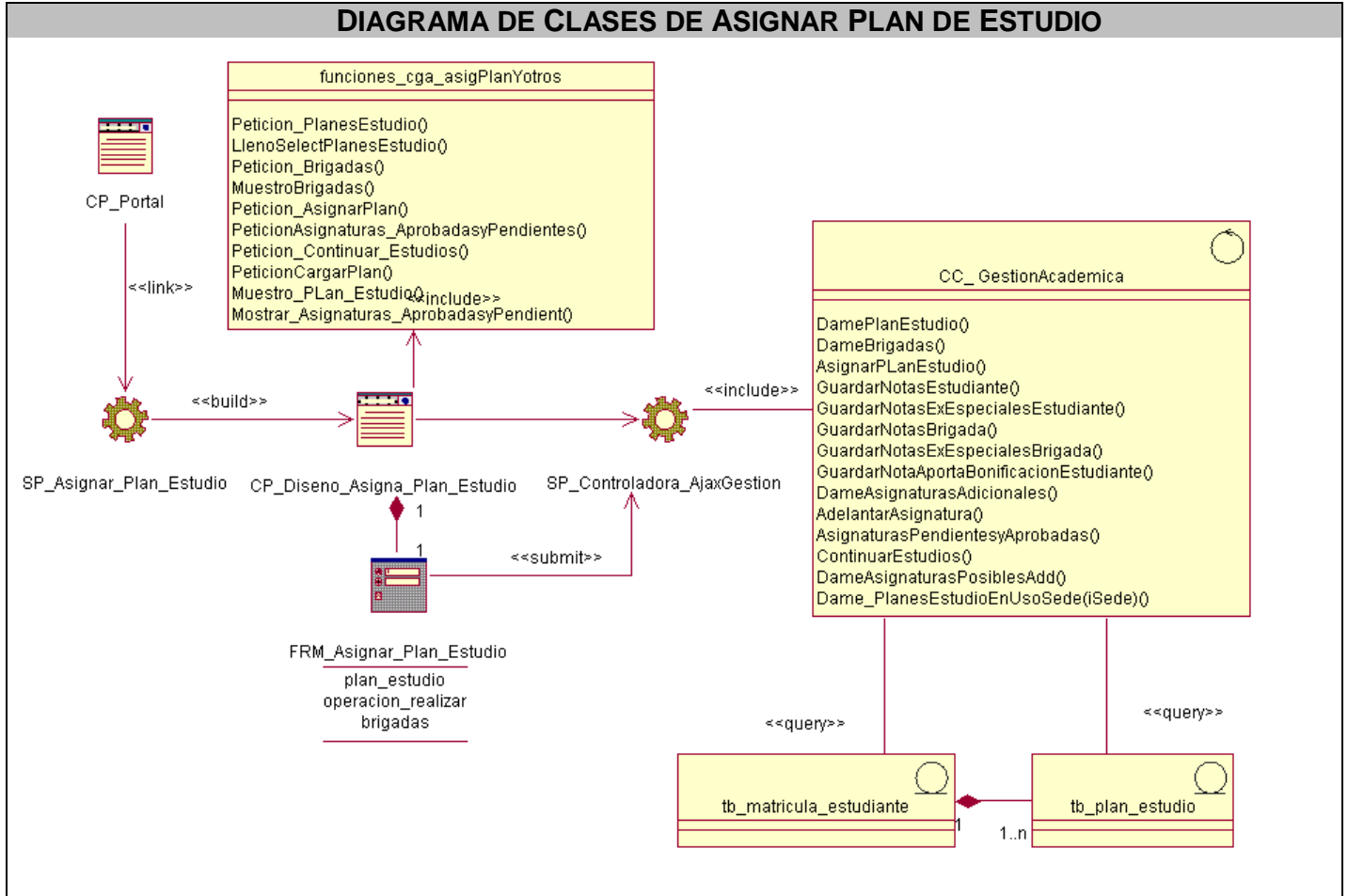


Fig. 3.10 Diagrama de clases del diseño del caso de uso Asignar Plan de Estudio.

DIAGRAMA DE CLASES DE ADELANTAR ASIGNATURA

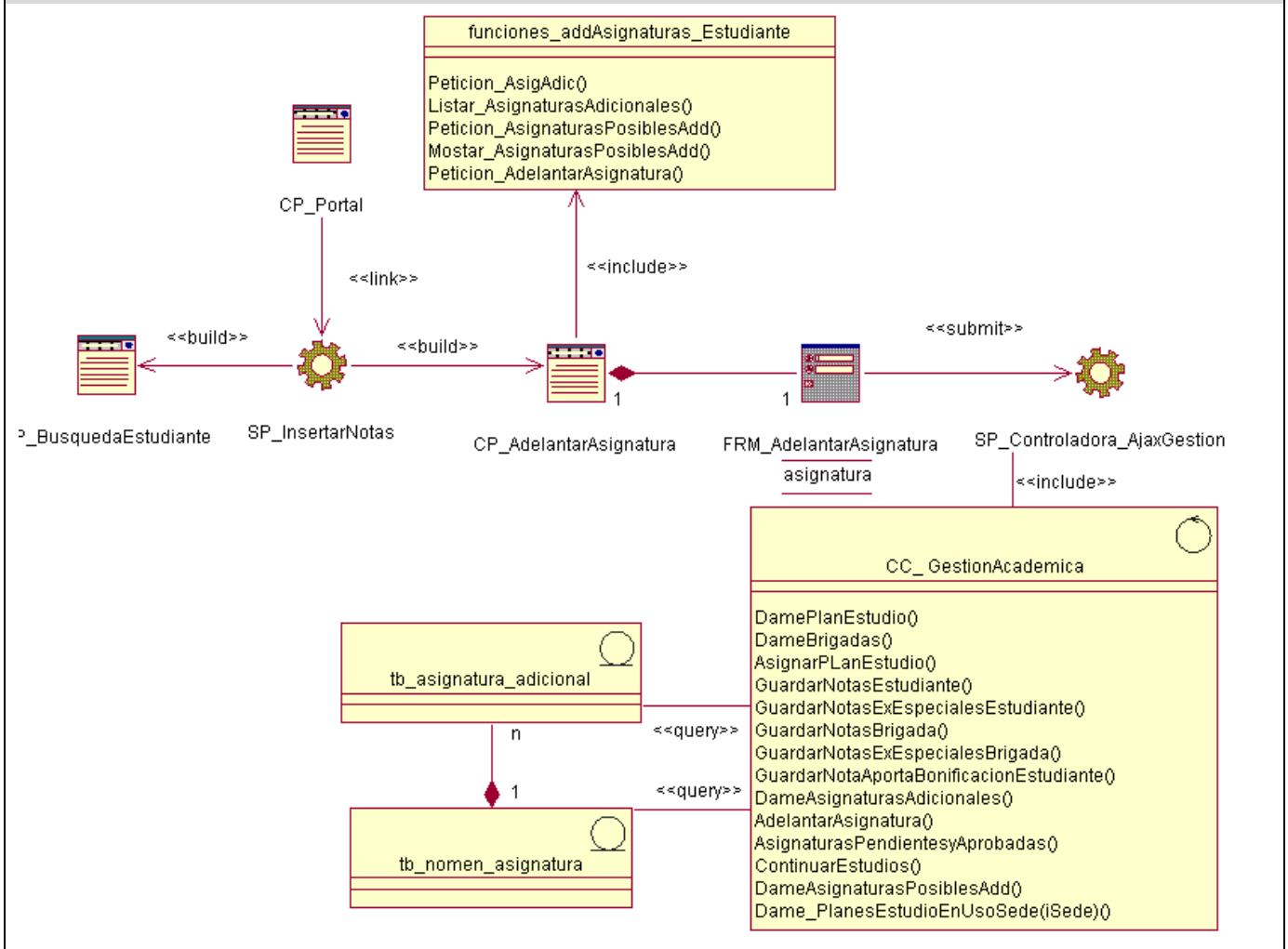


Fig: 3.11 Diagrama de clases del análisis del caso de uso Adelantar Asignatura.

DIAGRAMA DE CLASES DEL ANALISIS, CU GESTIONAR EVALUACIONES

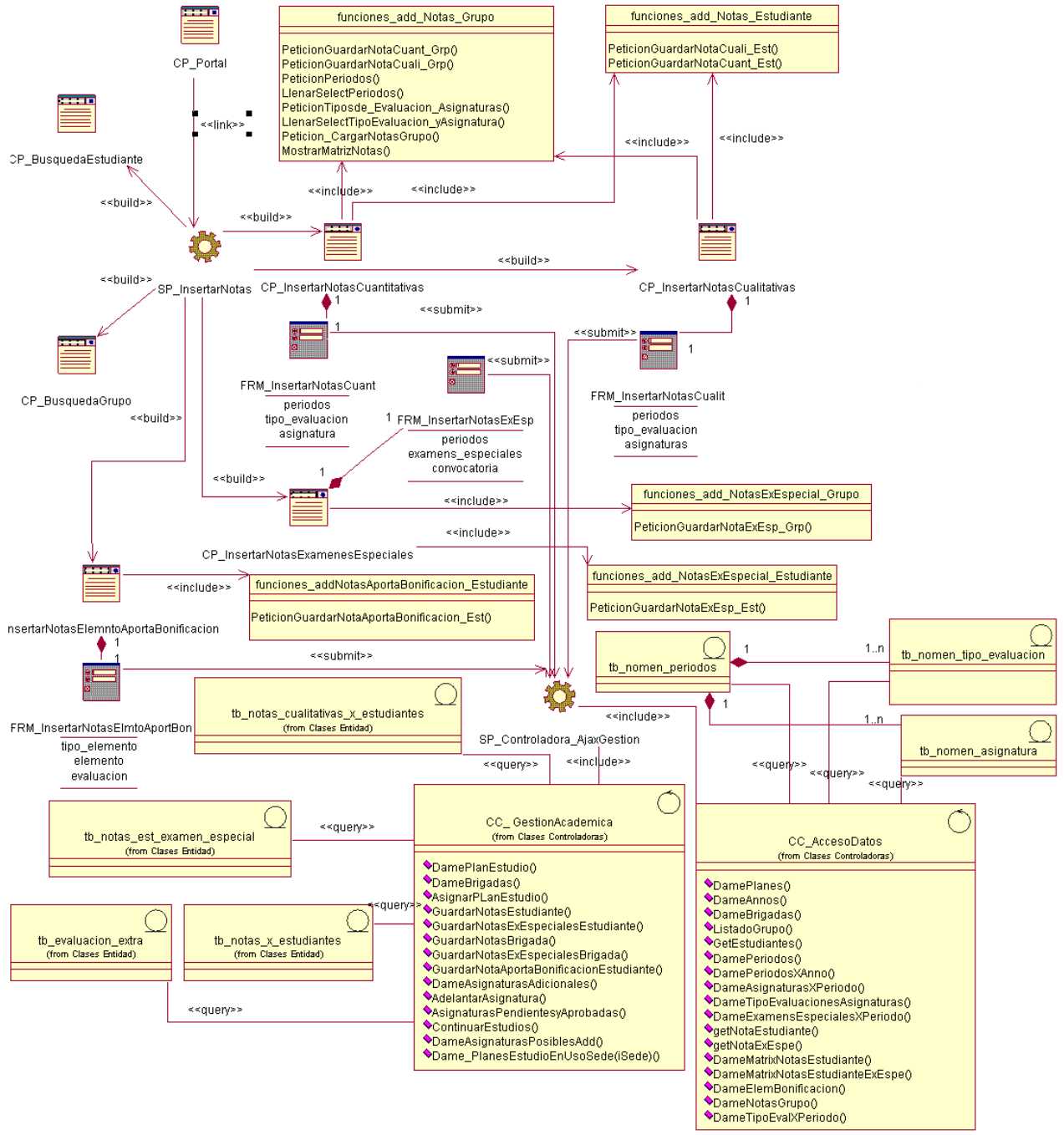


Fig: 3.12 Diagrama de clases del diseño del caso de uso Gestionar Evaluaciones.

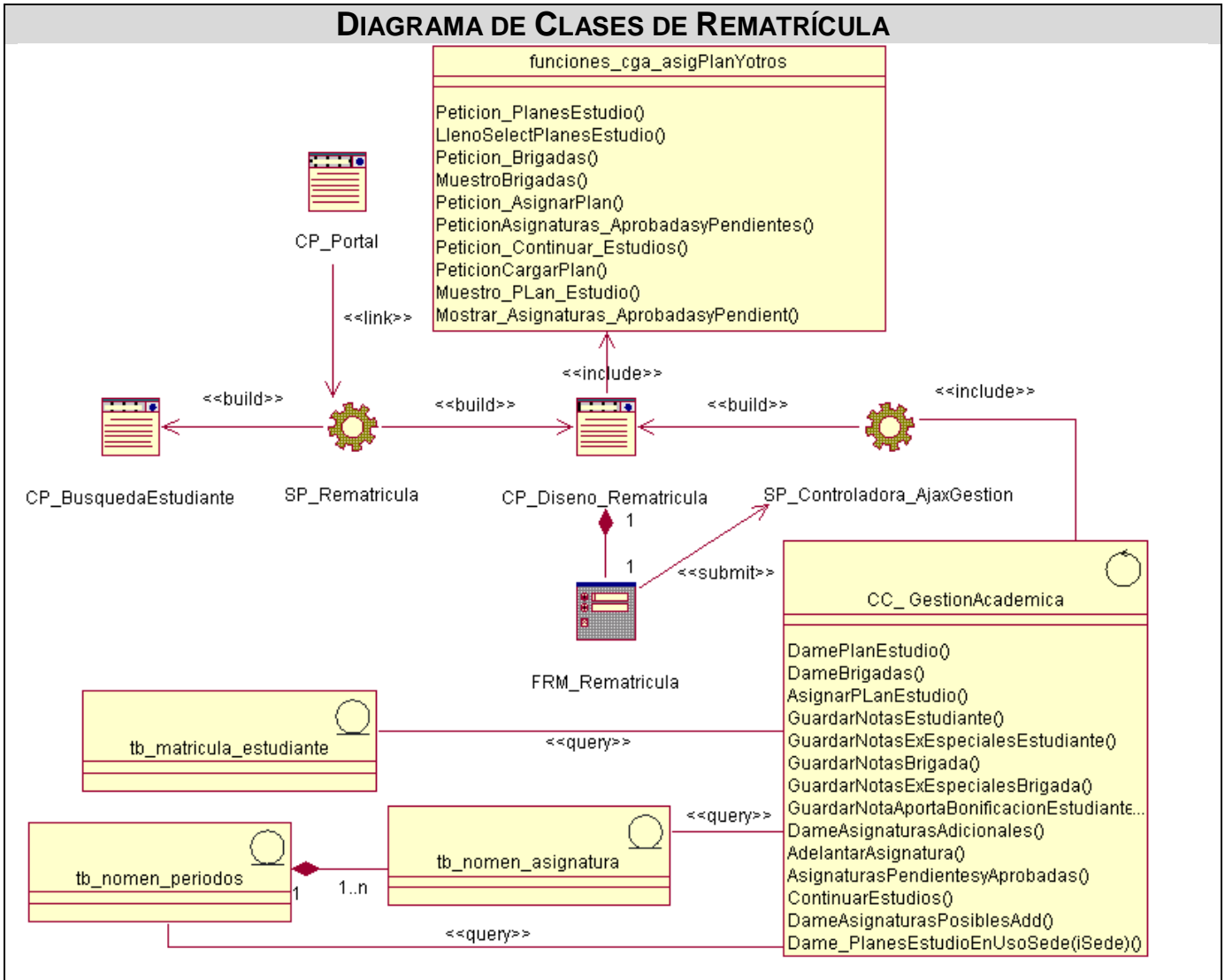


Fig: 3.13 Diagrama de clases del diseño del caso de uso Rematricula.

DIAGRAMA DE CLASES DE GESTIONAR REPORTES

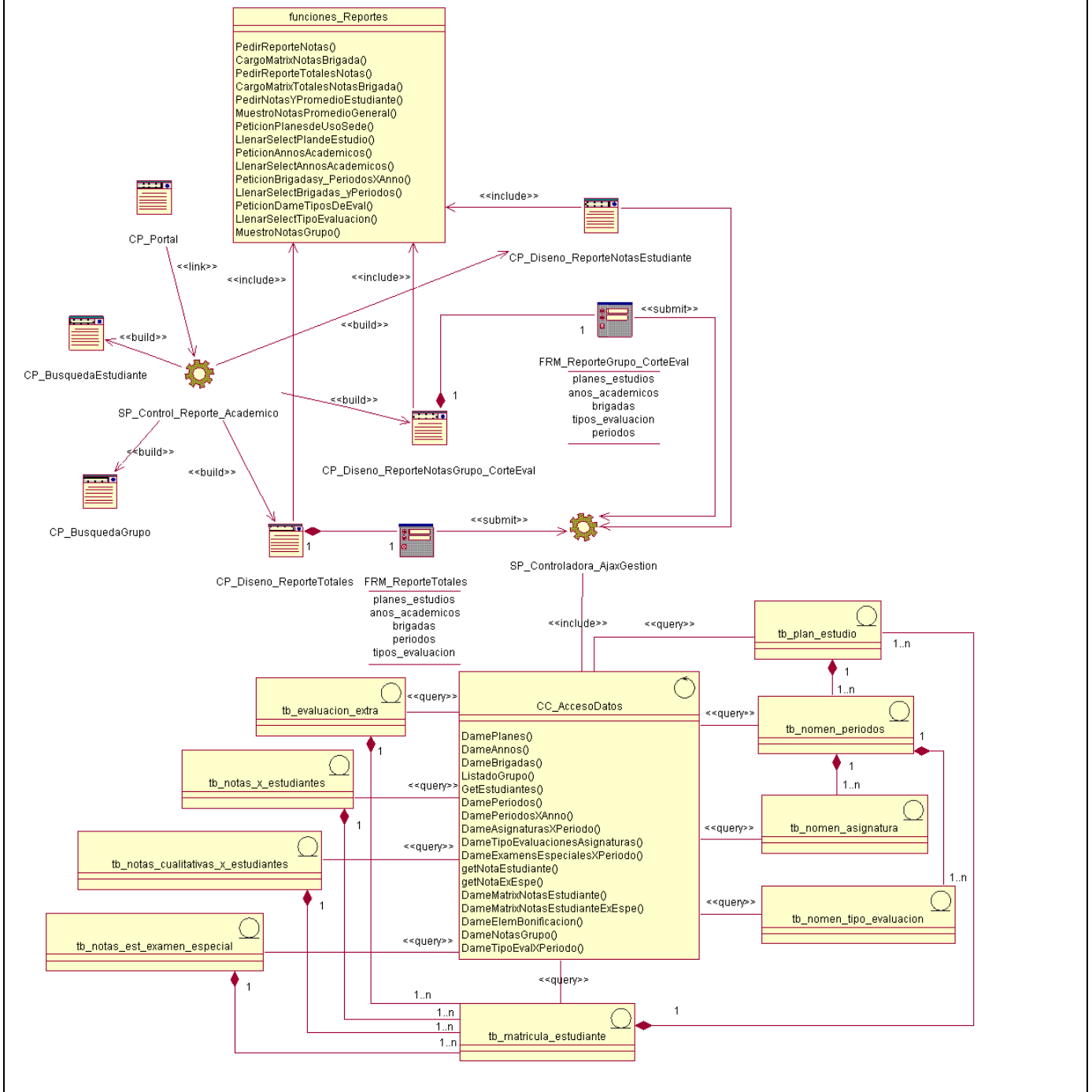


Fig: 3.14 Diagrama de clases del diseño del caso de uso Gestionar Reportes.

3.3 Descripción de las clases.

Nombre: tb_plan_estudio (CU-A1) (CU-G1) (CU-G5)	
Tipo de clase: Entidad.	
Atributo	Tipo
ildPlanEstudio	Integer
sNombrePlanEstudio	String
sDistribuido	String
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.1 Clase Entidad del Diseño, tb_plan_estudio.

Nombre: tb_elementos_aportan_bonificacion (CU- A2) (CU-G2)	
Tipo de clase: Entidad.	
Atributo	Tipo
ildElemento	Integer
sNombreElemento	String
sPromedia	String
sTipoEvaluacion	String
fBonificacionAlta	Float
fBonificacionMedia	Float
fBonificacionBaja	Float
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.2 Clase Entidad del Diseño, tb_elementos_aportan_bonificacion.

Nombre: tb_nomen_periodos (CU-A1) (CU-G2) (CU-G3) (CU-G5)	
Tipo de clase: Entidad.	
Atributo	Tipo
ildPeriodo	Integer
sDescripcion	String
sIdioma	String
iAnoAcademico	Integer
sPrecedenciaPeriodo	Integer
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.3 Clase Entidad del Diseño, tb_nomen_periodos.

Nombre: tb_nomen_tipo_evaluacion (CU-A1) (CU-G2) (CU-G5)	
Tipo de clase: Entidad.	
Atributo	Tipo
ildTipoEvaluacion	Integer
sNombreTipoEvaluacion	String
sDefinePeriodo	String
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.4 Clase Entidad del Diseño, tb_nomen_tipo_evaluacion.

Nombre: tb_nomen_asignatura (CU- A1) (CU-G1) (CU-G2) (CU-G5)	
Tipo de clase: Entidad.	
Atributo	Tipo
ildAsignatura	Integer
sNombreAsignatura	String
sNombreCorto	String

sPrecedenciaAsignatura	Integer
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.5 Clase Entidad del Diseño, tb_nomen_asignatura.

Nombre: tb_matricula_estudiante (CU-G1) (CU-G3) (CU-G5)	
Tipo de clase: Entidad.	
Atributo	Tipo
ildMatricula	Integer
ildEstudiante	String
ildEscuela	Integer
sNombreBrigada	String
ildBrigada	Integer
sTipoCurso	String
iAñoAcademico	Integer
sPropiedad	String
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.6 Clase Entidad del Diseño, tb_matricula_estudiante.

Nombre: tb_notas_x_estudiantes (CU-G2) (CU-G5)	
Tipo de clase: Entidad.	
Atributo	Tipo
ildMatricula	Integer
ildAsignatura	Integer
iTipoEvaluacion	Integer
iEvaluacion	Integer
ildUserIntrodujo	String

sNombreProfesor	String
dFecha	Date
sAlmacenado	String
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.7 Clase Entidad del Diseño, tb_notas_x_estudiantes.

Nombre: tb_notas_cualitativas_x_estudiantes (CU- G2) (CU-G5)	
Tipo de clase: Entidad.	
Atributo	Tipo
ildMatricula	Integer
ildAsignatura	Integer
iTipoEvaluacion	Integer
sEvaluacion	String
ildUserIntrodujo	String
sNombreProfesor	String
dFecha	Date
sAlmacenado	String
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.8 Clase Entidad del Diseño, tb_cualitativas_x_estudiantes.

Nombre: tb_notas_est_examen_especial (CU-G2) (CU-G5)	
Tipo de clase: Entidad.	
Atributo	Tipo
ildMatricula	Integer
ildExamenEspecial	Integer
iConvocatoria	Integer

iEvaluacion	Integer
ildUserIntrodujo	String
sNombreProfesor	String
dFecha	Date
sAlmacenado	String
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.9 Clase Entidad del Diseño, tb_notas_est_examen_especial.

Nombre: tb_evaluacion_extra (CU-G2) (CU-G5)	
Tipo de clase: Entidad.	
Atributo	Tipo
ildMatricula	Integer
ildElemento	Integer
iEvaluacion	Integer
ildUserIntrodujo	String
dFecha	Date
Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.10 Clase Entidad del Diseño, tb_evaluacion_extra.

Nombre: tb_asignatura_adicional (CU-G4)	
Tipo de clase: Entidad.	
Atributo	Tipo
ildAsignatura	Integer
ildMatricula	Integer

Responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.11 Clase Entidad del Diseño, tb_asignatura_adicional.

Nombre: CC_AdminAcademico (CU-A1) (CU-A2)	
Tipo de clase: Controladora.	
Atributo	Tipo
Responsabilidad:	
Nombre:	AdicionarNomenclador(sNomenclador, aValores, aCampos)
Descripción:	Se encarga de adicionar cualquier tipo de nomenclador que conforma un plan de estudio, para lo cual se pasa como parámetros la tabla en la que se va a adicionar, los campos de esta y los nuevos valores a agregar.
Nombre:	EditarNomenclador(sNomenclador,aValores,aCampos,ildSeleccionado)
Descripción:	Se encarga de editar cualquier tipo de nomenclador que conforma un plan de estudio, para lo cual se pasa como parámetros la tabla en la que se va a editar, los campos de esta, los valores a editar y el identificador del nomenclador seleccionado.
Nombre:	EliminarNomenclador (sNomenclador,sCampold,ildSeleccionado)
Descripción:	Se encarga de eliminar cualquier tipo de nomenclador que conforma un plan de estudio, para lo cual se pasa como parámetros, la tabla en la que se va a eliminar y el campo llave de esta y el identificador del nomenclador seleccionado.
Nombre:	ListarDatosNomenclador(sNomenclador)
Descripción:	Se encarga de listar todo el contenido de un nomenclador, para lo cual se pasa como parámetro el identificador del nomenclador seleccionado.

Tabla 3.12 Clase Controladora del Diseño, CC_Adminacademico.

Nombre: CC_GestionAcademica (CU-G1) (CU-G2) (CU-G3) (CU-G4) (CU-G5)	
Tipo de clase: Controladora.	
Atributo	Tipo

Responsabilidad:	
Nombre:	DamePlanEstudio(ildPlanSeleccionado)
Descripción:	Se encarga de devolver un plan de estudio con todos sus períodos y este a su vez con todas sus asignaturas y exámenes especiales para lo cual se pasa como parámetro el identificador del plan seleccionado.
Nombre:	DameBrigadas (ildEscuela, ildPlan)
Descripción:	Se encarga de devolver el identificador y nombre de brigadas para lo cual se pasa como parámetros identificador de la escuela a la que pertenecen y el identificador del plan de estudio que tienen asignado.
Nombre:	AsignarPLanEstudio(ildPlanSel , aldBrigadas)
Descripción:	Se encarga de asignar un plan de estudio a las brigadas para lo cual se pasa como parámetros el identificador de un plan y los identificadores de las brigadas.
Nombre:	GuardarNotasEstudiante(sTablaEvaluacion,iPeriodo,iTipoEvaluacion, iCodEstudiante,aMatriz)
Descripción:	Se encarga de guardar las notas cualitativas o cuantitativas de un estudiante para lo cual se pasa como parámetros la tabla según si la evaluación es cualitativa o cuantitativa, el identificador del período, el identificador del corte evaluativo, el carnet del estudiante y la matriz de asignaturas y notas respectivamente del estudiante.
Nombre:	GuardarNotasBrigada(sTablaEvaluacion,iPeriodo,iTpoEval,CarnetNotas, aAsigNotas, aNotas)
Descripción:	Se encarga de guardar las notas cualitativas o cuantitativas de una brigada para lo cual se pasa como parámetros la tabla según si la evaluación es cualitativa o cuantitativa, el identificador del período, el identificador del corte evaluativo, el arreglo de los carnets de los estudiantes, el arreglo de asignaturas y el arreglo de notas respectivamente de los estudiantes.
Nombre:	GuardarNotasExEspecialesBrigada(iPeriodo,iExamen,iConvocatoria, aCodEstudiante,aNotasBrigada)
Descripción:	Se encarga de guardar las notas de lo exámenes especiales de una brigada para lo cual se pasa como parámetros el identificador del período, el identificador del examen especial, la convocatoria, el arreglo de los carnets de los estudiantes y el arreglo de

	notas respectivamente de los estudiantes.
Nombre:	GuardarNotaAportaBonificacionEstudiante(sTipo,iElemento,sNota, iCodEstudiante)
Descripción:	Se encarga de adicionarle a un estudiante un elemento que aporta bonificación con su respectiva nota para lo cual se pasa como parámetros el tipo de elemento si es examen de premio o gestionado a nivel nacional, el identificador del elemento, la nota y el carnet del estudiante.
Nombre:	AdelantarAsignatura(iAsignatura, iCodEstudiante)
Descripción:	Se encarga de adicionarle a un estudiante una nueva asignatura para lo cual se pasa como parámetros el identificador de la nueva asignatura y el carnet del estudiante.
Nombre:	ContinuarEstudios(iCodEstudiante)
Descripción:	Se encarga de rematricular el estudiante en el período predecesor al que se encuentra para lo cual se pasa como parámetro el carnet del estudiante.

Tabla 3.13 Clase Controladora del Diseño, CC_GestionAcademica.

Nombre: CC_AccesoDatos (CU-G2) (CU-G5)	
Tipo de clase: Controladora.	
Atributo	Tipo
Responsabilidad:	
Nombre:	DamePlanes()
Descripción:	Se encarga de devolver el identificador y el nombre de los planes de estudio que se encuentran en uso en la sede.
Nombre:	DameAnnos (ildPlan)
Descripción:	Se encarga de devolver los años académicos por lo que esta compuesto un plan de estudio para lo cual se pasa como parámetro el identificador de un plan.
Nombre:	DameBrigadas(ildPlan , iAno)
Descripción:	Se encarga de devolver las brigadas que tienen asignado cierto plan de estudio para lo cual se pasa como parámetro el identificador de un plan y si lo desea el año académico.

Nombre:	ListadoGrupo(ildBrigada)
Descripción:	Se encarga de listar todos los estudiantes de una brigada para lo cual se pasa como parámetro el identificador de esta.
Nombre:	GetEstudiantes(sNombre,sSegundoNombre,sPrimerApellido,sSegundoApellido,cSexo)
Descripción:	Se encarga de devolver el carnet y el nombre de los estudiantes que cumplan con algunas características comunes para lo cual se pasa como parámetros el nombre, el segundo nombre, el primer apellido, el segundo apellido y el sexo.
Nombre:	DamePeriodos(iCodEstudiante , ildBrigada)
Descripción:	Se encarga de devolver el identificador y nombre de los periodos actuales y anteriores para un estudiante o un grupo para lo cual se pasa como parámetro el carnet de un estudiante o el identificador de una brigada.
Nombre:	DamePeriodosXAnno (iCodPlan,iAnno)
Descripción:	Se encarga de devolver el identificador y nombre de los períodos de un plan de estudio y año académico específico para lo cual se pasa como parámetros el identificador de un plan de estudio y el año académico.
Nombre:	DameTipoEvaluacionXPeriodo(ilPeriodo)
Descripción:	Se encarga de devolver los tipo de evaluación de un período para lo cual se pasa como parámetro el identificador de un período.
Nombre:	DameAsignaturasXPeriodo(ilPeriodo)
Descripción:	Se encarga de devolver las asignaturas de un período para lo cual se pasa como parámetro el identificador de un período.
Nombre:	DameExamenesEspecialesXPeriodo(ilPeriodo)
Descripción:	Se encarga de devolver los exámenes especiales de un período para lo cual se pasa como parámetro el identificador de un período.
Nombre:	DameMatrixNotasEstudiantes(sTablaEvaluacion, iPeriodo, iTipoEvaluacion, iCodEstudiante)
Descripción:	Se encarga de devolver las evaluaciones cualitativas o cuantitativas, el nombre y carnet de un estudiante para lo cual se pasa como parámetros el tipo de tabla según si es cualitativa o cuantitativa, el identificador de período, el identificador del corte

	evaluativo, y el carnet del estudiante.
Nombre:	DameMatrixNotasEstudiantesExamenEspecial(iPeriodo,ildExamen, sNombreExamen, iConvocatoria, iCodEstudiante)
Descripción:	Se encarga de devolver las evaluaciones de los exámenes especiales, el nombre y carnet de un estudiante para lo cual se pasa como parámetros el identificador de período, el identificador del examen especial, la convocatoria, y el carnet del estudiante.
Nombre:	DameElementosBonificacion()
Descripción:	Se encarga de devolver el identificador y el nombre de los elementos que aportan bonificación gestionados a nivel nacional.
Nombre:	DameNotasGrupo(sTablaEvaluacion, iPeriodo, iTipoEvaluacion, ildBrigada)
Descripción:	Se encarga de devolver las evaluaciones cualitativas o cuantitativas, el nombre y carnet de los estudiantes para lo cual se pasa como parámetro el tipo de tabla según si es cualitativa o cuantitativa, el identificador de período, el identificador del corte evaluativo, y el identificador de la brigada.

Tabla 3.14 Clase Controladora del Diseño, CC_AccesoDatos.

Nombre: CP_Diseño_GestionarPlanEstudio (CU- A1)	
Tipo de clase: Interfaz.	
Atributo	Tipo
nomenclador Estudio	String
nomenclador Período	String
nomenclador Asignatura	String
nomenclador Plan de Estudio	String
nomenclador Examen Especial	String
nomenclador Tipo de Evaluacion	String
nomenclador Relación Estudio Período	String
nomenclador Relación Período Asignatura	String
nomenclador Relación Plan	String

Estudio Período	
nomenclador Relación Período Evaluaciones	String
nomenclador Relación Período Examen Especial	String
Responsabilidad:	
Nombre:	
Descripción:	A partir de esta clase interfaz se selecciona el nomenclador que se desee para que posteriormente se adicione uno nuevo, se edite o se elimine uno existente.

Tabla 3.15 Clase Interfaz del Diseño, CP_Diseño_GestionarPlanEstudio.

Nombre: CP_GestionNomenPlanEstudio (CU- A1)	
Tipo de clase: Interfaz.	
Atributo	Tipo
nombre	String
distribuido	Boolean
Responsabilidad:	
Nombre:	Peticion_ AdicionarNomenclador(nomen, valores, campos)
Descripción:	Se encarga de realizar la petición a la clase controladora para adicionar un nuevo nomenclador, en este caso un Plan de Estudio.
Nombre:	Peticion_ EditarNomenclador()
Descripción:	Se encarga de realizar la petición a la clase controladora para modificar los datos de un nomenclador, en este caso un Plan de Estudio.
Nombre:	Peticion_ EliminarNomenclador()
Descripción:	Se encarga de realizar la petición a la clase controladora para eliminar un nomenclador, en este caso un Plan de Estudio seleccionado.
Nombre:	ListarDatosNomenclador()
Descripción:	Se encarga de mostrar en la interfaz una lista de todos los nomencladores, en este caso de todos los planes de estudio existentes distribuidos y no distribuidos.

Tabla 3.16 Clase Interfaz del Diseño, CP_GestionNomenPlanEstudio.

Nombre: CP_GestionarElemAportanBonif (CU- A2)	
Tipo de clase: Interfaz.	
Atributo	Tipo
nombre	String
promedia	Boolean
tipo de evaluacion	String
aporta	Float
Responsabilidad:	
Nombre:	
Descripción:	Utiliza los métodos descritos en la clase interfaz anterior para insertar, editar, eliminar y mostrar los datos de los elementos que aportan bonificación.

Tabla 3.17 Clase Interfaz del Diseño, CP_GestionarElemAportanBonif.

Nombre: CP_Diseno_Asigna_Plan_Estudio (CU- G1)	
Tipo de clase: Interfaz.	
Atributo	Tipo
plan_estudio	String
operacion_realizar	String
brigadas	String
Responsabilidad:	
Nombre:	Peticion_PlanesEstudio(sede)
Descripción:	Se encarga de realizar la petición a la clase controladora para la búsqueda de los planes de estudio pertenecientes a la sede universitaria.
Nombre:	LlenoSelectPlanesEstudio()
Descripción:	Se encarga de llenar una lista desplegable con los planes de estudio obtenidos.
Nombre:	Peticion_Brigadas(sede, select_plan)
Descripción:	Se encarga realizar la petición a la clase controladora para la búsqueda de las brigadas de la sede en dependencia de la operación a realizar. En caso de que se seleccione Asignar plan, se realiza a petición para buscar las brigadas que no posean un plan de estudio asignado; en caso de seleccionarse Actualizar plan, se

	realiza a petición para buscar las brigadas que tengan asignado el plan de estudio seleccionado con anterioridad.
Nombre:	MuestroBrigadas()
Descripción:	Se encarga de mostrar las brigadas resultantes de la búsqueda.
Nombre:	Peticion_AsignarPlan(select_plan, select_brigadas)
Descripción:	Se encarga de realizar la petición a la clase controladora para asignar un plan de estudio seleccionado a una (o varias) brigada(s).

Tabla 3.18 Clase Interfaz del Diseño, CP_Disenio_Asigna_Plan_Estudio.

Nombre: CP_AdelantarAsignatura (CU-G4)	
Tipo de clase: Interfaz.	
Atributo	Tipo
asignatura	String
Responsabilidad:	
Nombre:	Peticion_AsigAdic(estd)
Descripción:	Se encarga de realizar la petición a la clase controladora para obtener las asignaturas adicionales de un estudiante buscado con anterioridad.
Nombre:	Listar_AsignaturasAdicionales()
Descripción:	Se encarga de listar las asignaturas adicionales pertenecientes a un estudiante buscado.
Nombre:	Peticion_AsignaturasPosiblesAdd(estd)
Descripción:	Se encarga de realizar la petición a la clase controladora para obtener las asignaturas posibles a adelantar.
Nombre:	Mostar_AsignaturasPosiblesAdd()
Descripción:	Se encarga mostrar en pantalla las asignaturas posibles a adelantar.
Nombre:	Peticion_AdelantarAsignatura(estd, select_asig)
Descripción:	Se encarga de realizar la petición a la clase controladora para adelantar una asignatura.

Tabla 3.19 Clase Interfaz del Diseño, CP_AdelantarAsignatura.

Nombre: CP_InsertarNotasCuantitativas (CU-G2)	
Tipo de clase: Interfaz.	
Atributo	Tipo
periodos	String
tipo_evaluacion	String
asignatura	String
Responsabilidad:	
Nombre:	Peticion_CargarNotasGrupo(brigada)
Descripción:	Se encarga de realizar la petición a la clase controladora para obtener la matriz de notas de un grupo especificado, según el período, tipo de evaluación y asignatura.
Nombre:	MostrarMatrizNotas()
Descripción:	Se encarga de mostrar la matriz de notas de una brigada.
Nombre:	PeticionGuardarNotaCuant_Grp(brigada, matriz_eval)
Descripción:	Se encarga de realizar la petición a la clase controladora para guardar una(o varias) evaluación(es) cuantitativas pertenecientes a una brigada.
Nombre:	PeticionGuardarNotaCuant_Est(estd,select_periodo,select_tipo_eval, eval, select_asig)
Descripción:	Se encarga de realizar la petición a la clase controladora para guardar una(o varias) evaluación(es) cuantitativas pertenecientes a un estudiante buscado con anterioridad según el período, tipo de evaluación y asignatura seleccionada.

Tabla 3.20 Clase Interfaz del Diseño, CP_InsertarNotasCuantitativas.

Nombre: CP_Disenio_Rematricula (CU-G3)	
Tipo de clase: Interfaz.	
Atributo	Tipo
Responsabilidad:	
Nombre:	PeticionAsignaturas_AprobadasyPendientes(estd)
Descripción:	Se encarga de realizar la petición a la clase controladora para obtener las asignaturas aprobadas y pendientes de un estudiante buscado.
Nombre:	Mostrar_Asignaturas_AprobadasyPendient(aprobadas, pendientes)

Descripción:	Se encarga de mostrar las asignaturas aprobadas y pendientes de un estudiante.
Nombre:	Peticion_Continuar_Estudios()
Descripción:	Se encarga de realizar la petición a la clase controladora para realizar la matricula del estudiante en el siguiente año académico.

Tabla 3.21 Clase Interfaz del Diseño, CP_Diseno_Rematriculada.

Nombre: CP_Diseno_ReporteNotasGrupo_CorteEval (CU-G5)	
Tipo de clase: Interfaz.	
Atributo	Tipo
planes_estudios	String
anos_academicos	String
brigadas	String
tipos_evaluacion	String
periodos	String
Responsabilidad:	
Nombre:	Pedicion_ReporteNotas(select_brigadas, select_tipos_evaluacion, select_periodos)
Descripción:	Se encarga de realizar la petición a la clase controladora para obtener una matriz de notas según planes de estudio, años académicos, brigadas, tipos de evaluación y periodos seleccionados.
Nombre:	CargoMatrixNotasBrigada(sTablaEvaluacion)
Descripción:	Se encarga de mostrar la matriz de notas de una brigada.

Tabla 3.22 Clase Interfaz del Diseño, CP_Diseno_ReporteNotasGrupo_CorteEval.

3.3.1 Diagramas de Interacción

La esencia de la fase de diseño es la elaboración de los diagramas de interacción, estos muestran gráficamente cómo los objetos se comunicarán entre ellos en fin de cumplir con los requerimientos. Los diagramas de interacción, pueden ser de dos tipos: secuencia y colaboración. En este trabajo, se representan a través de diagramas de secuencia, cómo se realiza el flujo organizado de los eventos en cada caso de uso definido para el sistema, referirse al **Anexo 3**.

3.4 Diseño de la base de datos.

En el diseño de la base de datos se modela el tratamiento de la información con carácter persistente dentro del sistema. Varios son los métodos para modelar la persistencia de los datos, incluyendo una gran variedad de herramientas de modelado. La propuesta actual es modelar la persistencia de los datos a partir de los diagramas de clase, con herramientas modernas que realizan una traducción del modelo de clases a un modelo de datos relacional. A continuación se presenta el Modelo Entidad Relación.

MODELO ENTIDAD RELACION DE LA BASE DE DATOS

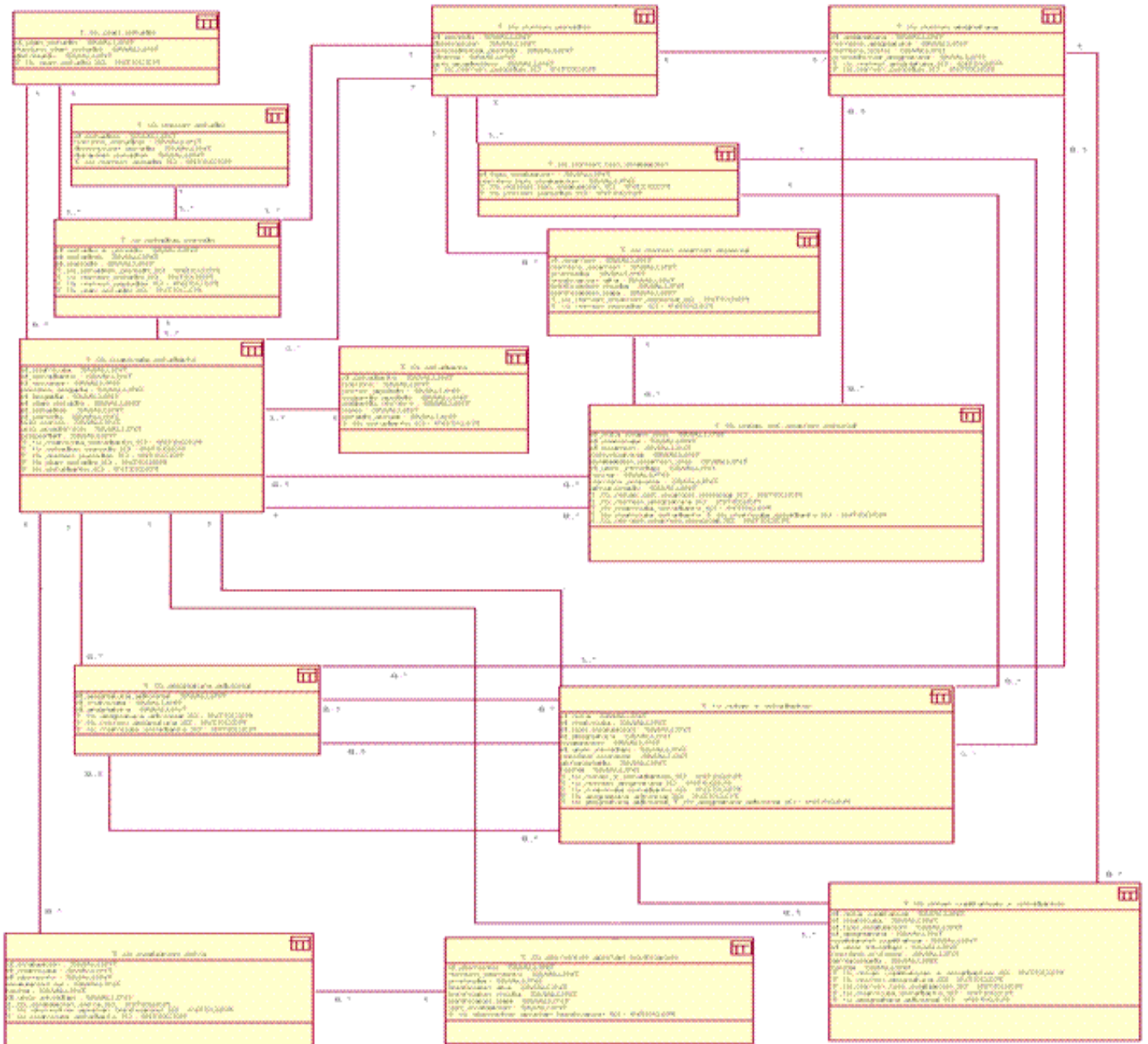


Fig 3.15 Modelo entidad relación de la base de datos.

3.4.1 Descripción de las tablas.

Nombre: tb_matrícula_estudiante		
Descripción: Esta tabla contiene la información referente la matrícula de un estudiante.		
Atributo	Tipo	Descripción
id_matricula	Integer	Representa el código de la matrícula del estudiante.
id_plan_estudio	Integer	Representa el identificador de plan de estudio.
id_estudios	Integer	Representa el identificador de un estudio.
id_periodo	Integer	Representa el identificador de un período.
id_estudiante	Varchar	Representa el identificador de un estudiante.
tipo_curso	Varchar	Representa el tipo de curso en el que va a estar matriculado un estudiante ej: diurno, nocturno.
ano_academico	Integer	Representa el año académico que va a estar cursando un estudiante.
id_cod_brigada	Integer	Representa el identificador de la brigada a la que pertenece.
propiedad	Varchar	Define si el estudiante es “recibido” de la base de datos del sistema o si se realizó alguna actualización (actualizado) por el componente.
id_escuela	Integer	Representa el identificador del centro docente donde se encuentra matriculado el estudiante.
nombre_brigada	Varchar	Representa el nombre de la brigada a la que pertenece el estudiante matriculado.

Tabla 3.23 Tabla de la base de datos, tb_matrícula_estudiante

Nombre: tb_plan_estudio		
Descripción: Esta tabla define solo el nombre y código de un plan de estudio.		

Atributo	Tipo	Descripción
id_plan_estudio	Integer	Representa el identificador del plan de estudio.
nombre_plan_estudio	Varchar	Representa el nombre del plan de estudio.
distribuido	Varchar	Representa si el plan de estudio se encuentra distribuido por las distintas sedes o no.

Tabla 3.24 Tabla de la base de datos, tb_plan_estudio.

Nombre: tb_estudiante		
Descripción: Esta tabla define solo el nombre y código de un plan de estudio.		
Atributo	Tipo	Descripción
id_estudiante	Integer Varchar	Representa el identificador del estudiante.
nombre	Varchar	Representa el nombre del estudiante.
segundo_nombre	Varchar	Representa el segundo nombre del estudiante.
primer_apellido	Varchar	Representa el primer apellido de un estudiante.
segundo_apellido	Varchar	Representa el segundo apellido de un estudiante.
sexo	Char	Representa sexo de un estudiante.
estado_actual	Varchar	Representa el estado actual de un estudiante, ya sea baja, traslado o licencia.

Tabla 3.25 Tabla de la base de datos, tb_estudiante.

Nombre: tb_nomen_estudio		
Descripción: Esta tabla contiene la información referente al nomenclador Estudio.		
Atributo	Tipo	Descripción
id_estudios	Integer	Representa el identificador del estudio.
nombre_estudios	Varchar	Representa el nombre del estudio.
descripción_estudios	Varchar	Representa la descripción del estudio
duración_estudios	Varchar	Representa la duración del estudio en años.

Tabla 3.26 Tabla de la base de datos, tb_nomen_estudio.

Nombre: tb_nomen_periodos		
Descripción: Esta tabla contiene la información referente al nomenclador Período.		
Atributo	Tipo	Descripción
id_periodo	Integer	Representa el identificador del período.
descripción	Varchar	Representa la descripción del período
idioma	Varchar	Representa el idioma en que se van a impartir las asignaturas del período.
ano_academico	Integer	Representa el año académico al que pertenece un período.
precedencia_periodo	Integer	Representa el período que precede al actual.

Tabla 3.27 Tabla de la base de datos, tb_nomen_periodos.

Nombre: tb_nomen_asignatura		
Descripción: Esta tabla contiene la información referente al nomenclador Asignatura.		
Atributo	Tipo	Descripción
id_asignatura	Integer	Representa el identificador de la asignatura.
nombre_asignatura	Varchar	Representa el nombre de la asignatura.
nombre_corto	Varchar	Representa una abreviatura del nombre de la asignatura.
precedencia_asignatura	Integer	Representa la asignatura que precede a la actual en caso de que exista.

Tabla 3.28 Tabla de la base de datos, tb_nomen_asignatura.

Nombre: tb_nomen_tipo_evaluacion		
Descripción: Esta tabla contiene la información referente al nomenclador Tipo de evaluación.		
Atributo	Tipo	Descripción
id_tipo_evaluacion	Integer	Representa el identificador del tipo de evaluación.
nombre_tipo_evaluacion	Varchar	Representa el nombre del tipo de evaluación.

define_periodo	Varchar	Representa el período al que pertenece el tipo de evaluación.
----------------	---------	---

Tabla 3.29 Tabla de la base de datos, tb_nomen_tipo_evaluacion.

Nombre: tb_notas_cualitativas_x_estudiante		
Descripción: Esta tabla contiene la información referente a las notas cualitativas de un estudiante.		
Atributo	Tipo	Descripción
id_notas_cualitativas	Integer	Representa el identificador de una nota cualitativa.
id_matricula	Integer	Representa el identificador de la matrícula.
id_asignatura	Integer	Representa el identificador de la asignatura.
id_tipo_evaluacion	Integer	Representa el identificador del tipo de evaluación.
evaluación	Varchar	Representa una evaluación cualitativa.
Id_user_introdujo	Varchar	Representa el identificador de la persona que introdujo la evaluación
nombre_profesor	Varchar	Representa el nombre del profesor que introdujo la evaluación
fecha	Date	Representa en que se introduce una evaluación.
almacenado	Varchar	Representa una descripción de donde está archivada la evaluación en formato duro.

Tabla 3.30 Tabla de la base de datos, tb_notas_cualitativas_x_estudiante

Nombre: tb_notas_x_estudiante		
Descripción: Esta tabla contiene la información referente a las notas de un estudiante.		
Atributo	Tipo	Descripción
id_notas	Integer	Representa el identificador de una nota.
id_matricula	Integer	Representa el identificador de la matrícula.

id_asignatura	Integer	Representa el identificador de la asignatura.
id_tipo_evaluacion	Integer	Representa el identificador del tipo de evaluación.
evaluación	Float	Representa una evaluación.
Id_user_introdujo	Varchar	Representa el identificador de la persona que introdujo la evaluación
nombre_profesor	Varchar	Representa el nombre del profesor que introdujo la evaluación
almacenado	Varchar	Representa una descripción de donde es que se va a almacenar la evaluación en formato duro.
fecha	Date	Representa la fecha en que se introdujo la evaluación.

Tabla 3.31 Tabla de la base de datos, tb_notas_x_estudiante

Conclusiones:

En este capítulo se realizó el análisis y diseño del Componente de Gestión Académica. Se modeló el sistema para dar soporte a los requisitos definidos tanto funcionales como no funcionales. En la realización del modelo de diseño se tuvo en cuenta la conservación de la estructura del modelo de análisis. Este constituye un punto de partida para las actividades de implementación subsiguientes.

CAPÍTULO 4: IMPLEMENTACIÓN

En la implementación se comienza con el resultado del diseño y se implementa el sistema en términos de componentes. El propósito general consiste en desarrollar la arquitectura y el sistema como un todo implementando las clases encontradas durante el diseño. Además se distribuye la aplicación físicamente en nodos en el diagrama de despliegue.

4.1 Diagrama de componentes

En la fase de implementación, se definen los componentes con los que va a contar el sistema. Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño. Existen diferentes tipos de estereotipos estándares, algunos de ellos corresponden a: <<executable>>, <<file>>, <<library>> y <<table>>. Estos componentes tienen relaciones de trazas con los elementos del modelo de implementación. A continuación se presentan un diagrama general de los componentes de la aplicación y posteriormente se dividió en subsistemas de implementación para lograr comprender mejor el sistema.

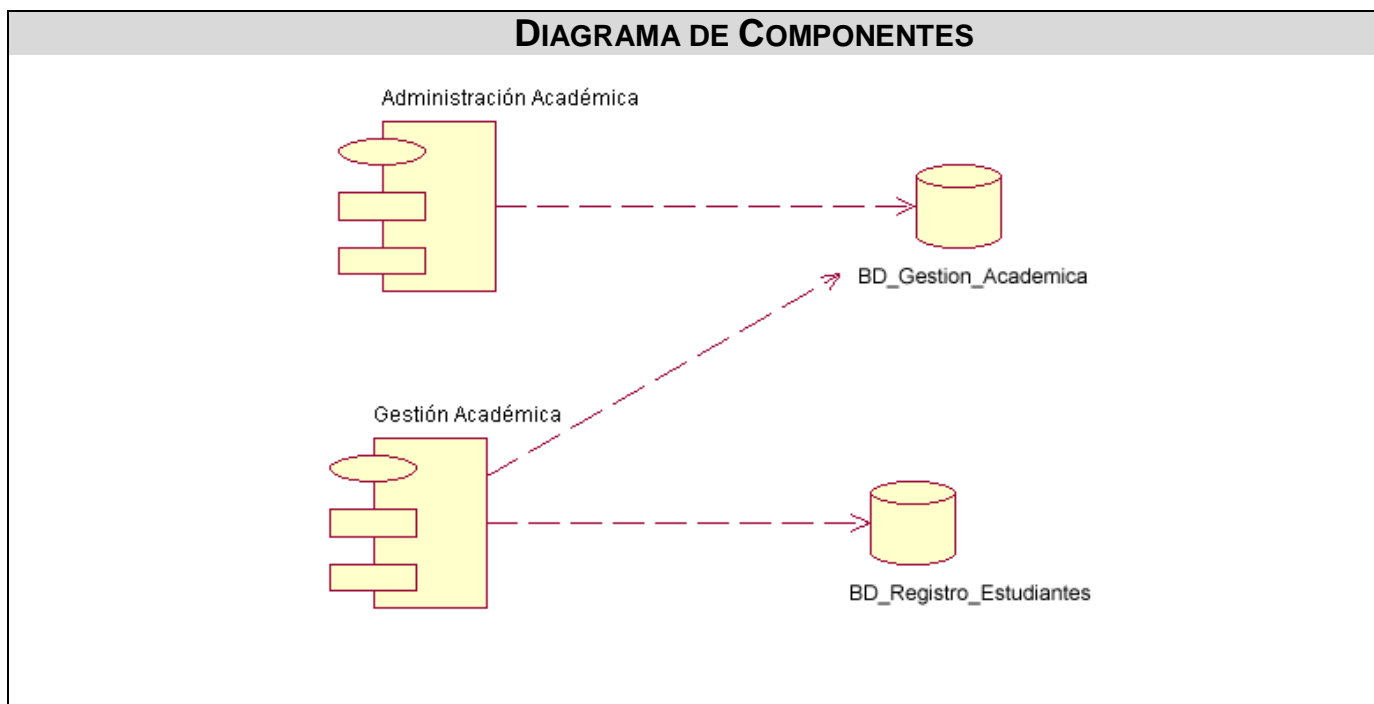


Fig 4.1 Diagrama de Componentes

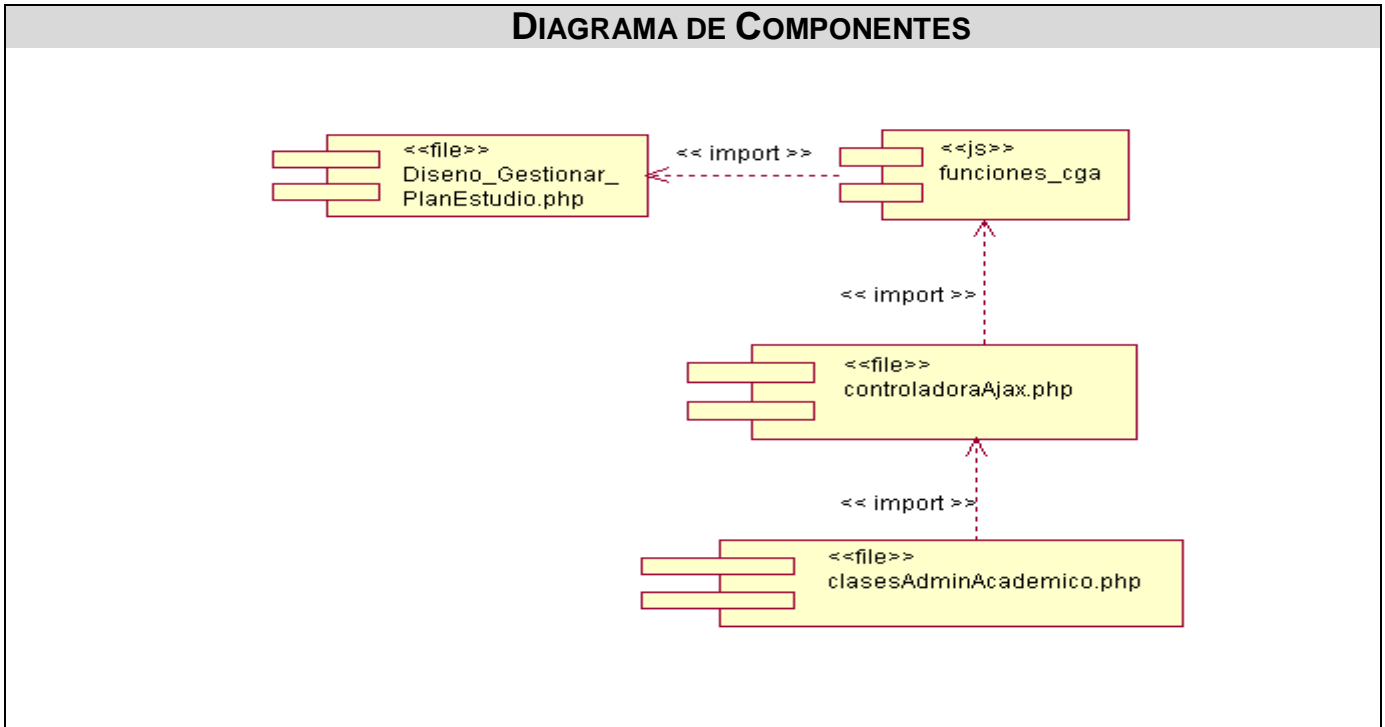


Fig 4.2 Diagrama de componentes Administración Académica, caso de uso Gestionar Plan de Estudio.

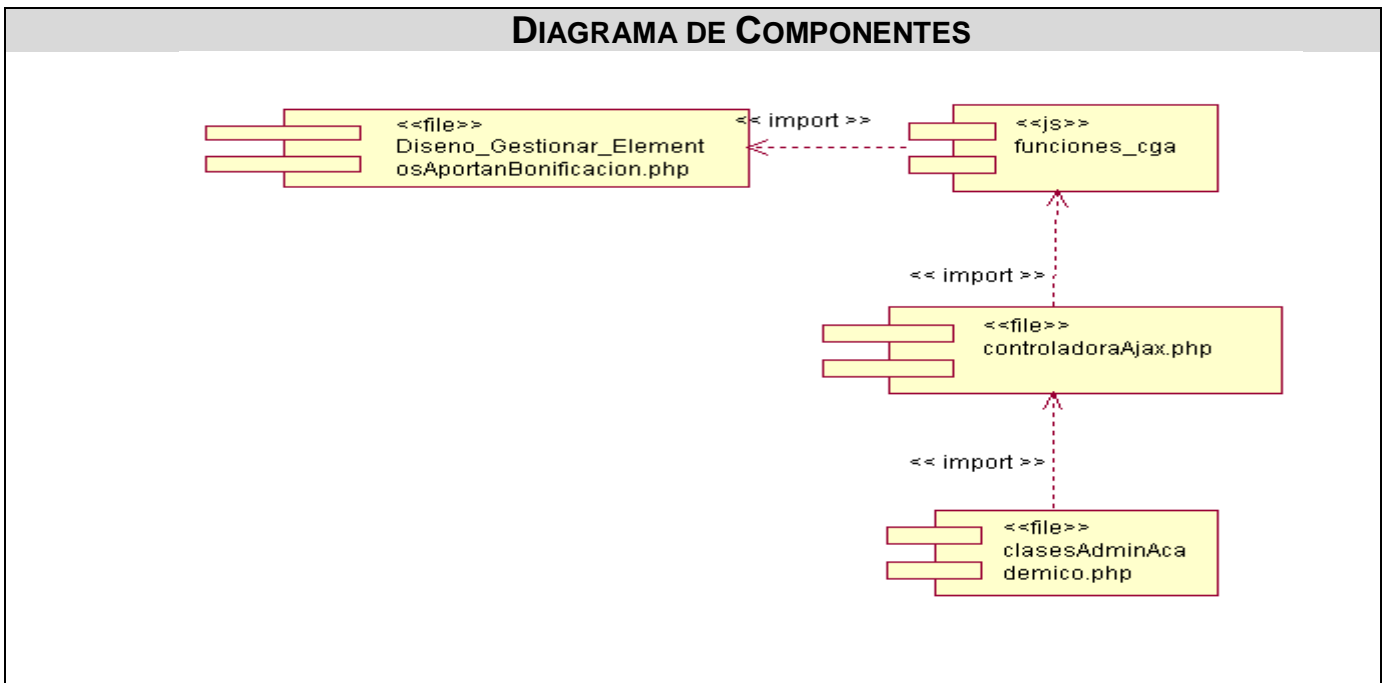


Fig 4.3 Diagrama de componentes Administración Académica, caso de uso Gestionar Elementos que Aportan Bonificación.

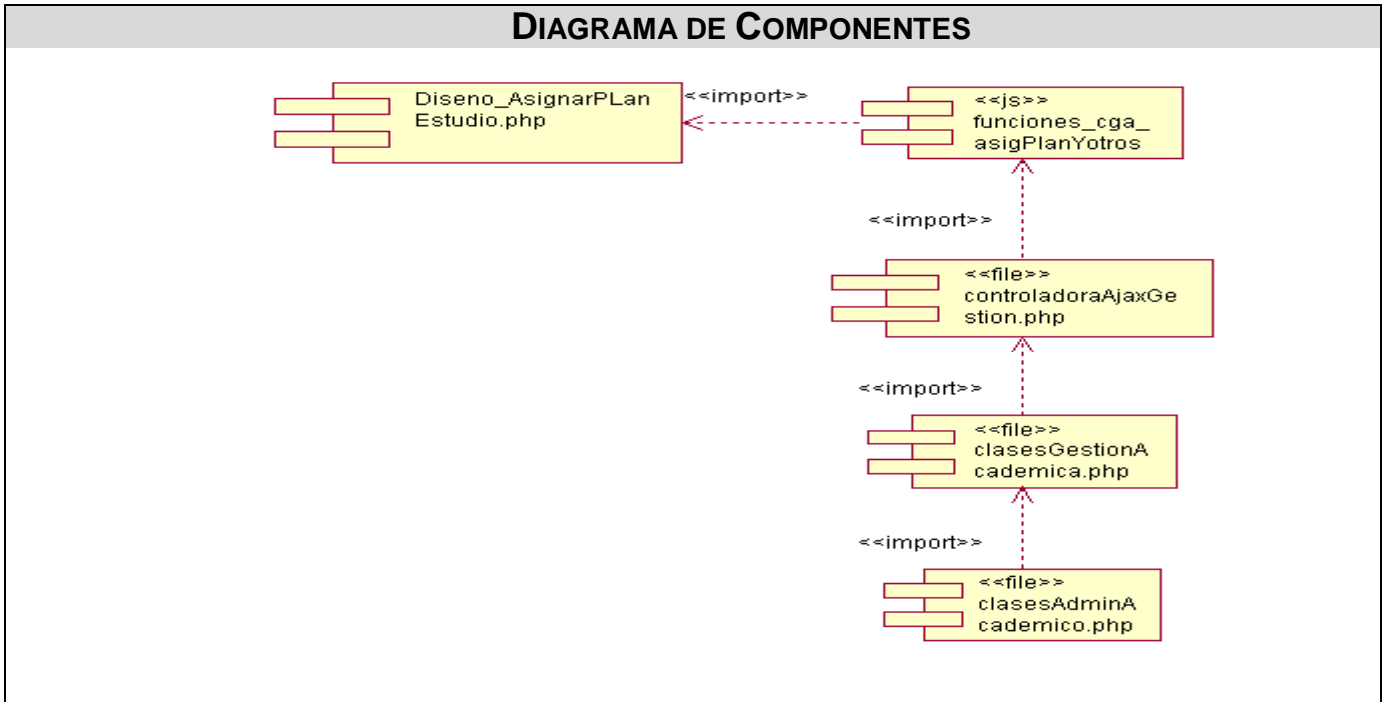


Fig 4.4 Diagrama de componentes Gestión Académica, caso de uso Asignar Plan de Estudio.

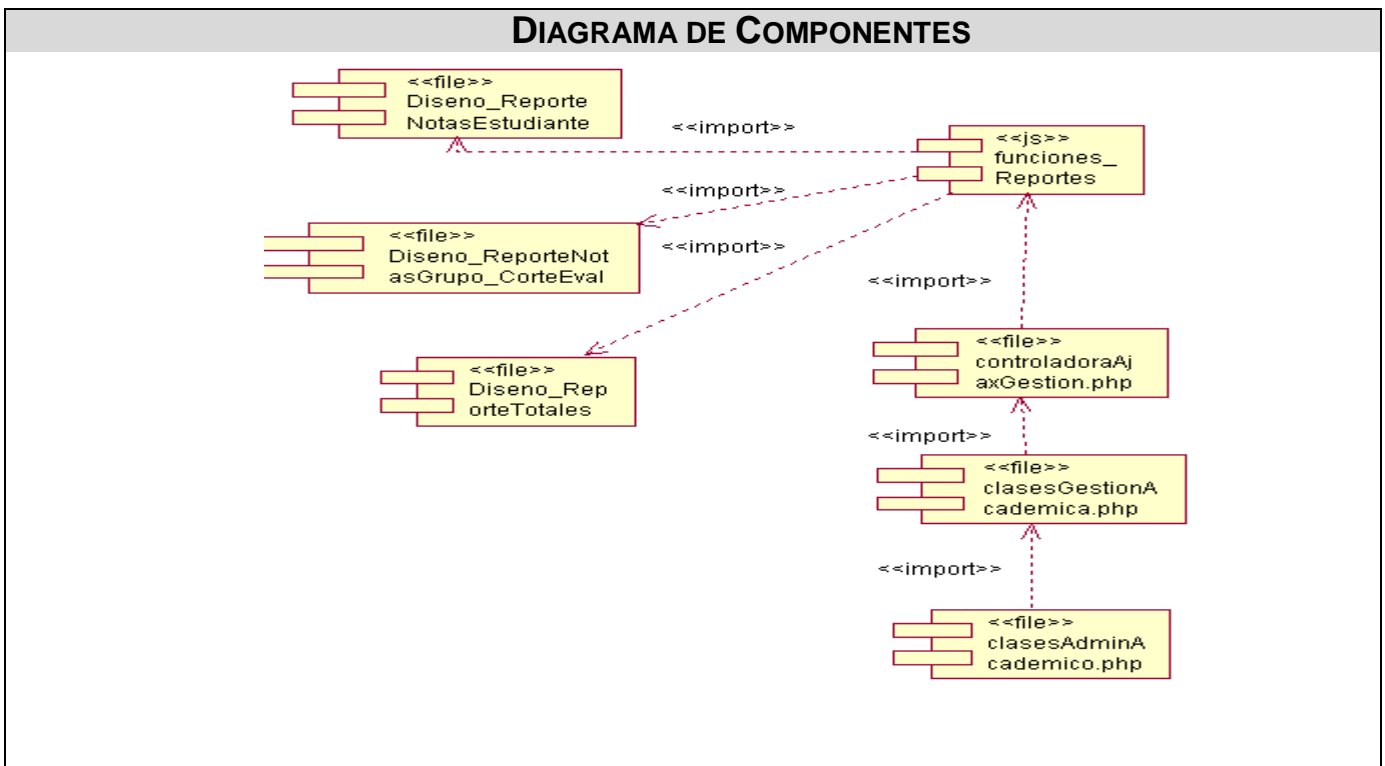


Fig 4.5 Diagrama de componentes Gestión Académica, caso de uso Gestionar Reportes

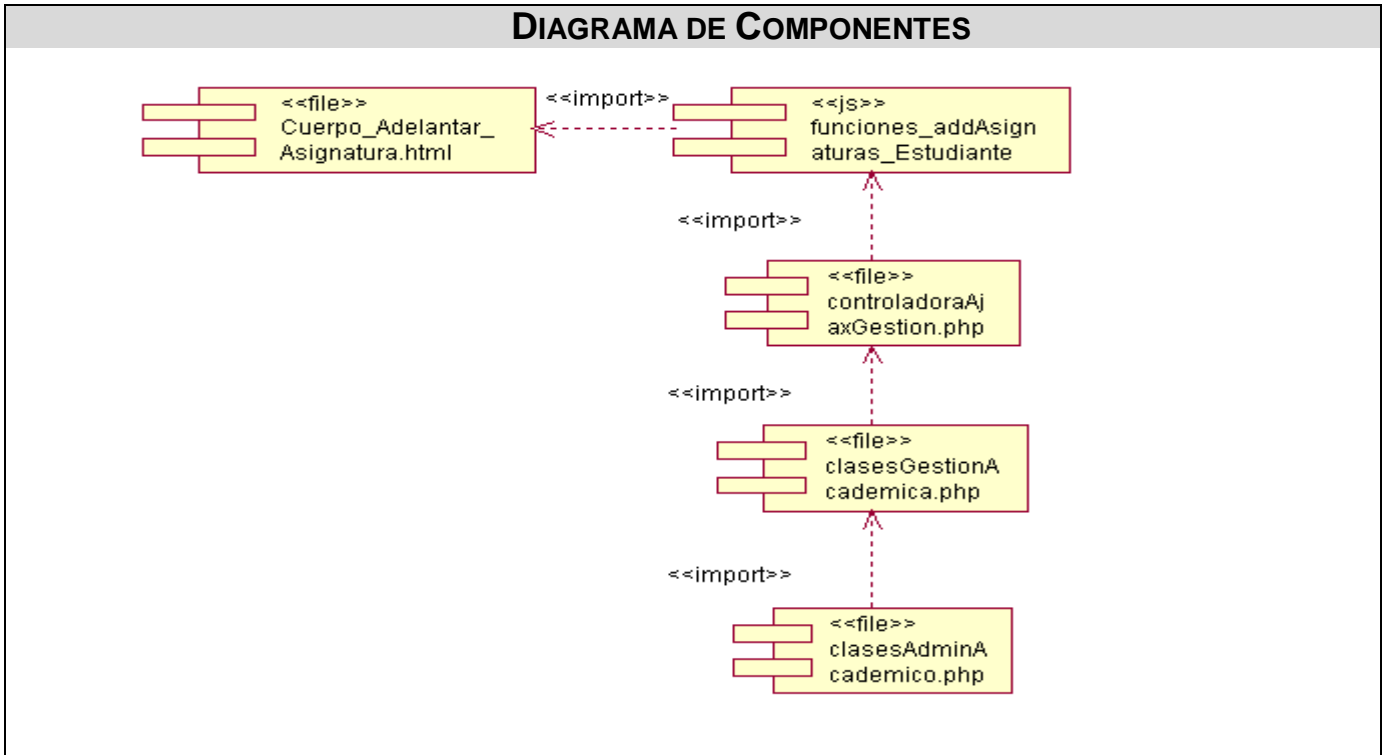


Fig 4.6 Diagrama de componentes Gestión Académica, caso de uso Adelantar Asignatura.

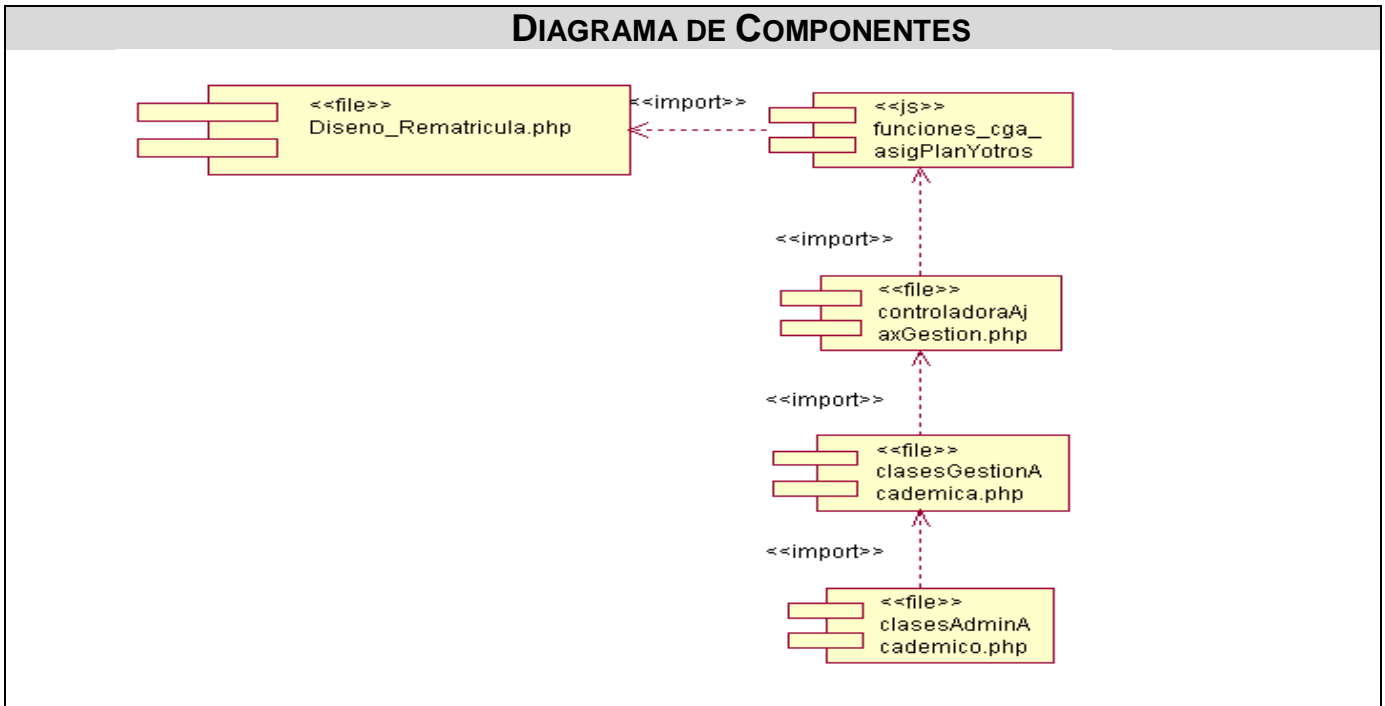


Fig 4.7 Diagrama de componentes Gestión Académica, caso de uso Rematricular.

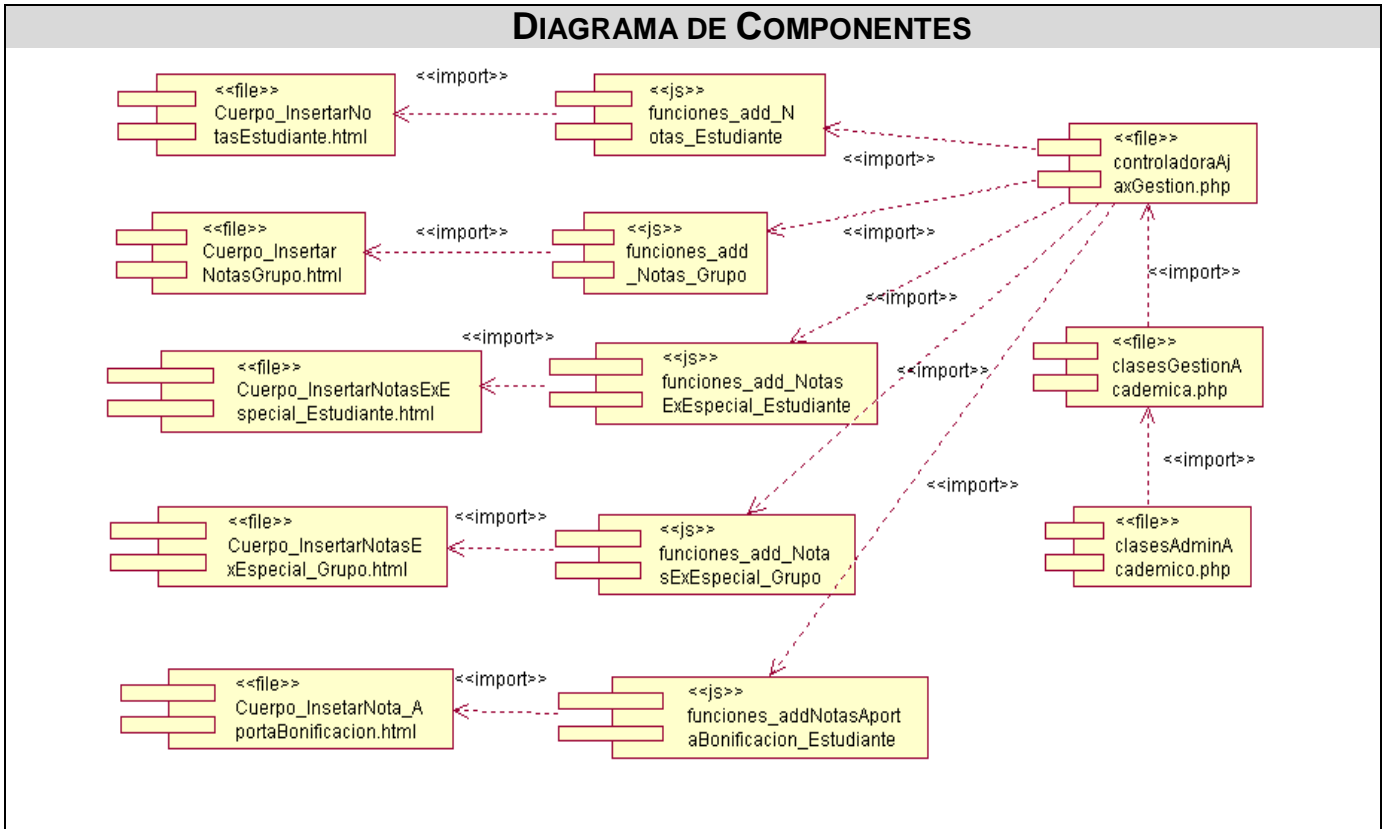


Fig 4.8 Diagrama de componentes Gestión Académica, caso de uso Gestionar Evaluaciones.

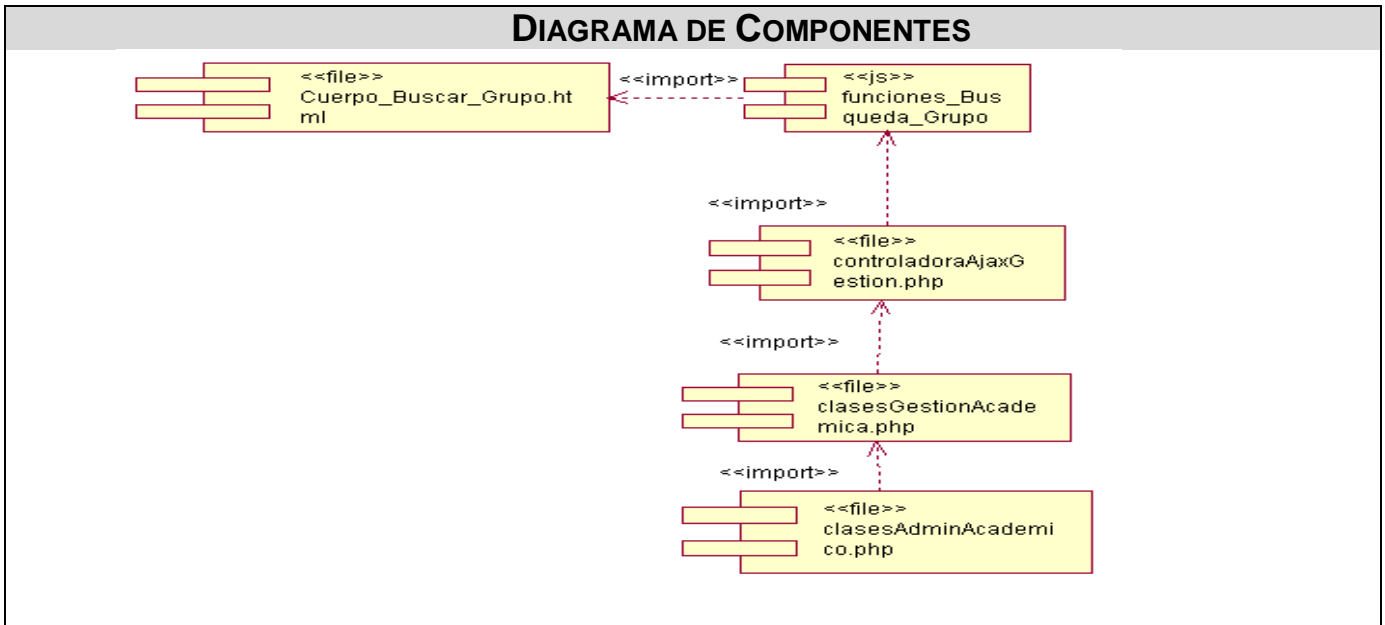


Fig 4.9 Diagrama de componentes Gestión Académica, caso de uso Buscar Grupo.

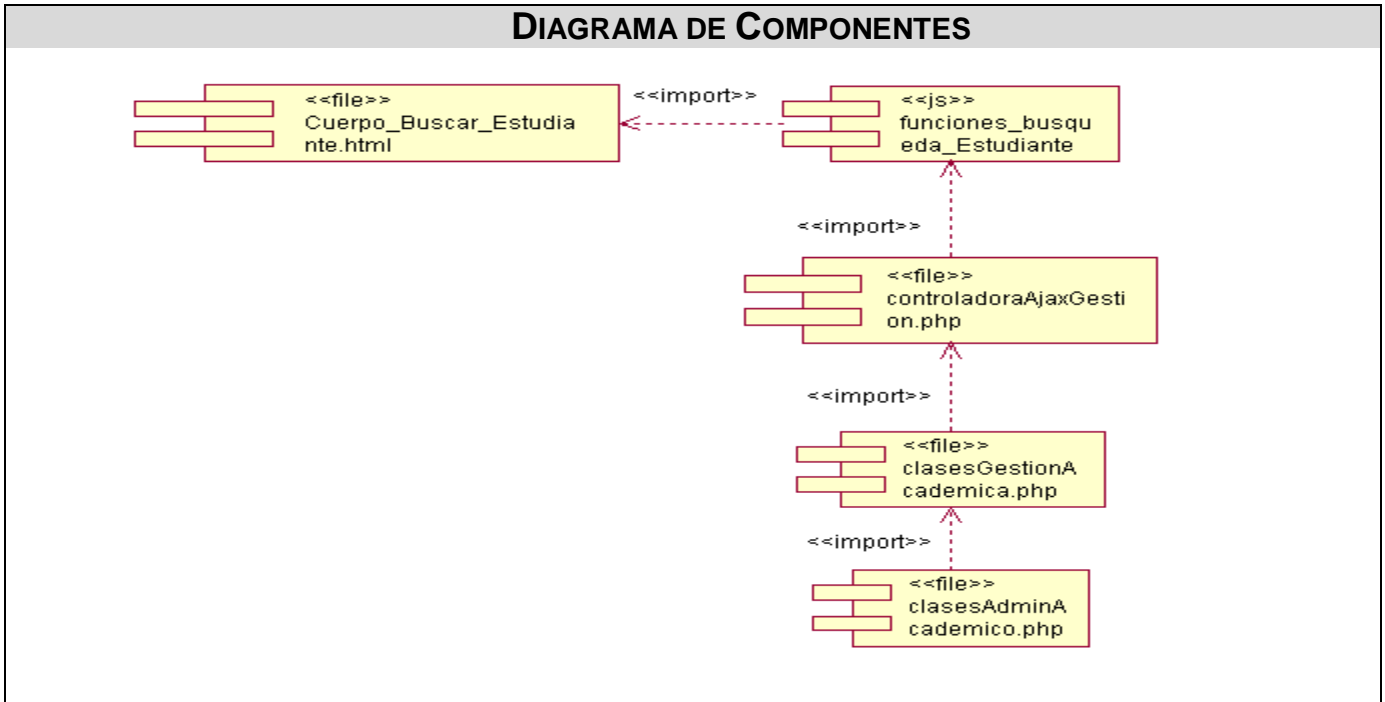


Fig 4.10 Diagrama de componentes Gestión Académica, caso de uso Buscar Estudiante.

4.2 Diagrama de despliegue.

El diagrama de despliegue describe la distribución física de los componentes sobre los nodos. La computadora cliente del siguiente diagrama representa cada una de las secretarías docentes de las facultades médicas y los puestos de trabajo a nivel nacional ubicados en el MINSAP, conectadas respectivamente a una impresora mediante los puertos USB ó LPT01, en dependencia del dispositivo con que se cuente en cada caso. Para las pc clientes, es necesario contar con un navegador que capaz de interpretar código javascript.

La distribución de los servidores corresponde a un servidor Web y de Aplicación conectado a la Red Telemática de Salud INFOMED y un tercero de Base de Batos que puede o no estar en el mismo servidor en que se encuentran los anteriores. Esta distribución depende de las posibilidades y necesidades del cliente. Los servidores estarán montados sobre Linux para garantizar una mayor seguridad y rendimiento.

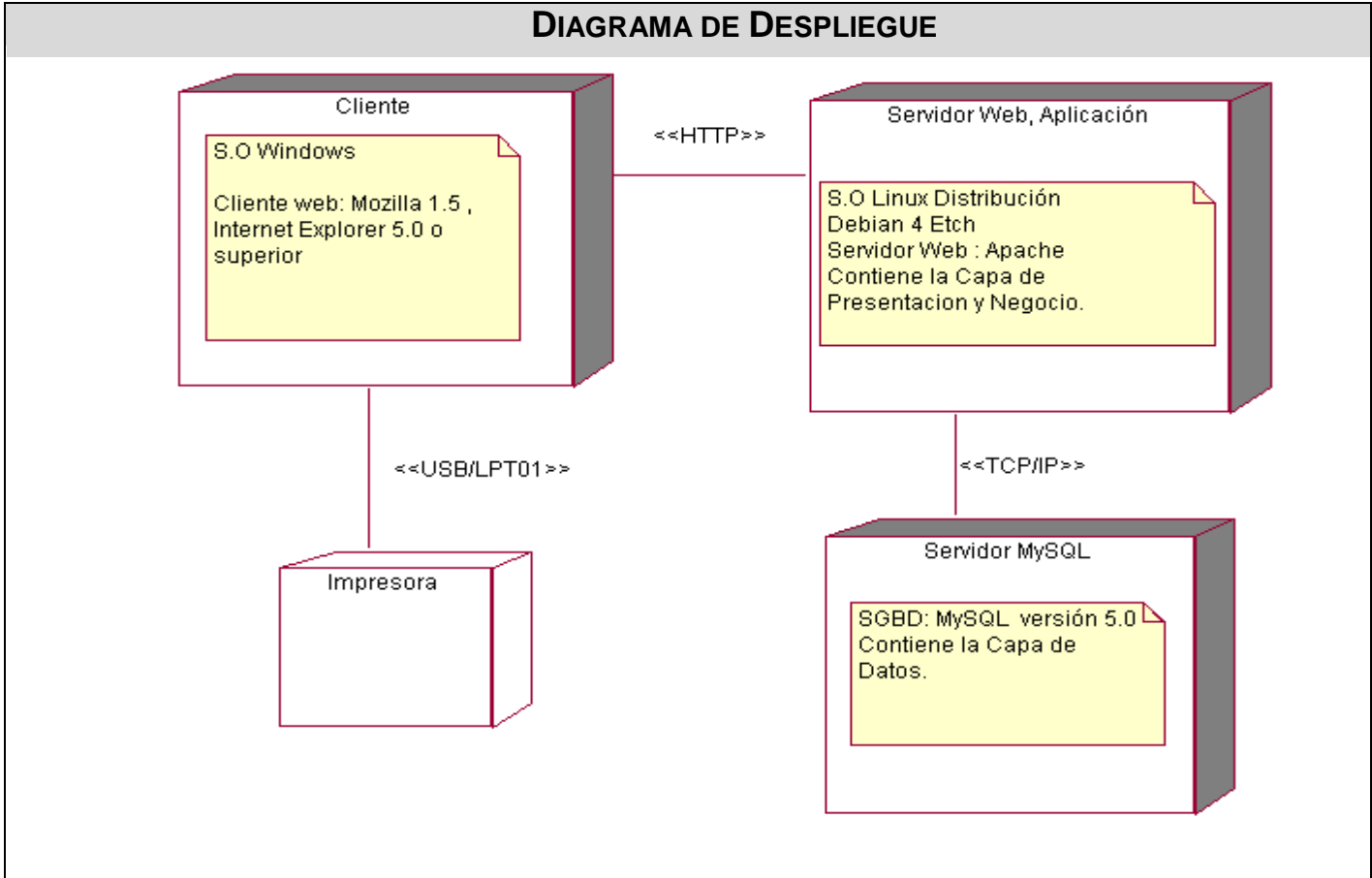


Fig 4.11 Diagrama de despliegue.

Conclusiones:

En este capítulo se abordó lo referente al flujo de trabajo de implementación, mostrándose los diagramas de componentes y de despliegue como parte de los artefactos que se generan en esta etapa. Una vez concluida la modelación del Componente de Gestión Académica y la fase de implementación en su totalidad. Se ha obtenido un producto con todas las funcionalidades prefijadas en los requisitos funcionales para ser incorporado al Sistema de Gestión de Información de Recursos Humanos en Salud.

CONCLUSIONES

Culminada la construcción del Componente de Gestión Académica, se llega a la conclusión de que se cumplieron los objetivos trazados obteniendo como resultado:

- El análisis del proceso de gestión académica en las instituciones clínico docentes del MINSAP, obteniendo una concepción del negocio a desarrollar y de los requerimientos funcionales del sistema.
- El diseño e implementación de un componente que permite realizar la gestión académica para el Sistema de Gestión de Información en el Proceso de Formación de Recursos Humanos en la Salud.

Gracias a la culminación del desarrollo del componente, las facultades médicas del país pueden contar con un sistema acorde con las necesidades actuales, que les facilitará el manejo de toda la información generada en el proceso de gestión académica. Además de obtener como principales beneficios: la estabilidad y actualidad de los datos; así como la integración a diferentes niveles de dirección, nacional y de sedes universitarias.

RECOMENDACIONES

La solución propuesta corresponde a un intento por adaptarse a las necesidades actuales del MINSAP para la gestión de los procesos académicos. En virtud de mejorar el componente se recomienda:

- Optimizar la gestión del plan de estudio adaptando la base de datos para que permita reutilizar los períodos para diferentes planes.
- Incrementar los reportes estadísticos según las necesidades que surjan a partir del uso diario de la aplicación.
- Optimizar la forma de registrar la identidad de los profesores responsables de las evaluaciones con la existencia en el sistema de un nomenclador de profesores que permita identificar a qué asignatura y grupo pertenece cada uno.
- Emplear el framework de implementación Symphony en el componente.

Por la importancia que reviste el trabajo para la gestión académica, se recomienda su divulgación para su uso e implementación en el resto de los centros de enseñanzas del país de acuerdo a las características específicas de estos.

REFERENCIAS BIBLIOGRÁFICAS.

1. Aplicación web. [Online] [Cited: Febrero 5, 2008.]
http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web.
2. Internet y la Web. [Online] [Cited: Febrero 5, 2008.]
http://www.portalplanetasedna.com.ar/web_internet.htm.
3. Familia de protocolos de Internet. [Online] [Cited: Febrero 8, 2008.]
http://es.wikipedia.org/wiki/Familia_de_protocolos_de_Internet.
4. Funcionamiento de la Web. [Online] [Cited: Febrero 15, 2008.]
<http://www.geocities.com/SiliconValley/Bay/8259/contenido.html>.
5. Usabilidad y arquitectura del software. [Online] [Cited: Febrero 15, 2008.]
<http://www.desarrolloweb.com/articulos/1622.php>.
6. Cristiá, Maximiliano. Introducción a la Arquitectura del Software. [Online] 2007. [Cited: Febrero 10, 2008.] <http://www.fceia.unr.edu.ar/ingsoft/intro-arq.pdf>.
7. Ídema la referencia 6. [Online]
8. KICILLOF. C. R.-N. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. 2004.
9. Alonso, Reinier. *Sistema de Gestión de Información en el Proceso de Formación de Recursos Humanos en Salud*. Ciudad de La Habana : s.n., 2007.
10. Definición arquitectura cliente servidor. [Online] [Cited: Febrero 16, 2008.]
<http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml#intro>.
11. Ídem a la referencia 10. [Online]
12. Oktaba, Hanna. Introducción a Patrones. [Online] [Cited: Febrero 20, 2008.]
<http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html>.
13. RODRÍGUEZ, J. T. Q. *Sistemas Distribuidos*.
14. SISTEMA GESTORES DE BASES DE DATOS. [Online] 2007. [Cited: Marzo 2, 2008.]
http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_base_de_datos.
15. Sistema gestores de Bases de datos, Guía Didáctica, Grado Superior. [Online] [Cited: Marzo 2, 2008.]

http://www.librosaulamagna.com/libro/SISTEMAS_GESTORES_DE_BASES_DE_DATOS._Guia_didactica._Grado_superior/18624/1762

16. Ídem a la referencia 9. [Online]

17. Ídem a la referencia 9. [Online]

18. Ídem a la referencia 9. [Online]

19. Ídem a la referencia 9. [Online]

BIBLIOGRAFÍA

- Alonso, Reinier. *Sistema de Gestión de Información en el Proceso de Formación de Recursos Humanos en Salud*. Ciudad de La Habana : s.n., 2007.
- Aplicación web. [Online] [Cited: Febrero 5, 2008.]
http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web
- Definición arquitectura cliente servidor. [Online] [Cited: Febrero 16, 2008.]
<http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml#intro>.
- BRADENBAUGH, J. *Aplicaciones JavaScript*. S.A., A. M., 2000.
- BRUEGGE, B. D., A. *Ingeniería de Software Orientado a Objetos*. Prentice Hall – Pearson Educación. 2002. p.
- CONALLEN, J. *Modeling web application architectures with UML*. 2000. p.
- Cristiá, Maximiliano. Introducción a la Arquitectura del Software. [Online] 2007. [Cited: Febrero 10, 2008.] <http://www.fceia.unr.edu.ar/ingsoft/intro-arq.pdf>.
- Familia de protocolos de Internet. [Online] [Cited: Febrero 8, 2008.]
http://es.wikipedia.org/wiki/Familia_de_protocolos_de_Internet.
- FRANCO NAVARRO, J. A. *UML en acción. Modelando Aplicaciones Web*, 2005.
- Funcionamiento de la Web. [Online] [Cited: Febrero 15, 2008.]
<http://www.geocities.com/SiliconValley/Bay/8259/contenido.html>.
- GALLEGO VÁZQUEZ, J. A. *Desarrollo Web con PHP y MySQL*. S.A, A. M., 2003.
- GILFILLAN, I. *La Biblia MySQL*., 2002.
- Internet y la Web. [Online] [Cited: Febrero 5, 2008.]
http://www.portalplanetasedna.com.ar/web_internet.htm.
- KICILLOF. C. R.-N. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. 2004.
- LARMAN, C. *UML y Patrones*. Prentice Hall Iberoamericana. 1999. p.
- Manual MySQL Server 5*. 2006. [Disponible en: <http://www.lawebdelprogramador.com>
- MUSCIANO, C. B. K. *HTML The Complete Guide* S.A., M.-H. I. E., 1999. 2.
- PRESSMAN, R. *Ingeniería del Software. Un enfoque práctico*. Mc Graw-Hill/ Interamericana de España. 2002. p.
- Oktaba, Hanna. Introducción a Patrones. [Online] [Cited: Febrero 20, 2008.]
<http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html>.

RODRÍGUEZ, J. T. Q. *Sistemas Distribuidos*.

SISTEMA GESTORES DE BASES DE DATOS. [Online] 2007. [Cited: Marzo 2, 2008.]
http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_base_de_datos.

Sistema gestores de Bases de datos, Guía Didáctica, Grado Superior. [Online] [Cited: Marzo 2, 2008.] http://www.librosaulamagna.com/libro/SISTEMAS_GESTORES_DE_BASES_DE_DATOS._

Reglamento para la organización del proceso docente educativo en los centros de educación médica superior., RESOLUCIÓN MINISTERIAL No. 15–1988 DEL MINSAP.

REYNOSO, C. B. *Introducción a la Arquitectura de Software*, 2004.

RUMBAUGH, J. J., IVAR; BOOCH, GRADY *El proceso unificado de desarrollo*. Addison Wesley.
2000a. p.

RUMBAUGH, J. J., IVAR; BOOCH, GRADY. *El lenguaje unificado de modelado*. Addison Wesley.
2000b. p.

Usabilidad y arquitectura del software. [Online] [Cited: Febrero 15, 2008.]
<http://www.desarrolloweb.com/articulos/1622.php>.

1. ANEXOS #1: MODELO DE NEGOCIO.

Actores del Negocio.

Actor	Descripción
Profesor	Interesado en procesar los resultados docentes alcanzados por el estudiante.
Estudiante	Interesado en que se le procese la información académica durante su estancia en el centro docente.
Facultades médicas	Interesado en hacer una solicitud para la confección de un nuevo Plan de estudio.

Tabla 1.1 Actores del Negocio.

Trabajadores del Negocio.

Trabajador	Descripción
Secretario docente	Es el encargado de realizar los procesos en la secretaría docente.
Metodólogo del MINSAP	Responsable de confeccionar y distribuir un nuevo plan de estudio que se acatará en las diferentes instituciones médicas según la especialidad.

Tabla 1.2 Trabajadores del Negocio.

Diagrama de Casos de Uso del negocio

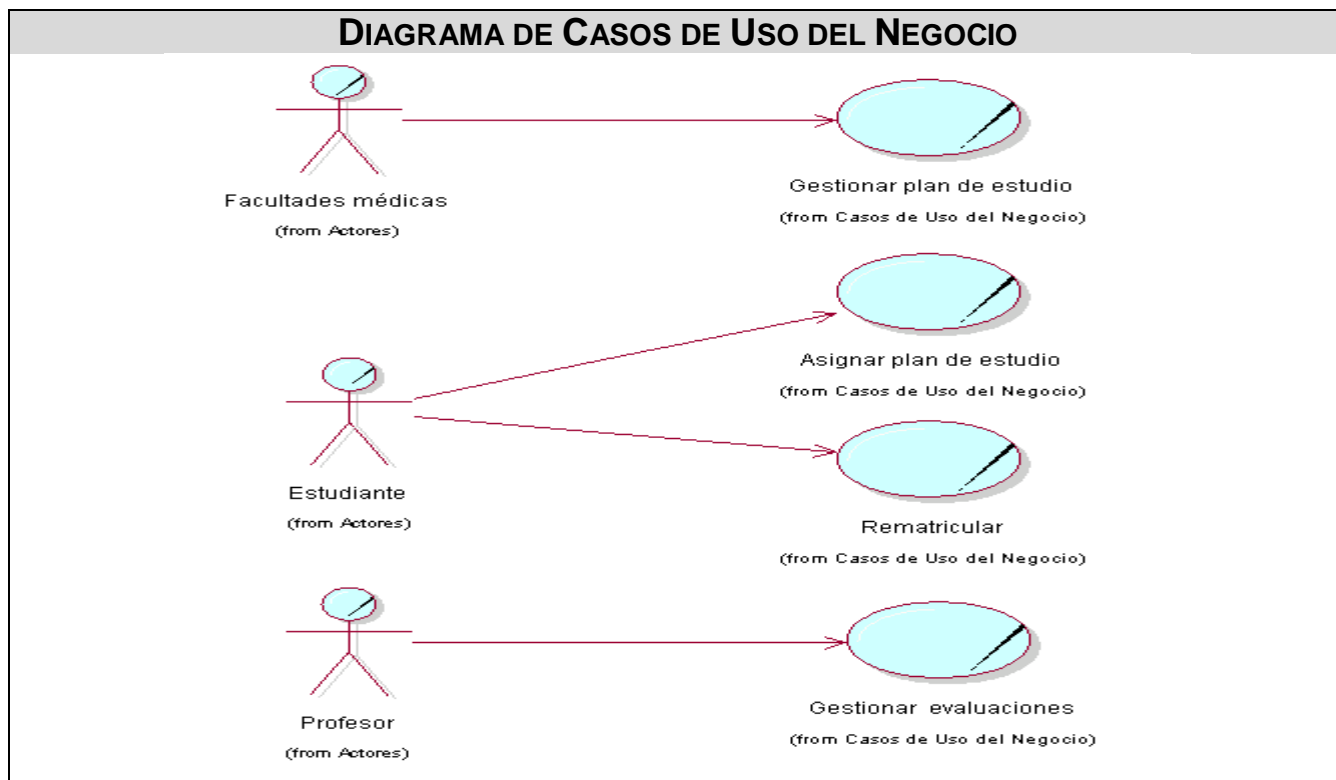


Fig 1 Diagrama de casos de uso del negocio.

Casos de uso del negocio

Descripción del Caso de Uso del Negocio Asignar Plan de Estudio.

Caso de uso del negocio	Asignar Plan de Estudio
Actores	Estudiante
Resumen	El caso de uso se inicia cuando el estudiante se presenta a recibir un plan de estudio por el cual se va a registrar el alumno a partir de ese momento.
Casos de uso asociados	
Acción del actor	Respuesta del proceso de negocio

Sección Asignar	
1. El estudiante se presenta para recibir un plan de estudio.	2. Se le asigna un plan de estudio por el cual se va a regir el alumno a partir de ese momento.
Mejoras propuestas	La mejora es que se le asigne de forma automática el plan de estudio a cada estudiante mediante un sistema integrado con el nivel nacional para recibir el plan de estudio directamente.

Tabla 1.3 Definición del CU Asignar Plan de Estudio.

Diagrama de actividades de Asignar Plan de Estudio

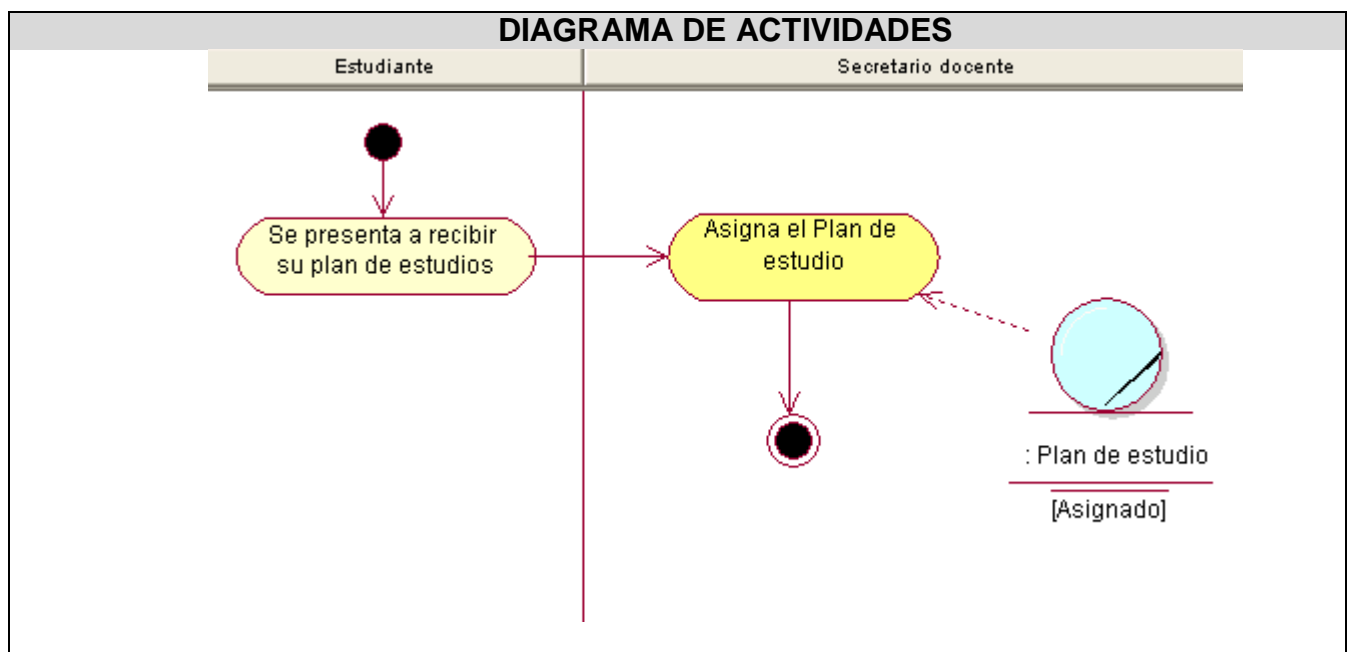


Fig 2 Diagrama de actividades del CU Asignar Plan de Estudio.

Descripción del Caso de Uso del Negocio Gestionar Evaluaciones.

Caso de uso del negocio	Gestionar Evaluaciones
Actores	Profesor
Resumen	El caso de uso se inicia cuando el profesor realiza una evaluación a los estudiantes la cual tiene que ser archivada en la secretaría docente.
Casos de uso asociados	
Acción del actor	Respuesta del proceso de negocio
1. El profesor realiza una evaluación.	2. Se archiva en la secretaría docente el resultado adquirido en la evaluación.
Mejoras propuestas	La mejora es que se archive la evaluación en formato digital a través de una interfaz amigable que permita calcular el promedio de forma automática una vez que se realicen las evaluaciones finales.

Tabla 1.4 Definición del CU Gestionar Evaluaciones.

Diagrama de actividades del Caso de Uso del Negocio Gestionar Evaluaciones.

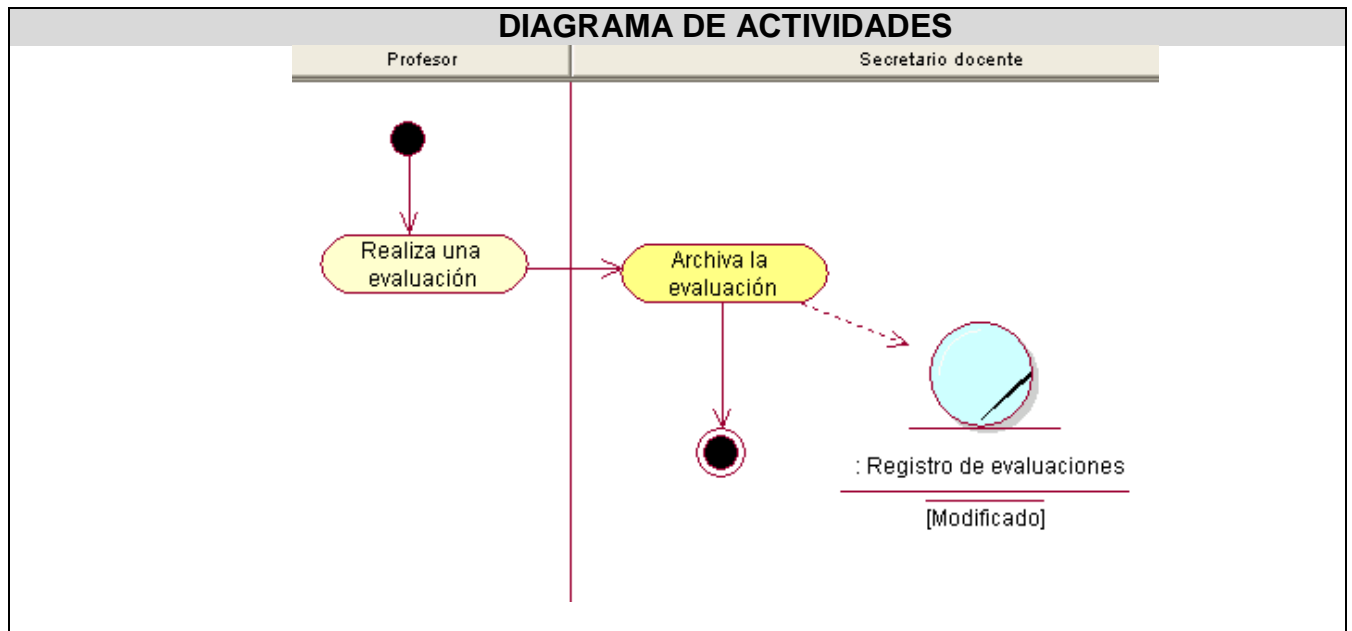


Fig 3 Diagrama de actividades del CU Gestionar Evaluaciones.

Descripción del Caso de Uso del Negocio Rematricular.

Caso de uso del negocio	Rematricular
Actores	Estudiante
Resumen	El caso de uso se inicia cuando el estudiante culmina el curso académico y pasa a realizar su rematrícula a la secretaría docente.
Casos de uso asociados	
Acción del actor	Respuesta del proceso de negocio
<p>1. El estudiante culmina el curso académico.</p> <p>4. Responde a las preguntas de la entrevista para recoger los datos de la rematrícula.</p>	<p>2. El secretario docente revisa el registro de matrículas para confirmar que el estudiante se encuentra matriculado.</p> <p>3. El secretario entrevista al estudiante para recoger los datos de rematrícula.</p> <p>5. Archiva los datos de rematrícula en el Registro de matrículas.</p>
Mejoras propuestas	La mejora que se propone del proceso de rematrícula es que exista una interfaz amigable donde la recogida de los datos se realice de forma rápida y eficiente.

Tabla 1.3 Definición del CU Rematricular.

Diagrama de actividades del Caso de Uso del Negocio Rematricular.

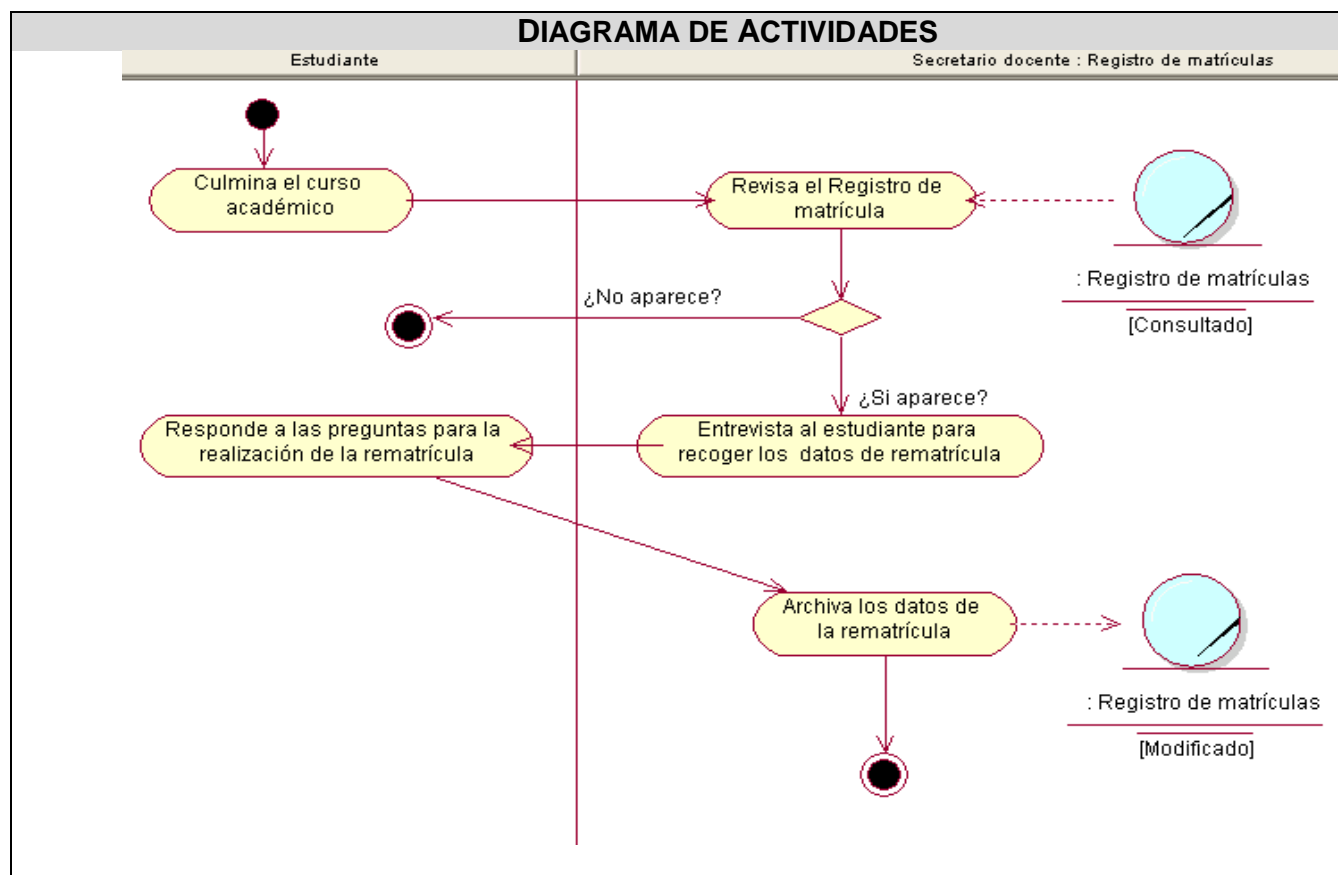


Fig 3 Diagrama de actividades del CU Rematricular.

Descripción del Caso de Uso del Negocio Gestionar Plan de Estudio.

Caso de uso del negocio	Gestionar Plan de Estudio
Actores	Dirección del MINSAP
Resumen	El caso de uso se inicia cuando las facultades médicas se ven en la necesidad de cambiar el plan de estudio existente y crear uno nuevo.
Casos de uso asociados	
Acción del actor	Respuesta del proceso de negocio
Sección 1	
1. Las facultades médicas del país se ven en la necesidad de cambiar el	2. El metodólogo nacional del MINSAP confecciona un nuevo plan de estudio para ser acatado por una

plan de estudio existente.	<p>especialidad médica.</p> <p>3. Se distribuye el nuevo plan de estudio a las sedes universitarias que lo requieran.</p> <p>4. El plan es recibido en las secretarías docentes de las facultades médicas.</p>
Mejoras propuestas	<p>La mejora es confeccionar planes de estudios con un alto grado de flexibilidad y además lograr una integración en la transmisión de información entre los diferentes niveles siendo más fácil la distribución por todo el país de los cambios que se generen a nivel central a las sedes universitarias en el sector de la salud.</p>

Tabla 1.5 Definición del CU Gestionar Plan de Estudio.

Diagrama de actividades del Caso de Uso del Negocio Gestionar Plan de Estudio.

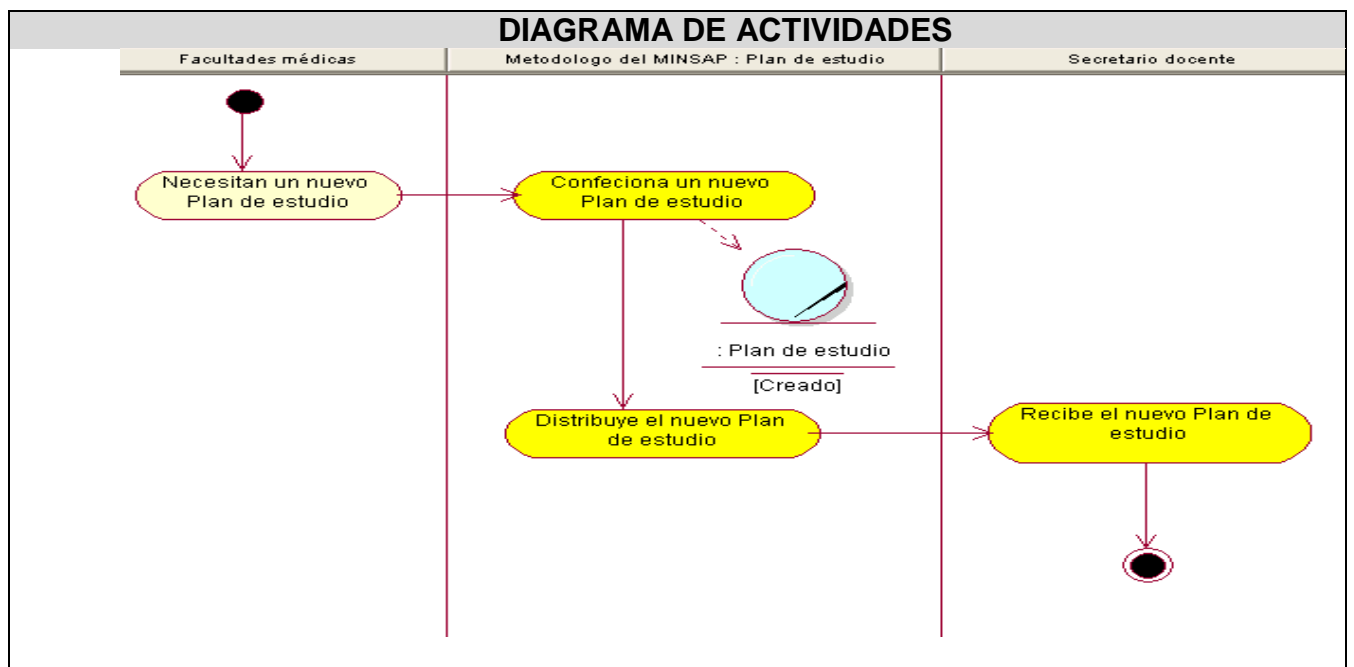


Fig 3 Diagrama de actividades del CU Gestionar Plan de Estudio.

Modelo de objetos

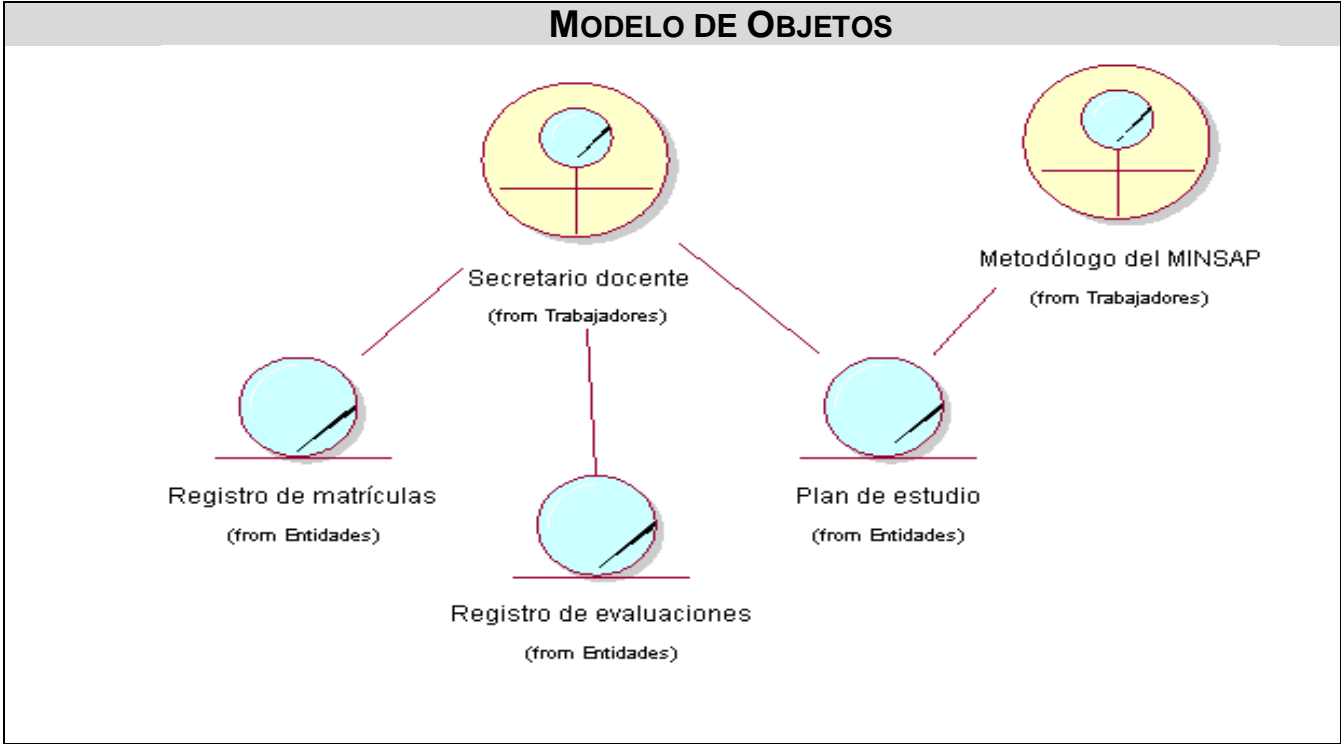


Fig 4 Modelo de Objetos.

ANEXOS #2: MODELO DE SISTEMA.

Diagrama de casos de uso

Los artefactos fundamentales que se utilizan en la captura de requisitos son el modelo de casos de uso, que incluye los casos de uso y los actores del sistema.

El modelo de casos de uso permite que los desarrolladores del software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. Describe lo que hace el sistema para cada tipo de usuario.

Este documento recoge los casos de uso que se definieron para el Componente de Gestión Académica pertenecientes al Sistema de Gestión de Información en el Proceso de Formación de Recursos Humanos en Salud.

Definición de los actores del sistema.

Actores	Justificación
Editor nacional	Es el encargado de confeccionar, eliminar o modificar el Plan de Estudio y los elementos que aportan bonificación al promedio del estudiante.
Editor sede	Secretario docente de los centros de estudios, es el responsable de asignar un plan de estudio a los estudiantes de la sede, registra además todas las evaluaciones que se realizan y se encarga realizar el proceso de matrícula. Solo tiene acceso a la información de su centro.

Tabla2.1 Definición de actores del sistema

Listados de casos de uso.

CU – A1	Gestionar Plan de Estudio.
Actor	Editor nacional.
Descripción	Este caso de uso se encarga de confeccionar un nuevo plan de estudio con todos los elementos que lo componen (pertenece a un estudio, contiene periodos, asignaturas, tipos de evaluación y evaluación). Por cada elemento se adiciona, edita o elimina según las corresponda.
Referencia	R1

CU – A2	Gestionar Elementos que Aportan Bonificación.
Actor	Editor nacional.
Descripción	Este caso de uso se encarga de adicionar, editar o eliminar un elemento que aporta bonificación al promedio de un estudiante.
Referencia	R2

CU – G1	Asignar Plan de Estudio.
Actor	Editor sede.
Descripción	Este caso de uso se encarga de asignar un plan de estudio a un grupo de estudiantes.
Referencia	R3

CU – G2	Gestionar Evaluaciones.
Actor	Editor sede.
Descripción	Este caso de uso se encarga de con registrar, modificar los resultados docentes de un estudiante o de un grupo docente. .
Referencia	R4

CU – G3	Rematricular.
Actor	Editor sede.
Descripción	Este caso de uso se encarga de verificar si un estudiante al terminar un año

	académico ha aprobado todas las asignaturas para pasarlo al próximo curso con las asignaturas correspondientes.
Referencia	R 8

CU – G4	Adelantar Asignatura.
Actor	Editor sede.
Descripción	Este caso de uso se encarga de posibilitar que un estudiante que esté interesado en adelantar alguna asignatura, se le reajuste su plan de estudio y la evaluación de la misma se registre en el período actual.
Referencia	R 7

CU – G5	Gestionar Reportes.
Actor	Editor sede.
Descripción	Este caso de uso se encarga de visualizar los resultados docentes de un estudiante o de un grupo docente.
Referencia	R 5, R6

Diagrama de casos de uso.

Los diagramas de casos de uso se mostrarán en dos paquetes para agruparlos los actores y casos de uso a nivel nacional y de sede según corresponda.

Paquete Administración Académica

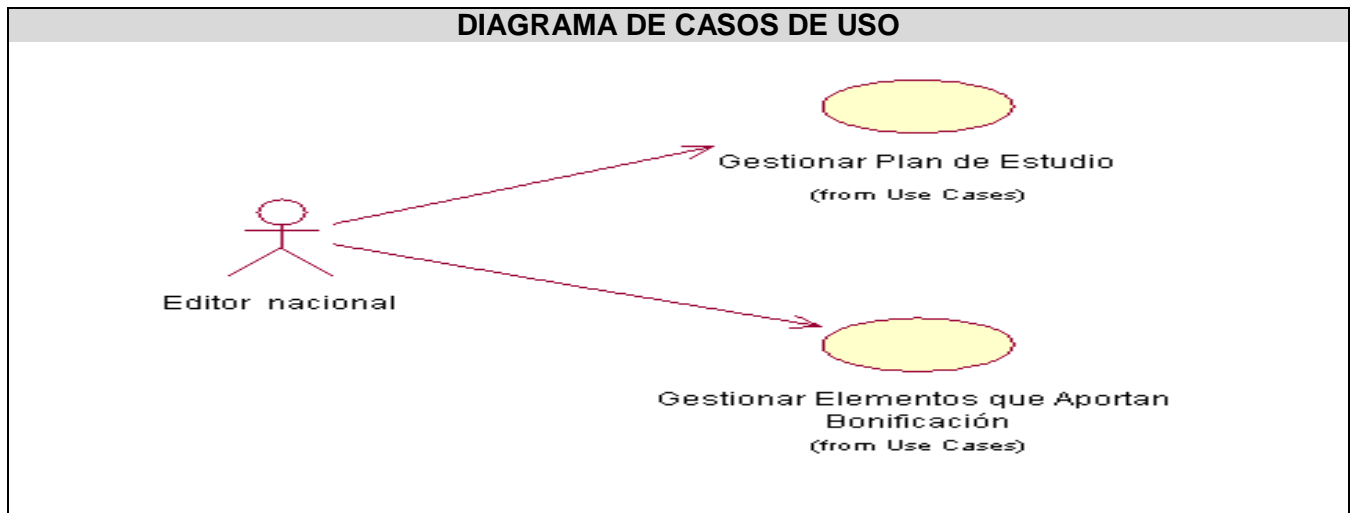


Fig1 Diagrama de casos de uso de Administración Académica

Paquete Gestión Académica

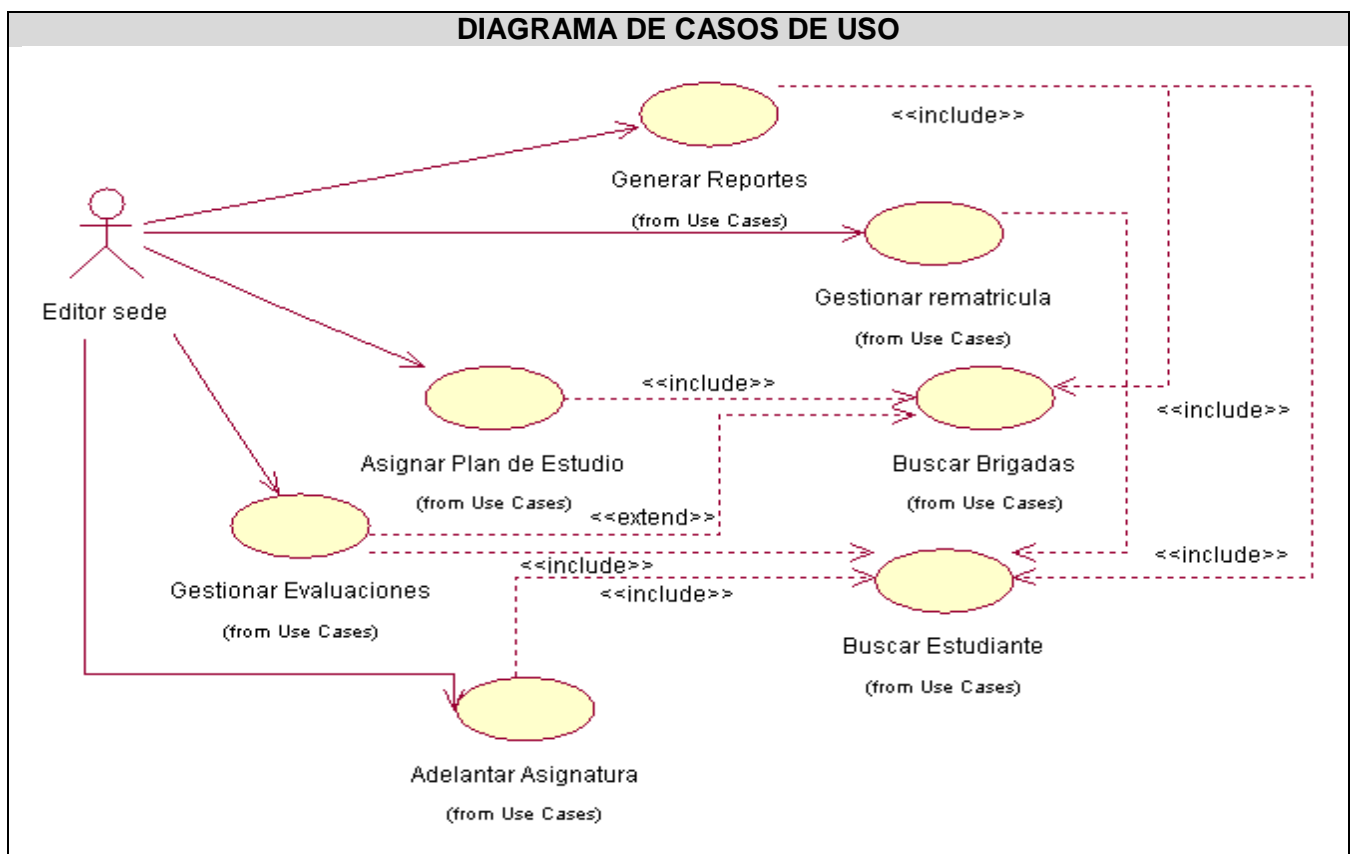


Fig2 Diagrama de casos de uso de Gestión Académica

Casos de uso expandidos.

Descripción Caso de Uso “Gestionar Elementos que Aportan Bonificación”.

Caso de Uso:	Gestionar Elementos que Aportan bonificación.	
Actores:	Editor nacional	
Resumen:	Este caso de uso se inicia cuando el editor nacional crea, elimina o modifica los elementos que aportan bonificación.	
Precondiciones:	Si se va a modificar o eliminar un elemento es necesario exista.	
Referencias	R2	
CU Asociados		
Prioridad	Media	
Flujo Normal de Eventos		
Acción del Actor		
Respuesta del Sistema		
1. El usuario accede a la sección correspondiente para la Gestión de Elementos que Aportan Bonificación.	2. Se muestra la interfaz correspondiente a Elementos que Aportan Bonificación 3. Se muestra una lista con todos los elementos que aportan bonificaciones existentes.	
Sección: Adicionar		
Acción del Actor		
Respuesta del Sistema		
1. Llena los campos: Nombre, Promedia (si o no), el Tipo de Evaluación (cuantitativa, cualitativa por lugar y cualitativa por relevancia) correspondientes a un elemento y da clic en el botón Adicionar.	2. El sistema adiciona el elemento. 3. Actualiza la lista con todos los elementos que aportan bonificación.	
Sección: Editar		

Acción del Actor		Respuesta del Sistema	
1. Selecciona de la lista de elementos el que se desee editar.		2. Aparece en pantalla los datos del elemento seleccionado.	
3. Modifican los datos necesarios y da clic en el botón Editar.		4. Modifica los datos. 5. Actualiza la lista con todos los elementos con los cambios efectuados.	
Sección Eliminar			
Acción del Actor		Respuesta del Sistema	
1. Selecciona de la lista de elementos que se desee eliminar y se da clic en el botón Eliminar.		2. Muestra en pantalla un cartel para confirmar que se desea eliminar el elemento o cancelar la operación.	
3. Da clic en el botón Aceptar		4. El sistema elimina el elemento. 5. Muestra en pantalla la lista de los elementos con los cambios efectuados.	
Flujo Alternativo			
Acción del Actor		Respuesta del Sistema	
1. Da clic en el botón Cancelar.		2. El sistema no realiza operación alguna.	
Poscondiciones			

Tabla 2.2 Descripción del CU Elementos que Aportan Bonificación

Descripción Caso de Uso "Asignar Plan de Estudio".

Caso de Uso:	Asignar plan de estudio.
Actores:	Editor sede.
Resumen:	Este caso de uso se inicia cuando el actor le va a asignar un plan de estudio a un grupo de estudiantes.
Precondiciones:	
Referencias	R3
CU Asociados	Buscar Brigada
Prioridad	Alta

Flujo Normal de Eventos	
Sección	
Acción del Actor	Respuesta del Sistema
1. El usuario accede a la sección correspondiente a Asignar Plan de Estudio.	2. Se muestra la interfaz correspondiente a Asignar Plan de Estudio. 3. Se cargan todos los planes de estudio distribuidos para las sedes universitarias.
4. Selecciona un plan de estudio la lista.	5. Se carga un panel con todos los elementos que conforman un plan de estudio (para ser visto en caso de que el usuario lo desee).
6. Selecciona la operación Asignar plan o Actualizar plan de una lista desplegable.	
Sección: Asignar	
Acción del Actor	Respuesta del Sistema
1. Selecciona la operación Asignar plan.	2. Aparecen en pantalla las brigadas que no tienen un plan de estudio asignado. 3. En caso de que no hayan brigadas disponibles se muestra un cartel notificándolo.
4. Selecciona la(s) brigada(s) a las que se les va a asignar el plan y se da clic en el botón Asignar Plan de Estudio.	5. Se asigna el plan de estudio a las brigadas seleccionadas.
Sección: Actualizar	
Acción del Actor	Respuesta del Sistema
1. Selecciona la operación actualizar plan.	2. Aparecen en pantalla todas las brigadas pertenecientes al plan seleccionado.
3. Selecciona la(s) brigada(s) a las que se les va a actualizar el plan de estudio escogido anteriormente.	

4. Selecciona un nuevo plan de la lista. 5. Da clic en el botón Actualizar Plan de Estudio.	6. Actualiza el plan de estudio en la(s) brigada(s) seleccionada(s).
Poscondiciones	

Tabla2.3 Descripción del CU Asignar Plan de Estudio

Descripción Caso de Uso “Gestionar Evaluaciones”.

Caso de Uso:	Gestionar Evaluaciones.	
Actores:	Editor sede	
Resumen:	Este caso de uso se inicia cuando el secretario docente a va a archivar las evaluaciones de los estudiantes de la sede.	
Precondiciones:	1. Cuando se va a insertar cualquier evaluación se debe realizar primero la búsqueda de un (varios) estudiante(s).	
Referencias	R4	
CU Asociados	Buscar Brigada, Buscar Estudiante	
Prioridad	Alta	
Flujo Normal de Eventos		
Sección: Insertar Notas Cuantitativas		
Acción del Actor	Respuesta del Sistema	
1. El usuario accede a la sección Insertar Notas Cuantitativas del estudiante o grupo buscado.	2. Se muestra la interfaz correspondiente a Insertar Notas Cuantitativas correspondiente al estudiante o el grupo buscado. 3. Se carga una lista que contiene todos los períodos del plan de estudio del estudiante o grupo buscado.	
4. Selecciona el período en el que desee insertar la evaluación la lista.	5. Carga automáticamente los tipos de evaluación y las asignaturas pertenecientes al período en una lista	

	desplegable respectivamente.
6. Selecciona el tipo de evaluación a insertar.	7. Se muestran todas las asignaturas del (los) estudiante(s) buscado(s) con la posibilidad de insertar la evaluación de cada una de las asignaturas y el tipo de evaluación.
8. Llena los datos. 9. Da clic en el botón Guardar.	10. El sistema guardan los cambios y muestra un cartel notificándolo.
Flujo Alterno	
Acción del Actor	Respuesta del Sistema
1. Selecciona una asignatura específica de la cual se quiere insertar una nota.	2. Aparece en pantalla solo la asignatura escogida con sus datos correspondientes.
3. Se llenan los datos y se da clic en el botón Guardar.	4. El sistema guardan los cambios y muestra un cartel notificándolo.
Sección: Insertar Notas Cualitativas	
Acción del Actor	Respuesta del Sistema
1. El usuario accede a la página Insertar Notas Cualitativas del estudiante o grupo buscado.	2. Se muestra la interfaz correspondiente a Insertar Notas Cualitativas correspondiente al estudiante o el grupo buscado. 3. Se carga una lista que contiene todos los períodos del plan de estudio del estudiante o grupo buscado.
4. Selecciona el período en el que se desee insertar la evaluación de la lista.	5. Carga automáticamente los tipos de evaluación y las asignaturas pertenecientes al período en una lista desplegable respectivamente.
6. Selecciona el tipo de evaluación a insertar.	7. Muestra todas las asignaturas del (los) estudiante(s) buscado(s) con la posibilidad de insertar la evaluación de cada una de las asignaturas y el tipo de

	evaluación.
8. Llenan los datos. 9. Da clic en el botón Guardar.	10. El sistema guardan los cambios y muestra un cartel notificándolo.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
1. Selecciona una asignatura específica de la cual se quiere insertar una nota.	2. Aparece en pantalla solo la asignatura escogida con sus datos correspondientes.
3. Llena los datos 4. Da clic en el botón Guardar.	5. El sistema guardan los cambios y muestra un cartel notificándolo.
Sección: Insertar Notas Exámenes Especiales	
Acción del Actor	Respuesta del Sistema
1. El usuario accede a la página Insertar Notas Exámenes Especiales del estudiante o grupo buscado.	2. Se muestra la interfaz correspondiente a Insertar Notas Exámenes Especiales correspondiente al estudiante o el grupo buscado. 3. Se carga una lista que contiene todos los períodos del plan de estudio del estudiante o grupo buscado.
4. Se selecciona el período en el que se desee insertar la evaluación de la lista.	5. Carga automáticamente los exámenes especiales pertenecientes al período en una lista desplegable.
6. Selecciona el examen especial a insertar.	7. Cargan automáticamente las convocatorias realizadas o se da la posibilidad de escoger una nueva. 8. Muestra en la interfaz el examen

	especificado con la convocatoria para el (los) estudiante(s) buscado(s) con la posibilidad de insertar la evaluación.
9. Llena los datos. 10. Da clic en el botón Guardar.	11. El sistema guardan los cambios y muestra un cartel notificándolo.
Sección: Insertar Notas Aportan Bonificación	
Acción del Actor	Respuesta del Sistema
1. El usuario accede a la página Insertar Notas Aportan Bonificación de un estudiante buscado.	2. Muestra la interfaz Insertar Notas Aportan Bonificación en la que se muestran todos los elementos que aportan bonificación con su correspondiente evaluación perteneciente al estudiante buscado.
3. Se selecciona el tipo de elemento que aportan bonificación de una lista desplegable, en este caso puedes ser Exámenes de Premio u Otros.	4. En caso de seleccionar Exámenes de Premio se cargan automáticamente los existentes con posibilidad de escoger una evaluación.
5. Selecciona un examen de premio a insertar. 6. Selecciona la evaluación y se da clic en el botón Guardar.	7. El sistema verifica que el estudiante no contenga ese examen, en este caso se muestra un cartel notificándolo. 8. Guarda los cambios y muestra un cartel notificándolo.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
1. Selecciona en Tipo de Examen que aporta bonificación la opción Otros.	2. En este caso aparecen en pantalla una lista desplegable con otros exámenes existentes.
3. Selecciona uno de estos elementos a insertar.	4. Muestra el examen con al posibilidad de escoger una nota.

5. Selecciona la evaluación y se da clic en el botón Guardar.	6. El sistema verifica que el estudiante no contenga ese examen, en este caso se muestra un cartel notificándolo. 7. El sistema guarda los cambios y muestra un cartel notificándolo.
Sección: Insertar Notas Aportan Bonificación “Editar”	
Acción del Actor	Respuesta del Sistema
1. El usuario accede a la página Insertar Notas Aportan Bonificación de un estudiante buscado.	2. Se muestra la interfaz Insertar Notas Aportan Bonificación mostrando todos los elementos que aportan bonificación con su correspondiente evaluación perteneciente al estudiante buscado.
3. Selecciona el tipo de elemento que aportan bonificación que se desee editar de la lista mostrada en pantalla y se da clic en el botón Editar.	4. Se da la posibilidad de modificar la evaluación.
5. Modifica la evaluación 6. Da clic en el botón Guardar.	7. El sistema verifica si se realizó algún cambio, en caso de no hacerlo aparece un cartel notificándolo. 8. El sistema guarda los cambios y aparece un cartel notificándolo.
Sección: Insertar Notas Aportan Bonificación “Eliminar”	
Acción del Actor	Respuesta del Sistema
1. El usuario accede a la página Insertar Notas Aportan Bonificación de un estudiante buscado.	2. Se muestra la interfaz Insertar Notas Aportan Bonificación mostrando todos los elementos que aportan bonificación con su correspondiente evaluación perteneciente al estudiante buscado.
3. Selecciona el tipo de elemento que aportan bonificación que se desee eliminar de la lista	4. Aparece un cartel para verificar si se desea realizar esta operación.

mostrada en pantalla y se da clic en el botón Eliminar.	
5. En caso de que se desee eliminar se da en el botón Aceptar.	6. El sistema elimina el elemento y aparece un cartel notificándolo.
Flujo Alterno	
Acción del Actor	Respuesta del Sistema
1. En caso de que no se desee eliminar se da clic en el botón Cancelar.	2. El sistema no realiza operación alguna.
Poscondiciones	

Tabla2.4 Descripción del CU Gestionar Evaluaciones.

Descripción Caso de Uso “Rematricular”.

Caso de Uso:	Gestionar Rematricula.
Actores:	Editor sede
Resumen:	Este caso de uso se inicia cuando el secretario docente procede a realizar la rematrícula de algún estudiante.
Precondiciones:	
Referencias	R8
CU Asociados	Buscar Estudiante
Prioridad	Media
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Realiza la búsqueda de un estudiante que vaya a rematricular.	2. Aparece en pantalla el estudiante buscado.
3. Accede a la página Rematrícula.	4. Se muestra la página correspondiente a Rematrícula apareciendo una lista de las asignaturas que tiene pendientes y las que tiene aprobadas.

5. Da clic en el botón Continuar Estudios.	6. Pone al estudiante en el siguiente año de la carrera en caso de que tenga todas las asignaturas aprobadas, de lo contrario, el sistema lo matriculará en el mismo año académico y período solamente con la asignatura desaprobada.
Poscondiciones	

Tabla2.5 Descripción del CU Rematricular

Descripción Caso de Uso “Adelantar Asignatura”.

Caso de Uso:	Adelantar Asignatura.
Actores:	Editor sede
Resumen:	Este caso de uso se inicia cuando el secretario docente realiza la búsqueda de un estudiante que desea adelantar una asignatura.
Precondiciones:	
Referencias	R7
CU Asociados	Buscar Estudiante
Prioridad	Media
Flujo Normal de Eventos	
Acción del Actor	
Respuesta del Sistema	
1. Realiza la búsqueda de un estudiante que desea adelantar una asignatura.	2. Se muestra en pantalla el nombre del estudiante buscado.
3. Accede a la página Adelantar Asignatura.	4. Se muestra la interfaz correspondiente cargando en una lista de las asignaturas que el estudiante está adelantando.
5. Selecciona la asignatura que se desea adelantar	7. Se agrega esta asignatura a las demás que está recibiendo el estudiante como parte del plan de estudio
6. Da clic en el botón Agregar al período actual.	

	correspondiente.
Editar una asignatura a adelantar	
Acción del Actor	Respuesta del Sistema
1. Marca la asignatura que se desea editar.	2. Se carga la asignatura en un select.
3. Modifican los datos. 4. Da clic en el botón Editar.	5. Guarda los cambios.
Eliminar una asignatura a adelantar	
Acción del Actor	Respuesta del Sistema
1. Marca la asignatura que se desea eliminar.	2. Se carga la asignatura en un select.
3. Da clic en el botón Eliminar.	4. El sistema verifica que la asignatura no posea evaluaciones existentes y lo elimina.
Flujo Alternativo	
	1. En caso de que la asignatura que se desee eliminar contenga evaluaciones no se puede proceder a eliminar mostrándose un cartel que notifique este hecho.
Poscondiciones	

Tabla2.6 Descripción del CU Adelantar Asignatura.

ANEXO # 3: MODELO DE DISEÑO.

Paquete Administración Académica.

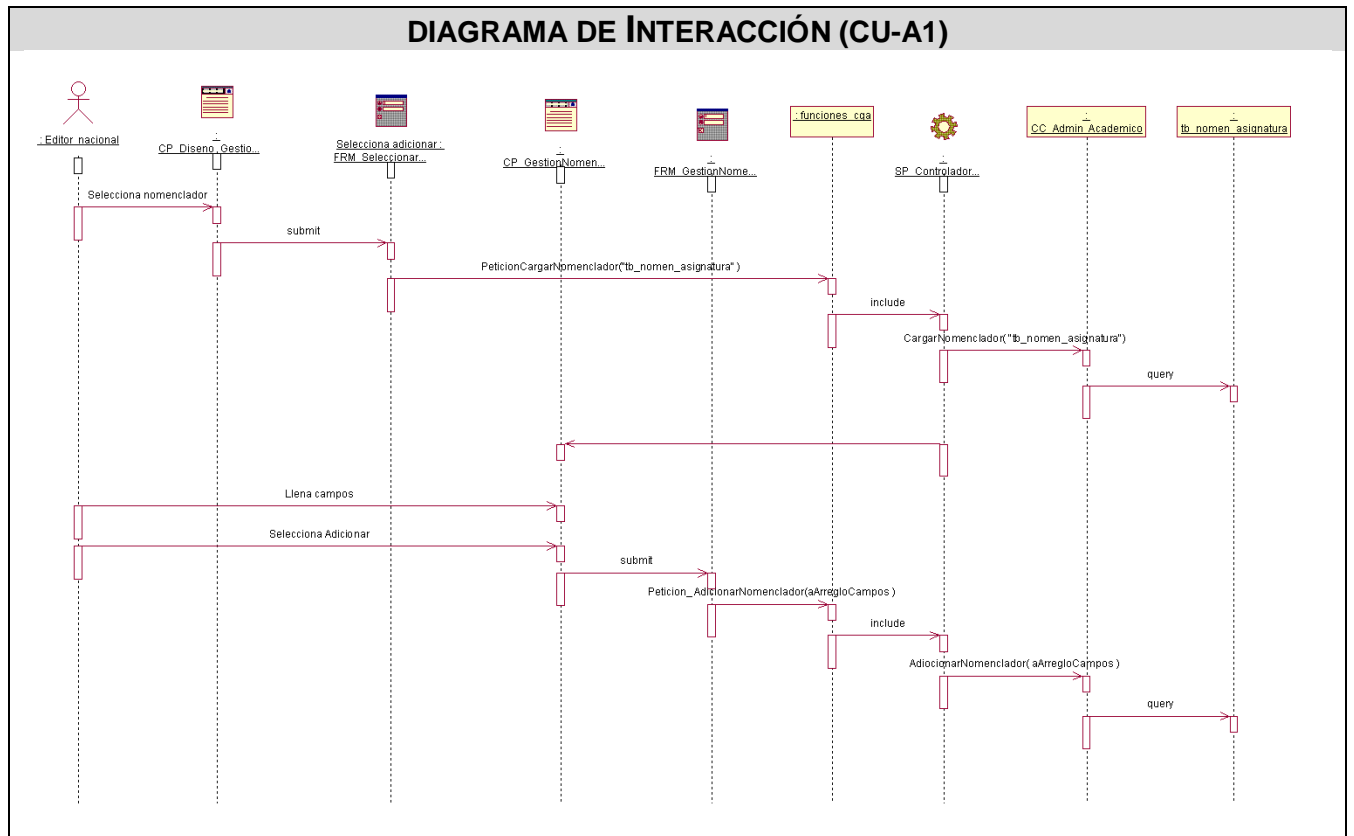


Fig 1 Diagrama de interacción del caso de uso Gestionar Plan de Estudio, sección Adicionar Asignatura.

Paquete Gestión Académica

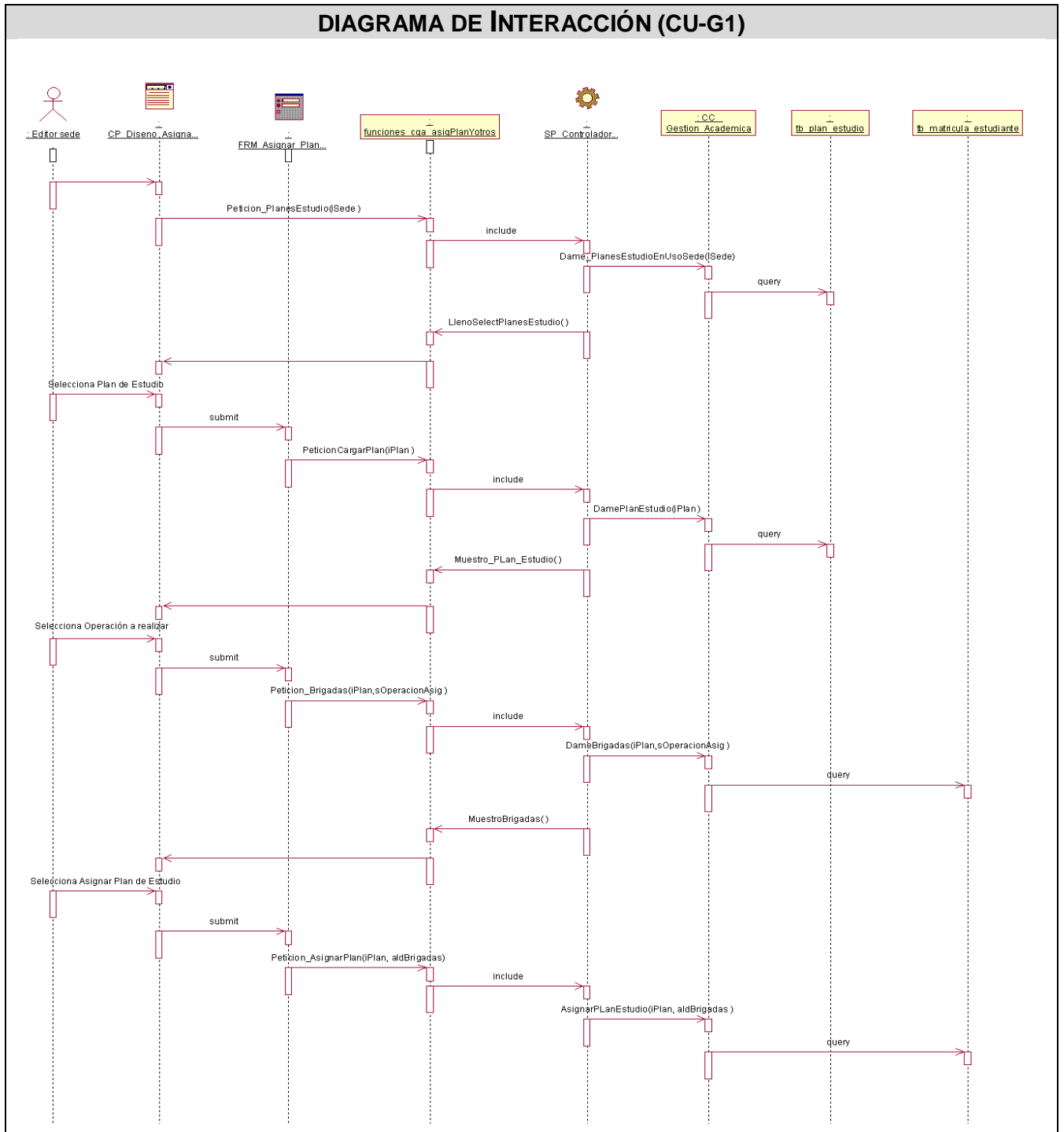


Fig 2 Diagrama de interacción del caso de uso Asignar Plan de Estudio

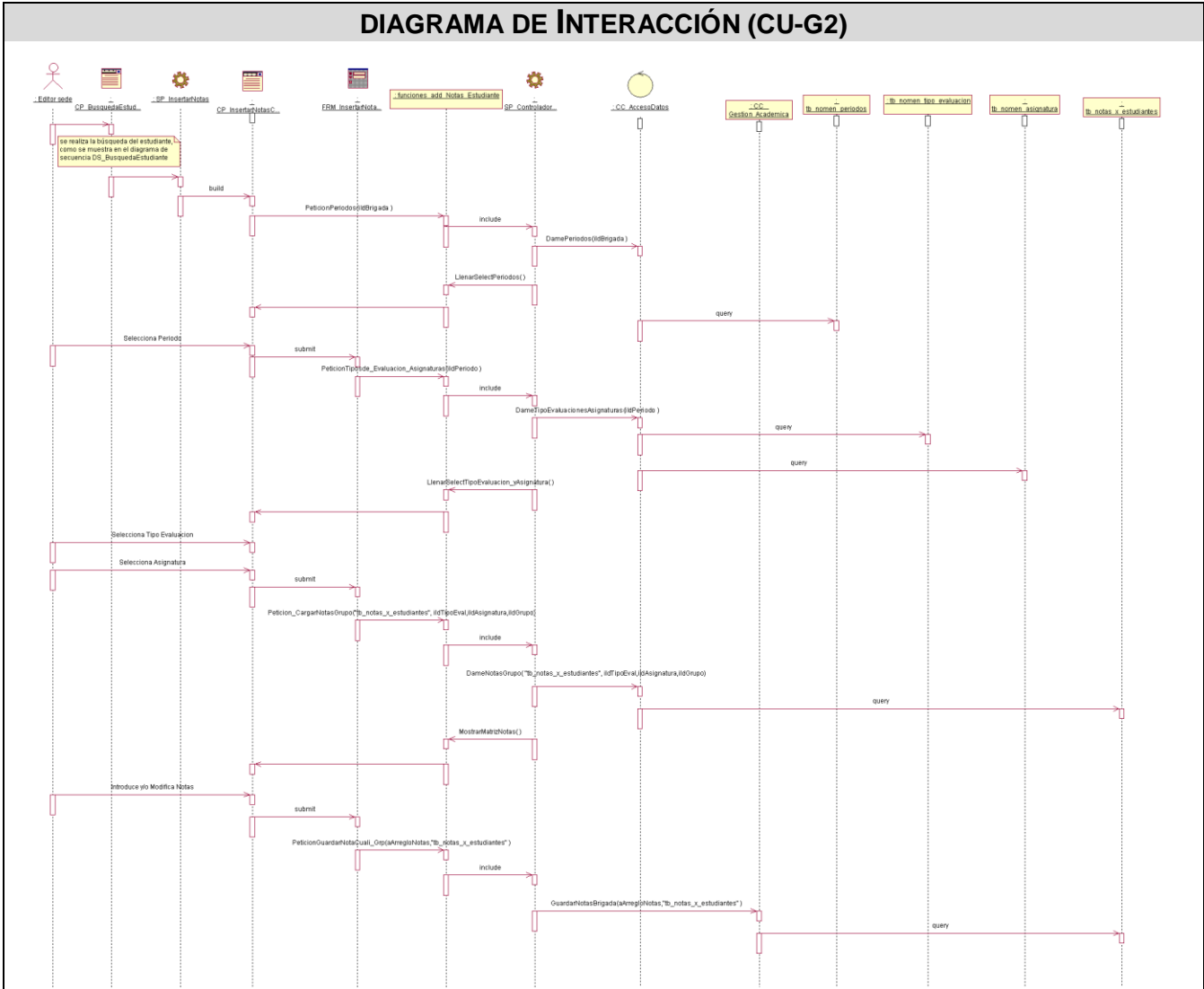


Fig 3 Diagrama de interacción del caso de uso Gestionar Evaluaciones, sección Insertar Evaluaciones Cuantitativas por estudiante.

DIAGRAMA DE INTERACCIÓN (CU-G2)

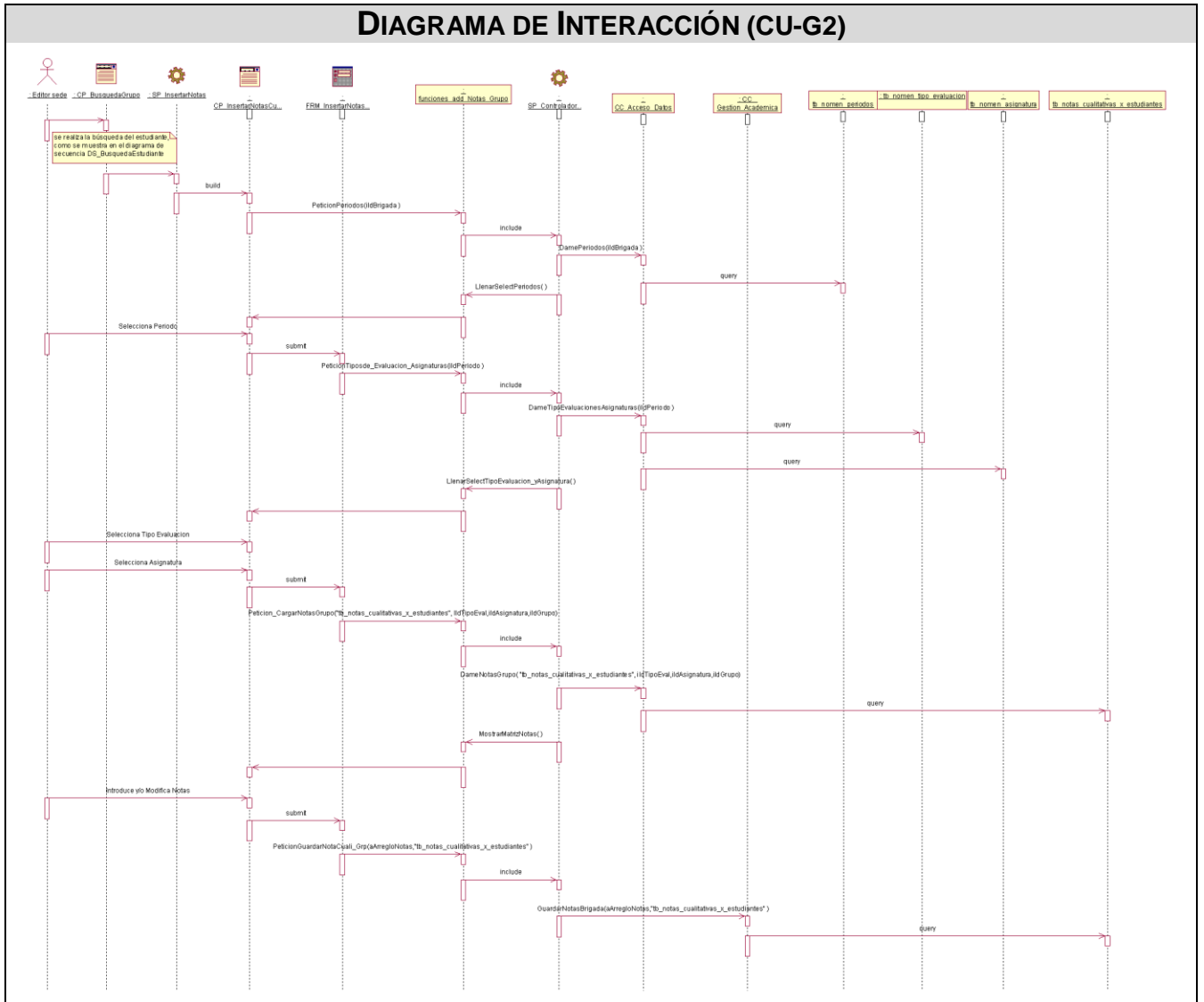


Fig 4 Diagrama de interacción del caso de uso Gestionar Evaluaciones, sección Insertar Evaluaciones Cualitativas por grupo.

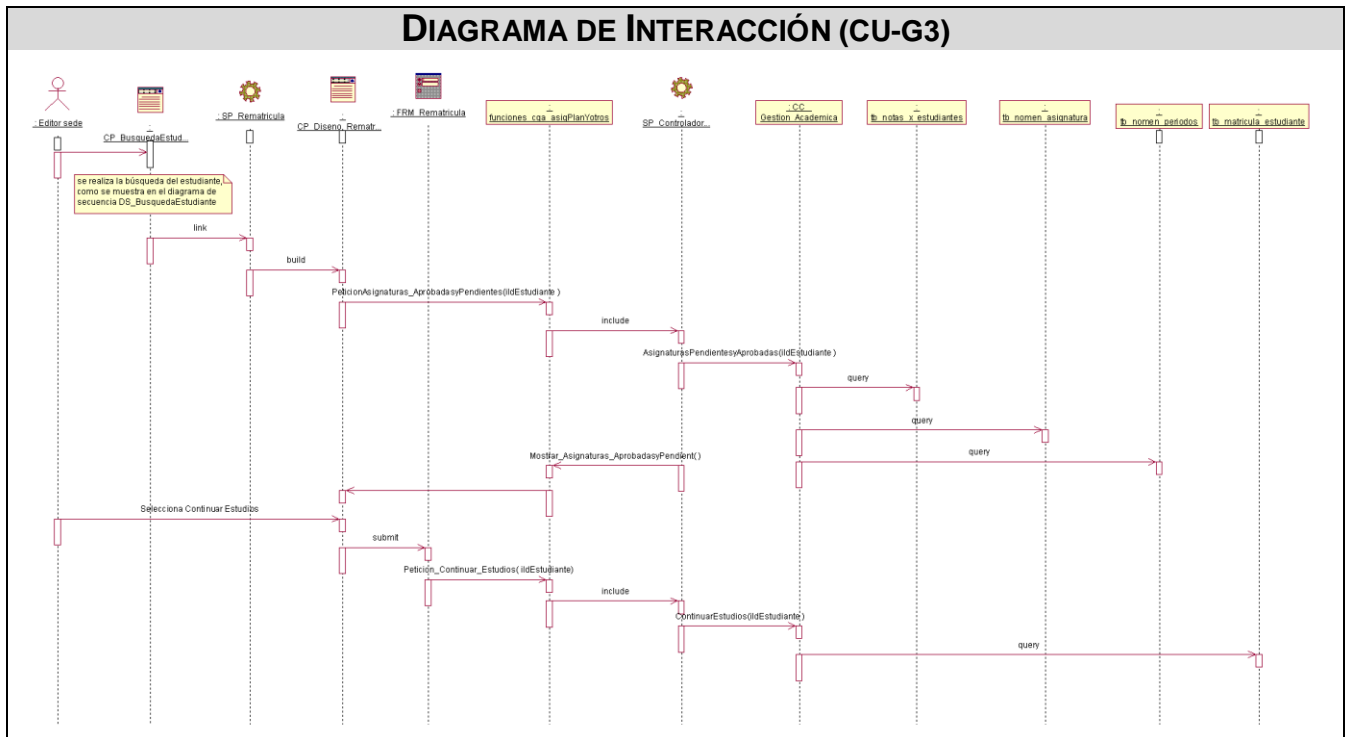


Fig 5 Diagrama de interacción del caso de uso Adelantar Asignatura.

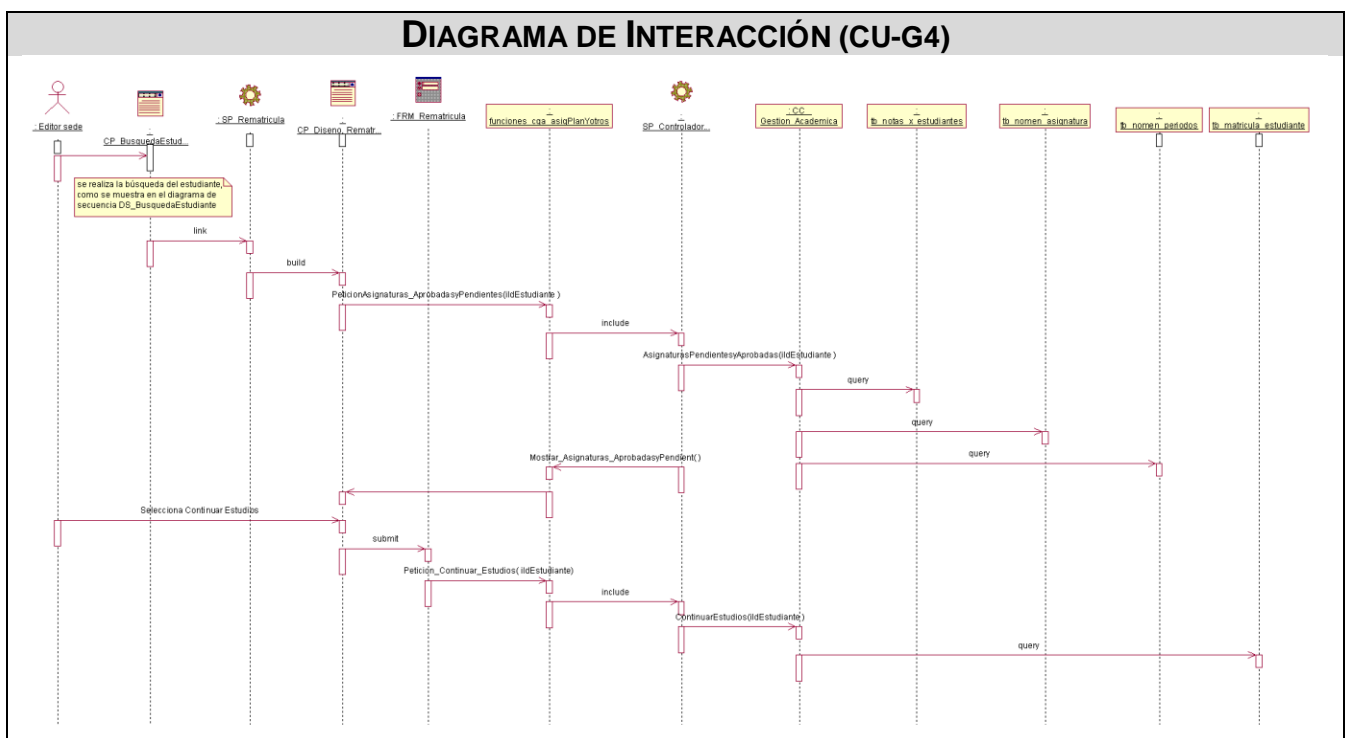


Fig 6 Diagrama de interacción del caso de uso Rematicular.

DIAGRAMA DE INTERACCIÓN (CU-G5)

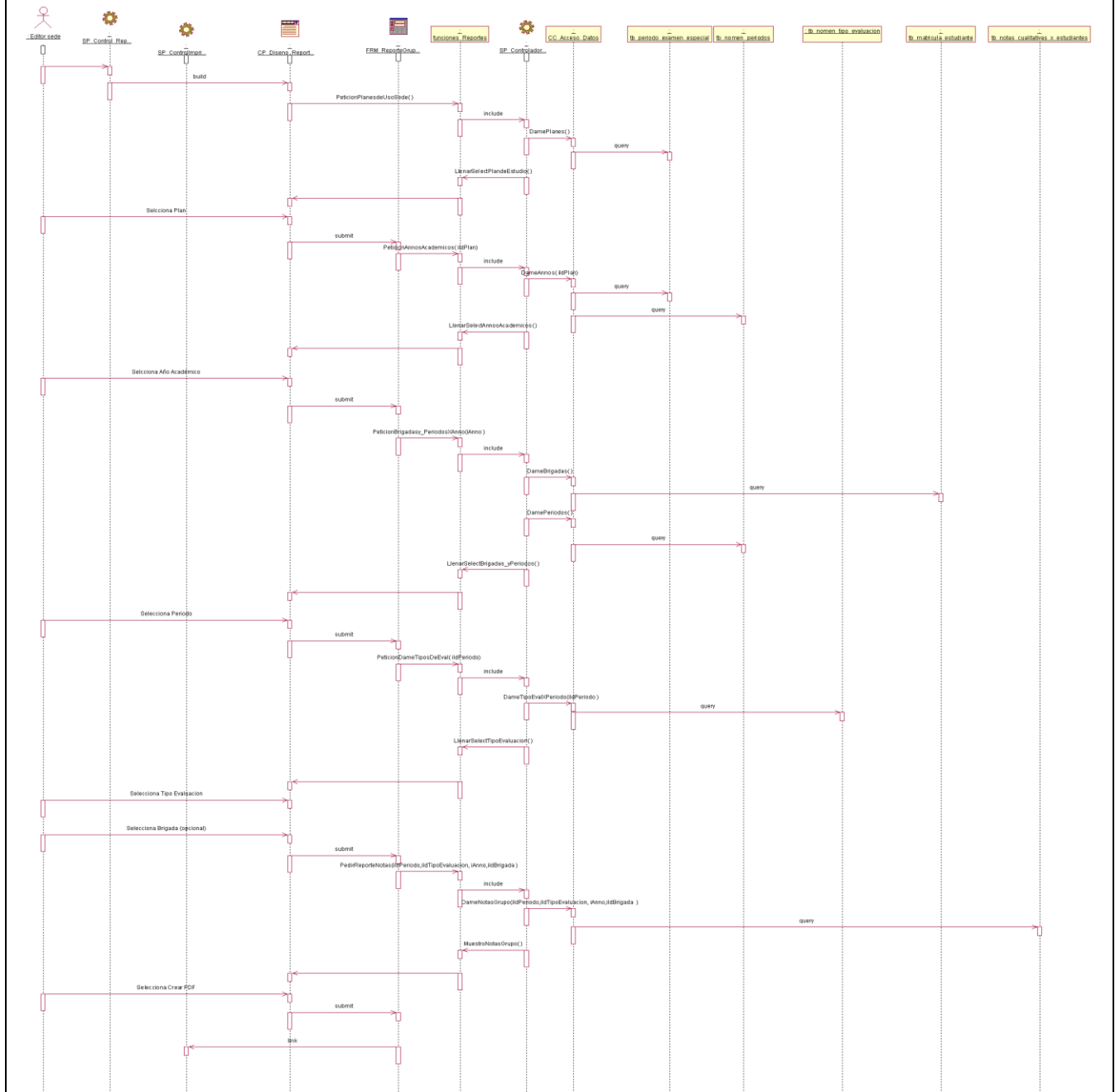


Fig 7 Diagrama de interacción del caso de uso Gestionar Reportes, sección Reporte de Corte Evaluativo por grupo.