

Universidad de las Ciencias Informáticas
“Facultad #7”



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

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

Autor(es): Reinier Alonso González.

Tutor(es): Ing. Yenisel Valido Pérez.

Co-tutor: Lic. Yasel Couce Sardiñas.

Ciudad de la Habana, Mayo del 2007.

AGRADECIMIENTOS.

A todos los que de una forma u otra han colaborado con mi formación profesional, especialmente a mis padres y mi familia por el apoyo que siempre me han dado.

A mi novia por su apoyo y paciencia.

A la dirección de la facultad por su preocupación por mis resultados.

Al grupo de estudiantes y profesores que trabajan en el desarrollo del proyecto.

A todos los colegas, que me han ayudado a lo largo de la carrera.

A la dirección de la revolución por darme la oportunidad de estudiar en una universidad como esta.

DEDICATORIA.

A mi familia y colegas...

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Grupo de Desarrollo del Área Temática de Sistemas de Apoyo a la Salud de la Facultad 7 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 25 días del mes de Mayo del año 2007.

Reinier Alonso González.

Ing. Yenisel Valido Pérez.

RESUMEN.

Con el desarrollo del presente trabajo se propone facilitar la gestión de información de las secretarías docentes en las instituciones clínico docentes cubanas, y lograr que toda esta se almacene en una base de datos central que pueda ser consultada desde cualquier punto del país mediante un servicio Web. El objetivo es crear un sistema informatizado que permita gestionar la información docente en el sector de la salud.

Resulta una tarea importante pues facilitará la toma de decisiones con respecto a los recursos humanos que se encuentran formándose en los centros docentes de salud de todo el país. Teniendo en cuenta que los software que se usan para gestionar esta información han perdido su funcionalidad, el diseño del software que proponemos es lo más flexible posible. La aplicación se desarrolla sobre tecnologías WEB, corriendo sobre un servidor LAMP (*Linux, Apache, MySQL, PHP*), utilizando herramientas de software libre.

Palabras Claves: Formación de Recurso Humanos, Sistema, Base de Datos.

ÍNDICE.

INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Sistemas existentes vinculados al campo de acción.....	5
1.2 Tendencias y tecnologías actuales.	6
1.2.1 Internet y su funcionamiento.....	7
1.2.2 Funcionamiento de la Web.....	7
1.2.3 Elementos de arquitectura.....	7
1.2.4 Definiciones de estilos de arquitectura.....	8
1.2.5 Arquitecturas en Capas.	8
1.2.6 La arquitectura Cliente/Servidor.	9
1.2.7 Servidor Web.....	9
1.2.8 Estilos de arquitecturas y patrones.....	11
1.2.9 ¿Por qué MVC?	11
1.2.10 Modelo Cliente Servidor de tres Capas (<i>Three Tier</i>).....	12
1.3 Metodología utilizada para el desarrollo de software.	13
1.3.1 UML	14
1.4 Sistema gestor de base de datos.	14
1.4.1 ¿Por que MySQL?	15
1.5 Lenguajes de programación utilizados.	16
1.5.1 Lenguajes utilizados del lado del cliente:	17
1.5.2 Lenguajes utilizados del lado del servidor:	18
1.6 ¿Por qué usar PHP y MySql?.....	21
1.7 Herramientas utilizadas en el desarrollo.	21
1.7.1 Dreamweaver 8.	21
1.7.2 Rational Suite 2003.	21
1.7.3 EMS 2005 y MySql Administrator.....	23
1.7.4 ZendStudio 5.2.0	23

1.7.5 Sistema de control de versión utilizado.	23
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.	26
2.1 Conceptos asociados al dominio del problema.	26
2.2 Objeto de estudio	26
2.2.1 Situación problemática.	26
2.2.2 Descripción de los procesos del negocio.	27
2.3 Propuesta de sistema.	29
2.4 Análisis comparativo de otras soluciones con la propuesta.	31
2.5 Modelo de negocio.	31
2.6 Especificación de los requisitos de software.	31
2.6.1 Requisitos Funcionales.....	31
2.6.2 Requisitos No Funcionales.....	36
2.7 Definición de los casos de uso del sistema.	38
CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA.	39
3.1 Análisis.	39
3.1.1 Paquete de Administración.....	40
3.1.2 Paquete Matrícula.....	42
3.2 Diseño.	45
3.2.1. Diagramas de clases de diseño.....	45
3.2.2 Descripción de las clases.	52
3.3 Diseño de la BD.....	79
3.3.1 Descripción de las tablas.....	80
3.4 Definiciones de diseño.	98
3.5 Tratamiento de errores.	99
3.6 Seguridad.	99
3.7 Estándar de codificación.	100
3.8 Concepción de la ayuda.	101
CAPÍTULO 4: IMPLEMENTACIÓN.	102

4.1 Diagrama de despliegue.....	102
4.2 Diagrama de componentes.	103
CONCLUSIONES.....	107
RECOMENDACIONES.	108
BIBLIOGRAFÍA.....	109
REFERENCIAS BIBLIOGRÁFICAS.	110
ANEXOS #1: MODELO DE NEGOCIO.....	112
ANEXOS #2: MODELO DE SISTEMA.	122
ANEXO # 3: MODELO DE DISEÑO.....	151

INTRODUCCIÓN.

En la actualidad, con el auge de los medios informáticos aparece el almacenamiento en soporte electromagnético, ofreciendo mayores posibilidades de almacenamiento, ocupando menos espacio y ahorrando un tiempo considerable en la búsqueda y tratamiento de los datos. Es en este momento donde surge el concepto de bases de datos, y con ellas las diferentes metodologías de diseño y tratamiento, surgen también arquitecturas de desarrollo que rompen los esquemas de pequeños sistemas, trayendo consigo el desarrollo de aplicaciones web que facilitan la comunicación y flujo de información. Estas tecnologías facilitan el desarrollo de sistemas para la gestión de grandes volúmenes de información como los que se manejan en los procesos relacionados con la actividad docente educativa de una institución, lo que se incluye dentro de las tareas de la formación de recursos humanos.

En la etapa actual del desarrollo de la Formación de Recursos Humanos de Salud en nuestro país, en el campo de la dirección y gestión, las organizaciones se ven sometidas a retos, desafíos y presiones a los cuales tienen que responder con alto grado de creatividad y realismo. Los principales retos están dados por la dinámica del proceso de formación de médicos y personal calificado en el área de la salud, la rápida aparición y aceptación de nuevos programas de estudios debido a la necesidad de cambiar los enfoques tradicionales. Este hecho redimensiona la relevancia de la formación de los recursos humanos pues constituye el centro de cualquier proceso de mejoramiento y eficiencia de la calidad y la productividad.

Como consecuencia de lo anteriormente planteado se tiende a mirar con ojos críticos los estilos de planificación, gestión y educación vigentes en los sistemas de servicios que son encargados de manejar y viabilizar la información referida con la formación de los recursos. Debido a esto aparecen nuevos enfoques de la planificación que valora la visión estratégica y la construcción participativa según la realidad, formando un estilo de gestión de personal que subordina lo administrativo y subraya la organización, las relaciones sociales e institucionales.

El proceso de Formación de Recursos Humanos tiene su centro básico en las secretarías docentes de cada una de las instituciones docentes del Ministerio de Salud Pública (*MINSAP*) ya que es esta la

base o nivel donde se conforma la información principal del proceso. Con este fin hasta el año 1999 se utilizó, y en algunos lugares aún se está utilizando el sistema SAIDO; 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.

A partir del año 2000 este sistema comenzó a presentar dificultades técnicas debido al conocido “efecto 2000”, además no se adapta a las transformaciones actuales de la enseñanza superior en el sector de la salud. A partir de esto se desarrollaron una serie de 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. Debido a esto, no se pueden integrar estas soluciones a un nivel central. Lo que imposibilita agrupar toda la información existente en un sólo sistema. Por esto se hace necesaria la realización de un nuevo sistema que se adapte a las condiciones actuales, y que contenga todas las especialidades médicas con las que ha crecido la Educación Médica Superior cubana en los últimos años.

Dada la situación anterior, el **problema** consiste en ¿Cómo facilitar el proceso de gestión de la información de las secretarías docentes y la toma de decisiones relacionada con ella en las instituciones clínico docentes del MINSAP?

El **objeto de estudio** se centra en el proceso de la gestión de información en las secretarías docentes del MINSAP, vinculado con el estudio de las herramientas y el proceso de desarrollo para el diseño de sistemas informáticos. El **campo de acción** apunta al proceso de gestión de la información relacionada con la matrícula y movimientos de estudiantes en las secretarías docentes de los centros educativos del MINSAP.

Para dar solución al problema antes mencionado se propone como **objetivo general**: Desarrollar una aplicación web para la gestión de la formación de recursos humanos en el Ministerio de Salud, de manera que facilite el desempeño y la toma de decisión de los trabajadores encargados de organizar y dirigir el proceso.

Donde las **tareas** propuestas para dar cumplimiento a los objetivos antes mencionados se definen como:

- Analizar los aspectos teóricos conceptuales en el proceso de formación del profesional de la salud, en específico, lo referente al curso de pregrado.
- Elaborar una fundamentación teórica de la aplicación para los procesos de gestión de la información del proceso de formación de recursos humanos en salud, orientados al control y gestión de información de las secretarías docentes de las instituciones clínico docentes del país.
- Realizar un estudio de las principales tendencias y tecnologías informáticas actuales para seleccionar el ambiente de desarrollo y arquitectura del sistema .
- Realizar el diseño e implementación de un sistema que permita gestionar la lógica del negocio del proceso de formación de los recursos humanos en salud.

En la investigación para dar cumplimiento a las tareas y objetivos planteados se utilizó la entrevista dentro de los métodos empíricos, como vía de obtención y elaboración de los datos y el conocimiento de los hechos fundamentales que caracterizan la situación actual, tales como: datos de entradas, tipos de reportes, restricciones de los procesos, etc.

El trabajo está estructurado por cuatro capítulos en los cuales se realiza el análisis y fundamentación de la solución propuesta que abarca el estudio de las tecnologías y herramientas usadas para el desarrollo de la aplicación, el modelado del negocio, el análisis y diseño del sistema y por último la implementación del mismo. A continuación se menciona el contenido tratado en cada capítulo:.

En el 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.

En el 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.

En el Capítulo 3 "Análisis y diseño del sistema" se realiza el análisis y diseño del sistema donde se muestran los diagramas de clases del análisis y del diseño, así como diagramas de interacción en el diseño, 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.

En el 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.

Este capítulo está dedicado a realizar un análisis detallado sobre el estado del arte del tema tratado, a nivel nacional. Además de investigar sobre los principales conceptos y tecnologías que pueden ser adecuados para el desarrollo del sistema. Teniendo en cuenta las necesidades y las características del entorno donde se aplicará la solución propuesta.

Se describen los sistemas integrados y distribuidos, el uso de las arquitecturas existentes y la metodología a utilizar para el análisis y diseño del sistema teniendo en cuenta las facilidades que puede aportar al trabajo. Se hace un estudio de algunos de los diferentes lenguajes de programación y de los sistemas gestores de base de datos (*SGBD*) más usados; definiéndose las tecnologías más adecuadas para el sistema.

1.1 Sistemas informáticos existentes en Cuba relacionados con las secretarías docentes.

El Ministerio de Salud Pública, como forma de retroalimentación del proceso docente educativo en las facultades, cuenta entre otros, con el sistema SAIDO, desarrollado por el Centro de Cibernética Aplicada a la Medicina. Este sistema es poco flexible con los nuevos modelos de enseñanza surgidos a partir de las transformaciones realizadas en la educación médica en los últimos tiempos. Razón por la cual, se han desarrollado distintas soluciones implementadas por las propias instituciones docentes con el apoyo de su personal, que por ser soluciones locales carecen de una integración, con el objetivo de lograr una base de datos única de donde se puedan obtener datos a distintos niveles de dirección.

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 dato 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 desarrollar al sistema tendrían como principal limitante que tendría que actualizarse en cada una de las secretarías docentes del país. Además que limita a funcionarios de otros niveles superiores de dirección el acceso a la información de forma real y actualizada.

Actualmente existen soluciones informáticas implementadas sobre la filosofía web que brindan la solución a las dificultades anteriormente mencionadas. Dentro de las experiencias existentes se encuentra el sistema AKADEMOS desarrollado y utilizado en la Universidad de las Ciencias Informáticas, el cual brinda una interfaz común para todos sus usuarios, además ofrece la información actualizada en tiempo real de las diez facultades con que cuenta la universidad.

Con el análisis realizado sobre los sistemas desarrollados con objetivos similares al sistema propuesto, se concluyó que no se utilizaría ningún componente desarrollado en ellos, producto al cambio del ambiente y filosofía de desarrollo como es el caso del SAIDO, pues reutilizar algún componente ya desarrollado comprendería una migración de tecnología que abarcaría mayor tiempo de desarrollo.

En el caso de AKADEMOS, a pesar de estar sobre tecnologías web no sería factible utilizarlo por las particularidades de la información de los estudiantes que se registran según los distintos modelos de formación además de la gestión académica que es diferente en varios aspectos. Otra cuestión fundamental, es que el sistema debe integrarse con otros ya desarrollados en ambientes de desarrollo distintos.

1.2 Tendencias y tecnologías actuales.

Constituye un objetivo fundamental para los diseñadores de software alcanzar un nivel técnico acorde con el desarrollo actual en la automatización de la información para la gestión de cualquier proceso a desarrollar. Por lo que es necesario hacer un estudio detallado de las tecnologías a utilizar y las posibilidades de desarrollo que estas brindan, así como los conceptos asociados a estas.

En la actualidad, la tendencia en el desarrollo de soluciones informáticas está dirigida a desarrollos web por las ventajas que trae una solución de este tipo. Por ejemplo, con sólo un cliente web cualquier usuario podría ver la aplicación independientemente de la versión del sistema operativo instalado en el cliente. Otra de sus ventajas más importantes es que en vez de crear clientes para Windows, Mac OS X, GNU/Linux, y otros sistemas operativos, la aplicación es escrita una vez y es mostrada en todos los clientes a través del acceso al servidor [1]. Por lo que las actualizaciones del sistema sólo se harían en el servidor lo cual sería transparente para los clientes, evitando posibles errores provocados por usuarios inexpertos que quisieran actualizar su aplicación.

A continuación en esta sección se describe los principales conceptos, tecnologías y herramientas propuestas para el desarrollo de la solución tratada en el trabajo.

1.2.1 Internet y su funcionamiento.

Desde el punto de vista técnico, se puede definir Internet como un inmenso conjunto de redes de ordenadores que se encuentran interconectadas entre sí. Los usuarios de cualquier ordenador en cualquiera de estas redes pueden utilizar las herramientas comunes, muchas veces las mismas que ya utilizan en su entorno local, para comunicarse con cualquier otro usuario o acceder a la información o recursos de otros ordenadores en otras redes conectadas en cualquier otra parte del mundo [2]. Para que todas estas computadoras puedan coexistir y comunicarse efectivamente entre sí, debe existir un camino físico que las una (*líneas telefónicas, conmutadas, redes digitales, enlaces satelitales, microondas, fibra óptica, cable coaxial, etc.*), además deben ponerse de acuerdo con la comunicación, es decir, usar el mismo protocolo de comunicación (*TCP/IP*) [3].

1.2.2 Funcionamiento de la Web.

Una vez que el usuario está conectado a Internet, tiene que instalar un programa capaz de acceder a páginas web, este programa que se usa para leer los documentos de hipertexto se llama "navegador", el "browser", "visualizador" o "cliente"[4]. Los más conocidos son el Explorer de Microsoft, Mozilla Firefox y el Netscape.

1.2.3 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 de software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones[5].

Una definición un poco más aclaratoria desde el punto de vista de ingeniería de software es: “ La organización fundamental de un sistema encarnada en sus componentes, las relaciones de los componentes con cada uno de los otros y con el entorno, y los principios que orientan su diseño y evolución” [6].

1.2.4 Definiciones de estilos de arquitectura.

Cuando se habla de una arquitectura determinada como por ejemplo arquitectura en tres capas, o una arquitectura cliente-servidor, u orientada a servicios, implícitamente se está haciendo referencia a un campo de posibilidades articuladoras, a una especie de taxonomía ("*ordenamiento*", "*norma*" o "*regla*") de las configuraciones posibles, en cuyo contexto la caracterización tipológica particular adquiere un significado distintivo [7].

1.2.5 Arquitecturas en Capas.

Los sistemas o arquitecturas en capas constituyen uno de los estilos que aparecen con mayor frecuencia mencionados como categorías mayores del catálogo, o, por el contrario, como una de las posibles encarnaciones de algún estilo más envolvente. La generalidad de las bibliografías definen el estilo en capas como una “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].

Las ventajas del estilo en capas son obvias. Primero que nada, 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.

1.2.6 La arquitectura Cliente/Servidor.

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes [9].

Dentro de las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.
- 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.
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado.
- Se reduce el tráfico de red considerablemente. Idealmente, el cliente se comunica con el servidor utilizando un protocolo de alto nivel de abstracción.

1.2.7 Servidor Web.

Un servidor web es un programa que implementa el *protocolo HTTP (hypertext transfer protocol)*. Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (*hypertext markup language*): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos [10].

Cabe destacar el hecho de que la palabra servidor identifica tanto al programa como a la máquina en la que dicho programa se ejecuta, en este caso nos referimos al programa.

Algunos de los servidores web más importantes y conocidos son:

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

En la actualidad uno de los más usados es el Apache. Su coste gratuito, gran fiabilidad y extensibilidad le convierten en una herramienta potente y muy configurable.

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

Apache fue hecho para proveer un alto grado de calidad y fortaleza para las implementaciones que utilizan el protocolo HTTP. Está ligado a la plataforma Linux, Windows, UNIX y es un software libre, porque sus desarrolladores defienden la teoría de que las transmisiones usando la red deben estar en las manos de todos, y que las compañías de software deben hacer el dinero ofertando servicios con valor añadido tales como módulos especializados, soportes, entre otros, y no siendo dueñas de un protocolo. Así, el proyecto de crear una implementación robusta con referencia absolutamente libre para quien lo quiera usar es un buen paso para evitar la propiedad sobre los protocolos [11]. Por todo lo expuesto anteriormente es el servidor web versión 2.0.55 que se utiliza en la aplicación que se presenta.

1.2.8 Estilos de arquitecturas y patrones.

Los patrones 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.

La dinámica incontenible de la producción de patrones en la práctica de la arquitectura de software, hace que todo el mundo acepte que existen diversas clases de patrones: de análisis, de arquitectura (*divididos en progresivamente estructurales, sistemas distribuidos, sistemas interactivos, sistemas adaptables*), de diseño (*conductuales, creacionales, estructurales*), de organización o proceso, de programación y los llamados idiomas, entre otros [12].

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. Dentro de las soluciones estudiadas el patrón más recomendado por especialistas para aplicaciones como la propuesta en el trabajo es el Modelo Vista Controlador (*MVC*) [13].

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.

1.2.9 ¿Por qué MVC?

Un propósito común en numerosos sistemas es el de tomar datos de un almacenamiento y mostrarlos al usuario. Luego que el usuario introduce modificaciones, las mismas se reflejan en el

almacenamiento. El flujo de información ocurrirá entre el almacenamiento y la interfaz, una tentación común, un impulso espontáneo (*hoy se llamaría un anti-patrón*) es unir ambas piezas para reducir la cantidad de código y optimizar la performance. Sin embargo, esta idea es antagónica al hecho de que la interfaz suele cambiar, o acostumbra depender de distintas clases de dispositivos (*clientes ricos, browsers, PDAs*); la programación de interfaces de HTML, además, requiere habilidades muy distintas de la programación de lógica de negocios. Otro problema es que las aplicaciones tienden a incorporar lógica de negocios que van más allá de la transmisión de datos.

Resumiendo lo anterior aplicando este patrón se logra separar el modelo de negocios de la presentación usada, así la información podrá ser vista en múltiples formatos. Esta separación permite diversas vistas para una misma lógica de negocios.

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

Es una realidad que el objetivo de las aplicaciones modernas no es sólo procesar transacciones y generar reportes, sino el de que sean sistemas de información robustos y que puedan cambiar conforme cambien las necesidades de la empresa o institución. Mientras las aplicaciones se utilicen en el ámbito y edificio de una empresa o institución de regular tamaño, el modelo cliente servidor de dos capas donde se tiene un servidor de aplicación que contiene mezclada la lógica del negocio con la interfaz que le brinda al usuario y un servidor de bases de datos pudiera ser adecuado. Pero cuando la aplicación rebasa los límites físicos de un edificio y se vuelve estatal o mundial, con numerosos clientes distribuidos en cualquier parte, o cuando la aplicación es complicada, el modelo de dos capas se hace insuficiente [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. Este tipo de arquitectura es la más compleja.

En esta arquitectura todas las peticiones de los clientes se controlan en la capa correspondiente a la lógica del negocio. Cuando el cliente necesita hacer una petición se la hace a la capa en la que se encuentra la lógica del negocio. Esto es bastante importante pues eso quiere decir que:

- El cliente no tiene que tener drivers ODBC ni la problemática consiguiente de instalación de los drivers por tanto se reduce el costo de mantener las aplicaciones cliente.
- El cliente y el Gestor de Reglas de negocio tienen que hablar el mismo lenguaje.
- El Gestor de Reglas de Negocio y el Servidor de Datos tienen que hablar el mismo lenguaje (ODBC).

1.3 Metodología utilizada para el desarrollo de software.

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no llevamos una metodología de por medio, lo que obtenemos es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Sin embargo, muchas veces no se toma en cuenta el utilizar una metodología adecuada, cuando el proyecto que se va a desarrollar es de gran envergadura, toma un sentido importante el basarse en una metodología de desarrollo que responda a las necesidades de desarrollo. En la actualidad existen diversas metodologías de desarrollo con particularidades específicas cada una de ellas, dentro de las más conocidas están: RUP, XP y MSF. Por las particularidades de la solución que se desarrolla se propone el uso de RUP.

El Proceso Unificado de Desarrollo o RUP (*Rational Unified Process*) es un proceso de desarrollo de software (*conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software*) que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de amplitud y diferentes tamaños de proyectos.

El Proceso Unificado está basado en componentes y utiliza el UML (*Lenguaje Unificado de Modelado*) para preparar todos los esquemas de un sistema de software. De hecho, UML es una parte esencial de RUP, sus desarrollos fueron paralelos. No obstante los verdaderos aspectos definitorios del proceso unificado se resumen en tres fases 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.

1.3.1 UML

El éxito de los proyectos de desarrollo de aplicaciones o sistemas se debe a que sirven como enlace entre quien tiene la idea y el desarrollador. El UML (*Lenguaje Unificado de Modelado*) es una herramienta que cumple con esta función, ya que le ayuda a capturar la idea de un sistema para comunicarla posteriormente a quien esté involucrado en su proceso de desarrollo; esto se lleva a cabo mediante un conjunto de símbolos y diagramas. UML se ha convertido en el estándar de facto para definir, organizar y visualizar los elementos que configuran la arquitectura de una aplicación.

UML (*Unified Modeling Language*) o Lenguaje de Modelación Unificado es un lenguaje gráfico para especificar, construir, visualizar y documentar las partes o artefactos. (*información que se utiliza o produce mediante un proceso de software*). Pueden ser artefactos: un modelo, una descripción que comprende el desarrollo de software, utilizándose también en el diseño Web.

De forma general las principales características son:

- Lenguaje unificado para la modelación de sistemas.
- Corrección de errores viables en todas las etapas.
- Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.
- Los modelos permiten la comunicación con el cliente en todas las etapas.

Además UML es desde finales de 1997, un lenguaje de modelado orientado a objetos estándar, de acuerdo con el Object Management Group, siendo utilizado diariamente por grandes organizaciones como: Microsoft, Oracle, Rational, etc.

1.4 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] [16]:

- **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 darnos la información solicitada y en almacenar los cambios realizados.

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

1.4.1 ¿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. MySQL 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 MySQL 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

- 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.

1.5 Lenguajes de programación utilizados.

Los lenguajes de programación son creados por el hombre para poder comunicarse con las computadoras. De esta forma 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 importante que son. 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 mas 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.

1.5.1 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 [17].

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 [18].

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. Éstas 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 [19].

AJAX es una combinación de tres tecnologías ya existentes:

- **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.

1.5.2 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 [20].

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 podemos usarlo 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.

1.5.3 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 [21].

Algunas de las características de Smarty:

- 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

1.6 ¿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.

1.7 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:

1.7.1 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.

1.7.2 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 [22].

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.

1.7.3 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.

1.7.4 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.

1.7.5 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 mas 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 [23].

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 [24].

Conclusiones.

En este capítulo se analizan y fundamentan los conceptos, tecnologías y metodología que son utilizadas para el desarrollo del sistema propuesto. Definiendo como entorno de desarrollo la tecnología LAMP (SO *Linux*, Servidor Web *Apache*, Gestor de BD *MySQL* y lenguaje de programación *PHP*) utilizando el motor de plantilla Smarty y tecnología AJAX.

Se definió una arquitectura en tres capas según el modelo Cliente Servidor, implementando el patrón Modelo Vista Controlador, se define RUP (*Proceso de desarrollo unificado*) como metodología de desarrollo. Se realizó un estudio sobre los antecedentes de los sistemas que fueron creados con el propósito de dar solución a la problemática trabajada.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

Este capítulo contiene el resultado de la búsqueda y análisis de la información vinculada al objeto de estudio, procesos a automatizar y conceptos asociados al dominio del sistema. Además se define la situación problemática, objetivos generales y se hace una descripción del sistema propuesto.

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.

Expediente Académico: Documentos que acreditan la estancia del estudiante en la institución docente donde se recogen los resultados docentes y no docentes alcanzados por un estudiante y debe ser conservado en el centro de enseñanza por tiempo indefinido.

Bajas: Son las acciones que se llevan a cabo para separar o desactivar a un estudiante de una carrera de Ciencias Médicas.

Licencia de Matrícula del Estudiante: Son las acciones que se llevan a cabo para autorizar a un estudiante de una carrera de Ciencias Médicas a no asistir a clases por un periodo determinado de tiempo.

Altas: Son las acciones que se llevan a cabo para incorporar un estudiante a una carrera de Ciencias Médicas.

Matrícula: Materializar la incorporación de un aspirante al Sistema de Educación.

2.2 Objeto de estudio

2.2.1 Situación problemática.

En Cuba actualmente la gestión de información en la formación de los recursos humanos del Ministerio de Salud, y específicamente de sus instituciones clínico docentes se hace manualmente o por sistemas no profesionales de tecnología obsoleta, con bajas prestaciones, que fueron desarrolladas sin un

continuo mejoramiento. Esto conlleva a que el nivel de errores y demora en la entrega de los datos para la toma de decisiones a todas las instancias de dirección sean considerables.

Este proceso se desarrolla en las secretarías docentes de las instituciones educativas de esta esfera, donde se llevan a cabo los diferentes procesos en la formación curricular de un estudiante, la clasificación de toda la información que contiene el trabajo de secretaría por perfiles, matrícula y ratificación de la misma, la confección de las salidas estadísticas, el registro y control de las evaluaciones de los estudiantes por año, y toda la actividad docente de forma general, lo cual se incluye dentro de las tareas de gestión académica.

2.2.2 Descripción de los procesos del negocio.

Proceso de control de altas.

Dentro del calificativo de altas se encuentra la matrícula y ratificación de la misma. Este proceso comienza desde que el estudiante de nuevo ingreso se presenta en la secretaria docente para brindar los datos que necesita la institución. Se recoge una gran cantidad de información de orden personal y docente, con el objetivo de tener una caracterización del estudiante. Una vez recogida toda la información la principal dificultad es el procesado y actualización de la información para darle una buena utilidad en los diferentes fines para la vida de la institución y del país.

Para los estudiantes continuantes se realiza una ratificación de la matrícula, donde se le actualizan una parte de los datos que se le recogieron en la matrícula inicial y conocer su disposición para continuar los estudios en el centro y en la carrera. Este proceso hace demasiado engorroso si la información está archivada en papel o en algún tipo de soporte donde se haga difícil la búsqueda de los datos del estudiante.

Proceso de control de bajas, licencias de matrícula y traslados.

Dentro del calificativo de bajas por las diferentes causas estipuladas por el Ministerio de Educación Superior, se encuentran causas como bajas académicas, bajas por deserción, bajas por inasistencia, bajas por sanción disciplinaria, bajas voluntarias, entre otras.

El proceso de licencias de matrícula es la posibilidad que se le concede a aquellos estudiantes que estando matriculados en la carrera se vean en la imperiosa necesidad de interrumpir transitoriamente sus estudios en el curso académico en que se encuentre, debido a las causas establecidas en la Resolución 15 del 88 de los CEMS.

Por último se encuentra la gestión de los traslados o movimientos de estudiantes ya sea interno en el centro o hacia otro centro de educación, procesos que actualmente se hacen de forma manual lo que implica un mayor tiempo de trámites.

 Proceso de confección de las salidas estadísticas.

En cada uno de los centros docentes la secretaria docente tiene que realizar una serie de reportes estadísticos de forma periódica, sobre la docencia y la matrícula real que tienen en sus instalaciones. Estos reportes miden una serie de parámetros con un formato estándar que resulta muy engorroso por la cantidad de información que hay que manejar para realizarlos. Una vez terminado el informe o reporte este se entrega al organismo superior que lo atiende, que está en dependencia del tipo de centro que sea, el que se encarga de verificar la información hasta que esté correcta.

Por la dinámica que tiene el proceso de formación de recursos humanos en salud, que se ha convertido por las características de la Revolución en un proyecto donde estudian jóvenes de diferentes países de Latinoamérica y de otros continentes, se hace necesaria otra serie de información de forma inmediata que no tienen parámetros de búsqueda fijos lo cual se hace mucho más engorroso debido a la cantidad de documentación que hay que manejar para poder darle cumplimiento a la entrega de la información solicitada.

 Proceso de registro y control de las evaluaciones de los estudiantes.

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.

Confección del expediente académico.

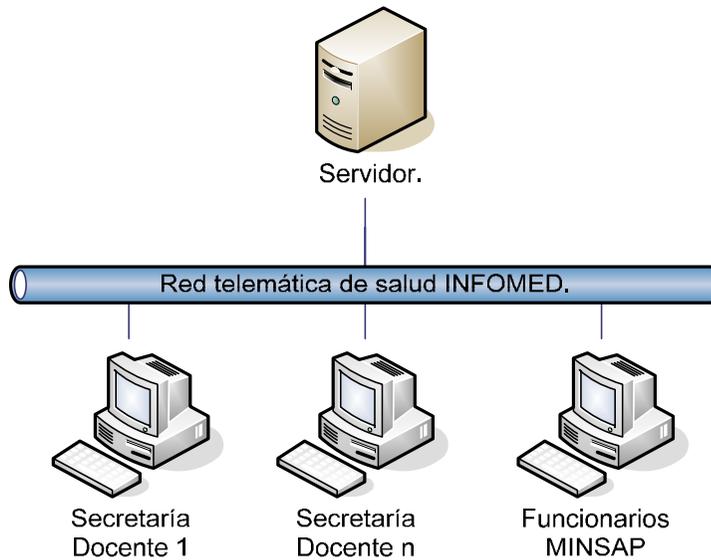
Conservar en copia dura los documentos que acreditan la estancia del estudiante en la institución y los resultados docentes y no docentes alcanzados en ella. El expediente es un documento oficial en el cual se registran: Los datos personales, las calificaciones obtenidas en las evaluaciones finales, ordinaria y extraordinaria; las evaluaciones anuales derivadas de la participación en el Proyecto Educativo de cada año; las distinciones que le han sido otorgadas; otros aspectos que permitan valorar el progreso del estudiante en su preparación; y los resultados del resto de las actividades del proceso docente educativo.

En el caso de los estudiantes extranjeros se registra un grupo de información que no se le registra a los estudiantes cubanos, como es el caso de la realización de un diagnóstico del estado de salud, donde se le realizan una serie de análisis y diagnósticos médicos que forman parte de la caracterización del estudiante, también se encuentra el curso de premédica, que consiste en un curso de nivelación sobre los conocimientos básicos que debe tener un estudiante para ingresar a la educación superior. Este curso tiene características muy variables ya que está en dependencia de donde provenga el grupo de estudiantes que recibirá el curso, por lo que su plan de estudio es variable.

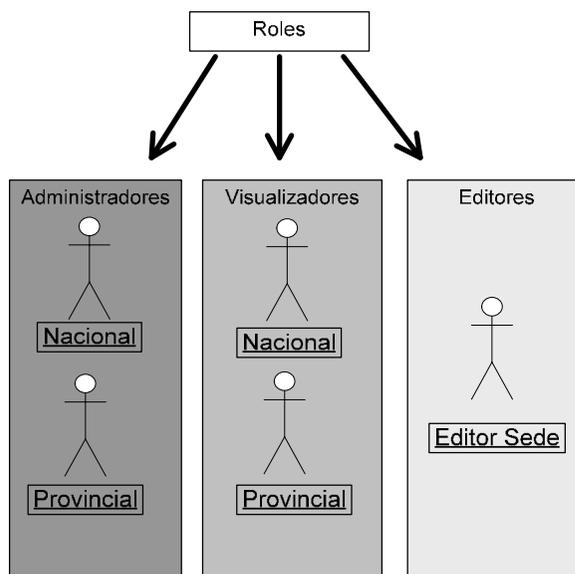
2.3 Propuesta de sistema.

El sistema que se propone en el trabajo se encontrará brindando servicio sobre la red telemática de salud cubana INFOMED, a través de ella todos los nodos (*secretarías docentes, puestos de trabajos relacionados con el sistema*) accederán al sistema, que contará con un módulo de administración que brindará flexibilidad ante los aspectos relacionados con la configuración, el cual dará la posibilidad de

configurar una máquina servidor en el nivel que el usuario necesite, previniendo así los problemas de conectividad. A continuación se muestra un esquema sobre cómo funcionará la aplicación.



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



2.4 Análisis comparativo de otras soluciones con la propuesta.

Después de haber realizado un estudio de algunas de las soluciones que actualmente se encuentran funcionales, con el objetivo de determinar el estado actual sobre el objeto de estudio del trabajo, se concluye, que es necesario implementar un sistema centralizado para garantizar el proceso de gestión y control relacionado con la formación de los recursos humanos de salud en el país.

Con este objetivo la solución propuesta cuenta con tecnologías actuales desarrolladas sobre la web, lo que posibilita a los diferentes niveles de dirección el control y consulta de la información registrada en cualquier centro clínico docente ubicado en cualquier lugar del país, y la obtención de los indicadores estadísticos establecidos en tiempo real, posibilidad que en la actualidad los sistemas existentes no brindan debido a que son soluciones locales que carecen de una integración a nivel nacional, además de que sistemas de este tipo no contribuirían a la gran plataforma informatizada de salud que se propone desarrollar el país.

El sistema se integrará con otros sistemas que formarán parte de la plataforma informatizada que se desarrolla en el Ministerio de Salud Pública cubano, utilizando arquitecturas orientadas a servicios contando con tecnologías novedosas.

2.5 Modelo de negocio.

El primer paso en el proceso de desarrollo de software es precisamente alcanzar cierto nivel de conocimientos sobre el problema en cuestión. Para lograr esos propósitos, el proceso de modelado de negocio permite obtener una visión de la organización que permita definir los procesos, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos desarrollados en el [Anexo 1](#).

2.6 Especificación de los requisitos de software.

2.6.1 Requisitos Funcionales.

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

R1.Registrar datos del estudiante (Planilla de matrícula: datos generales, personales, familiares y de la carrera).

1.1. Insertar datos del estudiante.

R2.Actualizar datos de la planilla de matrícula de un estudiante.

R3.Buscar un estudiante

3.1. Realizar búsqueda de un estudiante según su código de identificación.

3.2. Mostrar datos del estudiante buscado.

3.3. Realizar búsqueda avanzada de un estudiante.

3.4. Mostrar listado de estudiantes que cumplan con el criterio de búsqueda.

R4.Buscar estudiantes de un grupo.

4.1. Mostrar listado de estudiantes de un grupo buscado.

R5.Administrar usuarios del sistema.

5.1. Mostrar listado de usuarios.

5.2. Modificar datos de un usuario registrado.

5.3. Cambiar contraseña de un usuario registrado.

5.4. Eliminar un usuario existente.

5.5. Registrar un nuevo usuario.

5.6. Asignar rol y nivel de acceso al nuevo usuario a registrar.

R6. Obtener historial de accesos al sistema.

6.1. Buscar todos los accesos.

6.2. Buscar accesos por usuario y fecha de acceso.

6.3. Mostrar, según la búsqueda realizada, usuario, páginas a las que accedió, fecha, hora e IP de la máquina desde la cual realizó el acceso.

R7.Autenticar usuario al sistema.

7.1. Proteger información de accesos no autorizados: mostrar información según nivel de acceso.

R8. Gestionar datos de codificadores.

8.1. Registrar y modificar datos de un codificador.

8.2. Eliminar datos de un codificador.

8.3. Listar todos los codificadores existentes.

R9. Gestionar notas del curso académico.

9.1. Registrar y modificar notas, según el tipo de evaluación, de las asignaturas de cada semestre de la carrera para un estudiante buscado.

9.2. Registrar y modificar notas, según el tipo de evaluación, de las asignaturas de cada semestre de la carrera para los estudiantes de un grupo buscado.

R10. Mostrar resultados docentes de un estudiante.

10.1. Calcular índice académico de cada semestre.

10.2. Calcular índice académico general del año que cursa.

10.3. Calcular índice académico acumulado de un estudiante durante los años académicos que ha cursado.

R11. Mostrar resultados docentes de los estudiantes de un grupo.

R12. Registrar y actualizar ubicación docente y datos del curso académico de la carrera para los estudiantes de un grupo.

R13. Mantener actualizados y llevar un registro de los movimientos que realice un estudiante durante su formación: bajas, traslados, incorporaciones, licencias de matrícula.

R14. Mantener actualizada la existencia en secretaría docente de los documentos que se le solicitan al estudiante para su matrícula: título de bachiller, certificación de notas, certificación o partida de nacimiento, etc.

R15. Generar listado de estudiantes configurado por el usuario.

15.1. El usuario debe indicar alguno de los siguientes criterios de búsqueda: país; estado, departamento o provincia de procedencia; provincia en Cuba; centro de estudio; grupo en la carrera. Además debe indicar los criterios que desea mostrar para conformar el listado.

15.2. Mostrar listado de estudiantes con las características indicadas según los criterios seleccionados.

Resúmenes

R16. Generar resumen de estudiantes por país, año académico y ubicación en Cuba.

16.1. El usuario debe indicar la provincia.

16.2. Mostrar total de estudiantes por países en cada año de la carrera.

R17. Generar resumen de estudiantes por vía de ingreso y país, provincia y centro de estudios.

17.1. El usuario debe indicar provincia y centro de estudios.

17.2. Mostrar total de estudiantes de cada vía de ingreso por país.

R18. Generar resumen de estudiantes por edad, sexo y provincia.

18.1. El usuario debe indicar la provincia.

18.2. Mostrar total de estudiantes por sexo y grupos etarios (- 17, 17, 18, 19, 20, 21-25, 26-30, 31-35, +35).

R19. Generar resumen de estudiantes en la Carrera por provincia, municipio y centro de estudios.

19.1. El usuario debe indicar provincia y municipio.

19.2. Mostrar total de estudiantes en Premédico y en la Carrera por centros de estudios.

R20. Generar resumen de estudiantes por fecha de matrícula, provincia y centro de estudios.

20.1. El usuario debe indicar provincia y año.

20.2. Mostrar total de estudiantes por mes de matrícula y centro de estudios.

- R21. Generar resumen de estudiantes por estado civil, provincia y centro de estudios.
- 21.1. El usuario debe indicar la provincia
 - 21.2. Mostrar total de estudiantes por estado civil y centro de estudios.
- R22. Generar resumen de estudiantes por color de la piel, provincia y centro de estudios.
- 22.1. El usuario debe indicar la provincia.
 - 22.2. Mostrar total de estudiantes por color de piel y centro de estudios.
- R23. Generar resumen de estudiantes por la cantidad de hijos.
- 23.1. El usuario debe indicar la provincia.
 - 23.2. Mostrar total de estudiantes que poseen hijos por cantidad de hijos y centro de estudios.
- R24. Generar resumen de estudiantes por etnias, provincia y centro de estudios
- 24.1. El usuario debe indicar la provincia.
 - 24.2. Mostrar total de estudiantes por etnia y centro de estudios.
- R25. Generar resumen de estudiantes indígenas por provincia y centro de estudios
- 25.1. El usuario debe indicar la provincia.
 - 25.2. Mostrar total de estudiantes indígenas de cada centro.
- R26. Generar resumen de estudiantes creyentes y no creyentes, por provincia y centro de estudios.
- 26.1. El usuario debe indicar la provincia.
 - 26.2. Mostrar total de estudiantes creyentes y no creyentes por centro de estudio.
- R27. Generar resumen de estudiantes por talla y sexo, provincia y centro de estudios.
- 27.1. El usuario debe indicar la provincia y el centro de estudios.
 - 27.2. Mostrar total de estudiantes femeninos y masculinos por tallas para cada tipo de vestuario.

R28. Generar resumen de estudiantes por movimientos políticos, país, provincia y centro de estudios.

28.1. El usuario debe indicar la provincia y el centro de estudios.

28.2. Mostrar total de estudiantes por movimientos políticos y países.

R29. Generar resumen de estudiantes por procedencia social, país, provincia y centro de estudios

29.1. El usuario debe indicar la provincia y el centro de estudios.

29.2. Mostrar total de estudiantes por países y procedencia social.

2.6.2 Requisitos No Funcionales.

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

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.

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.

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.

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.

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.

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, de manera que cada uno tenga datos de acceso propios. Se crearán usuarios con diferentes niveles de acceso al sistema.

Se usarán mecanismos de encriptación de los datos que por cuestiones de seguridad no deben viajar al servidor en texto claro, como es el caso de las contraseñas. Se guardará encriptada esta información en la base de datos utilizando para ello MD5 como algoritmo de encriptación.

Se harán validaciones de la información tanto en el cliente como en el servidor.

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.

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.

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)

2.7 Definición de los casos de uso del sistema.

Un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores. En el [Anexo 2](#) se encuentra descrito el modelamiento 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.

Después de analizar las características del sistema propuesto se evidencia la necesidad de crear una aplicación que integre y facilite el proceso, por lo que se propone un sistema basado en la gestión de base de datos y tecnología Web que de solución a la problemática trabajada.

Se realiza el análisis del negocio que implementa el sistema generando todos los artefactos que propone RUP en el flujo de trabajo de Modelamiento del Negocio, Requerimientos. Así como la descripción de cada uno de los casos de uso, priorizándose aquellos a implementar en el primer ciclo de desarrollo.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA.

Tras la definición y descripción, en el anterior capítulo, de las funcionalidades deseadas y necesarias del sistema propuesto; se hace necesario definir cómo se desarrollará. Este capítulo está dedicado a la etapa de desarrollo donde se profundiza en el análisis del sistema, utilizando un lenguaje apropiado como UML para el entendimiento del mismo, dando paso posteriormente al diseño del sistema, etapa en la cual ocurre un refinamiento del análisis realizado.

En esta etapa se definen diagramas de clases del análisis de la aplicación, especificando qué clases de análisis toman parte del caso de uso y las relaciones entre ellas clasificándolas en clases interfaz, clases controladoras y clases entidad. Posteriormente en la etapa de diseño a través de los diagramas de interacción se muestra una descripción gráfica de la interacción entre los actores y el sistema. Además se describen 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.

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

3.1.1 Paquete de Administración.

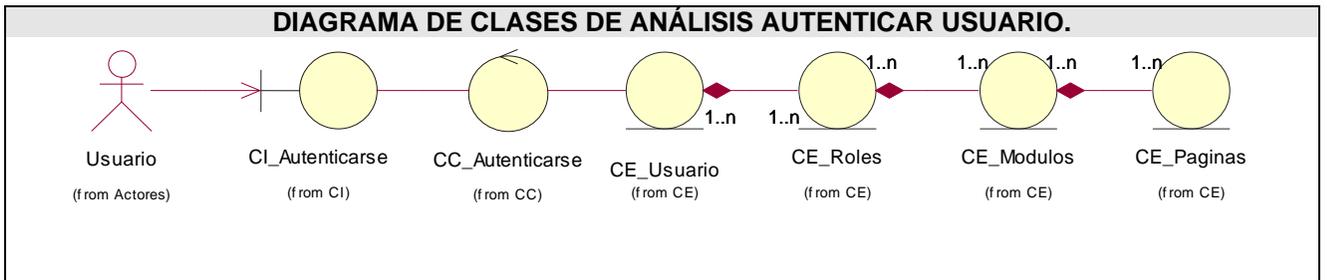


Fig. 1 Diagrama de clases de análisis CU Autenticas Usuario.

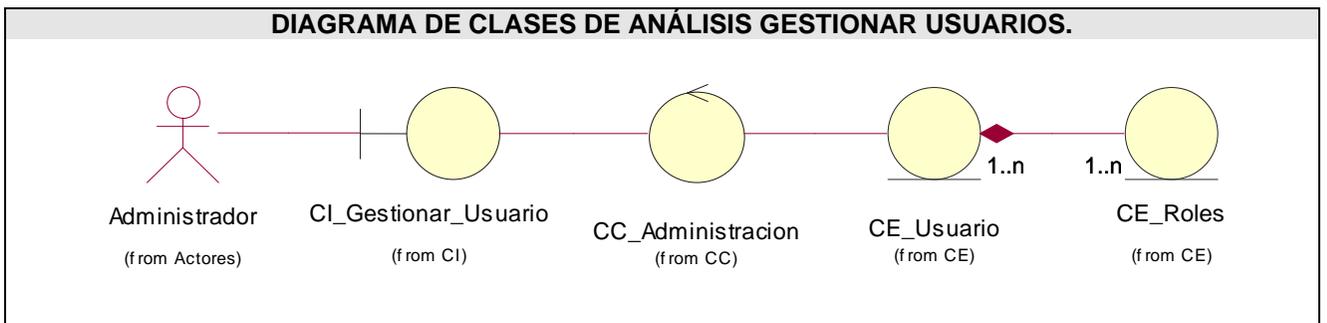


Fig. 2 Diagrama de clases de análisis CU Gestionar Usuarios.

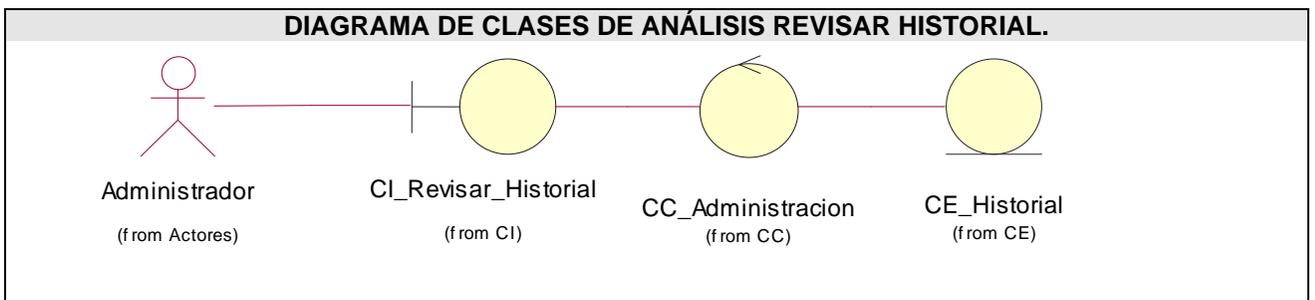


Fig. 3 Diagrama de clases de análisis CU Análisis Revisar Historial .

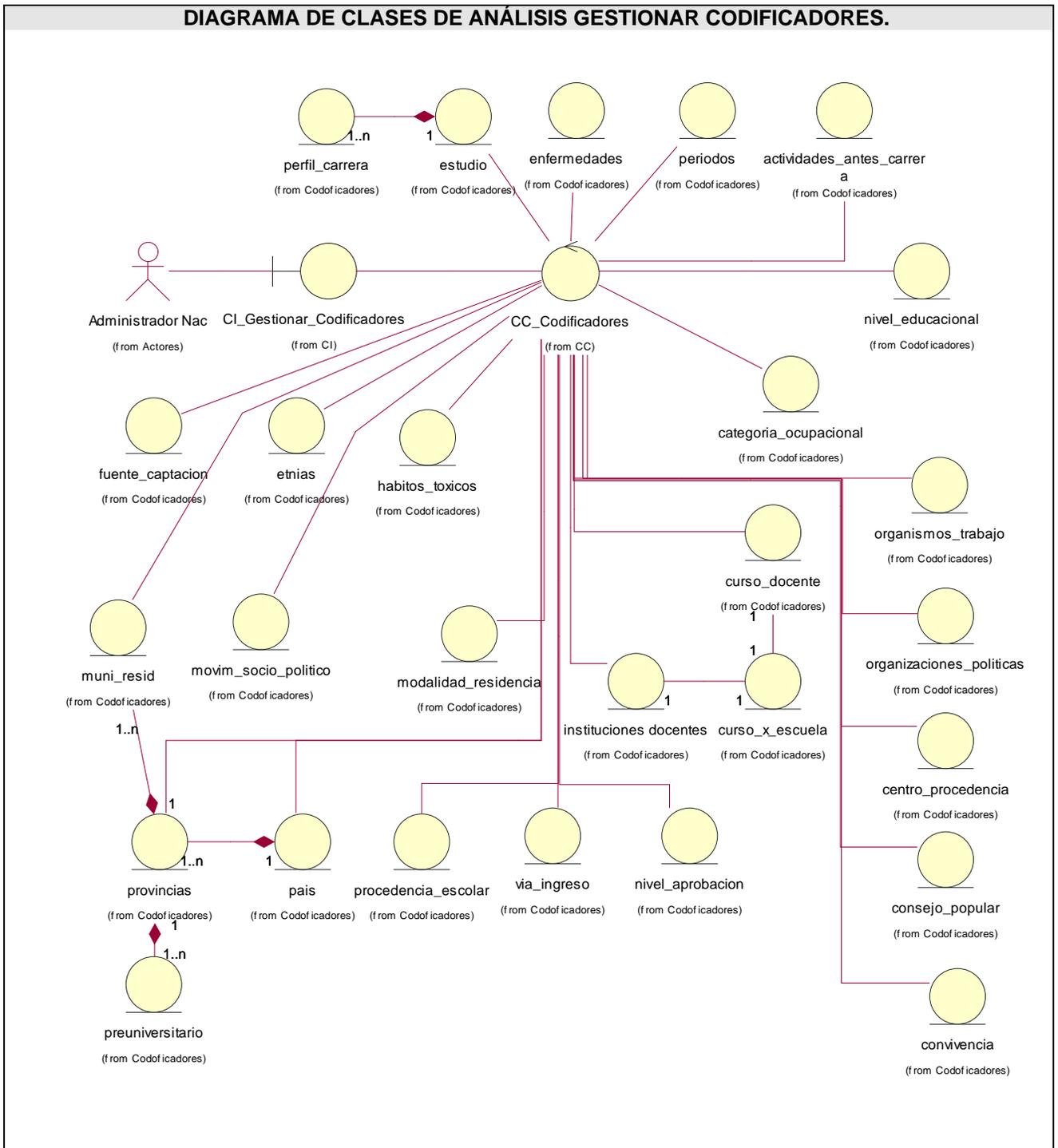


Fig. 4 Diagrama de clases de análisis CU Gestionar Codificadores .

3.1.2 Paquete Matrícula.

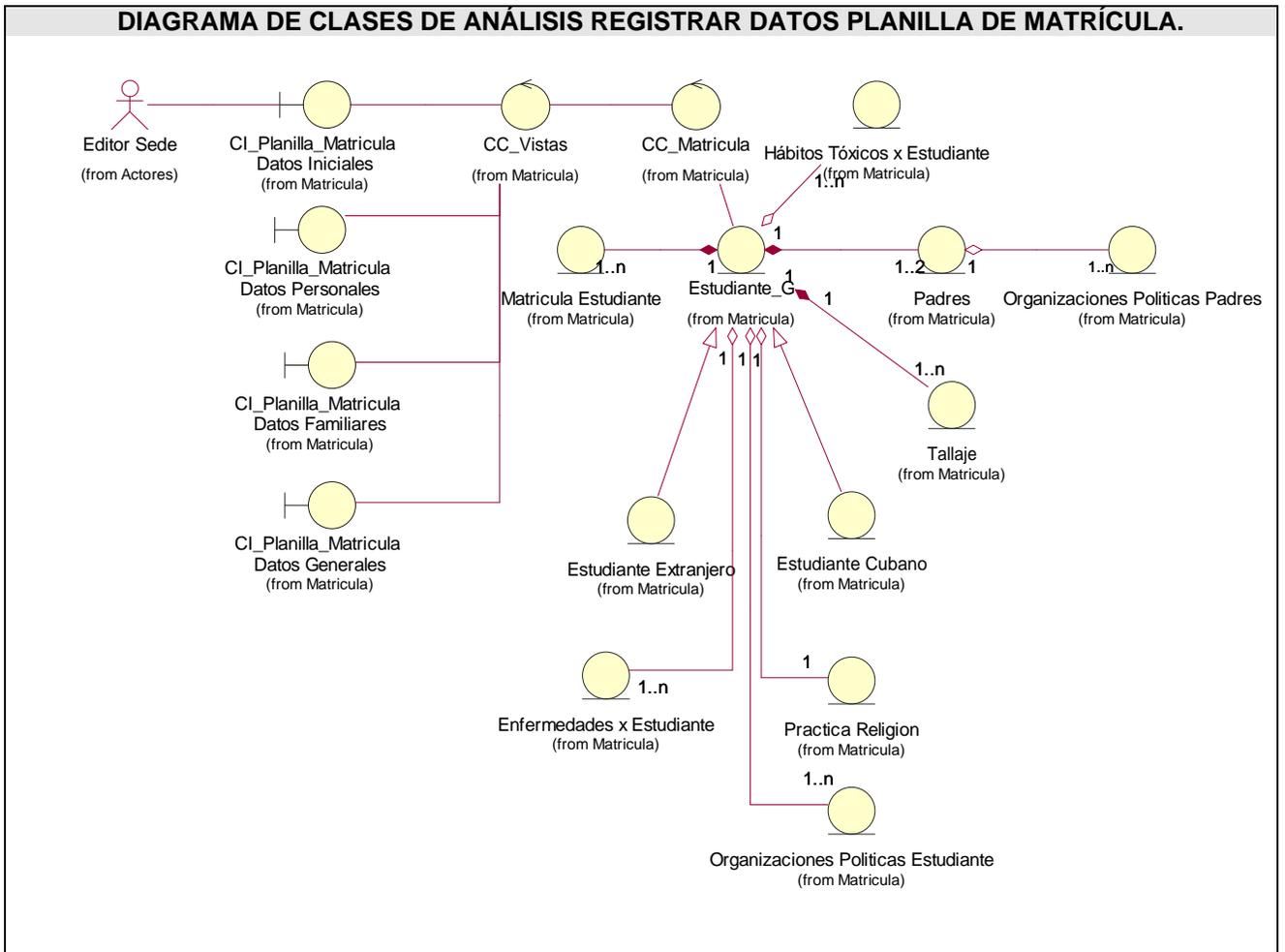


Fig. 5 Diagrama de clases de análisis CU Registrar Datos Planilla de Matrícula .

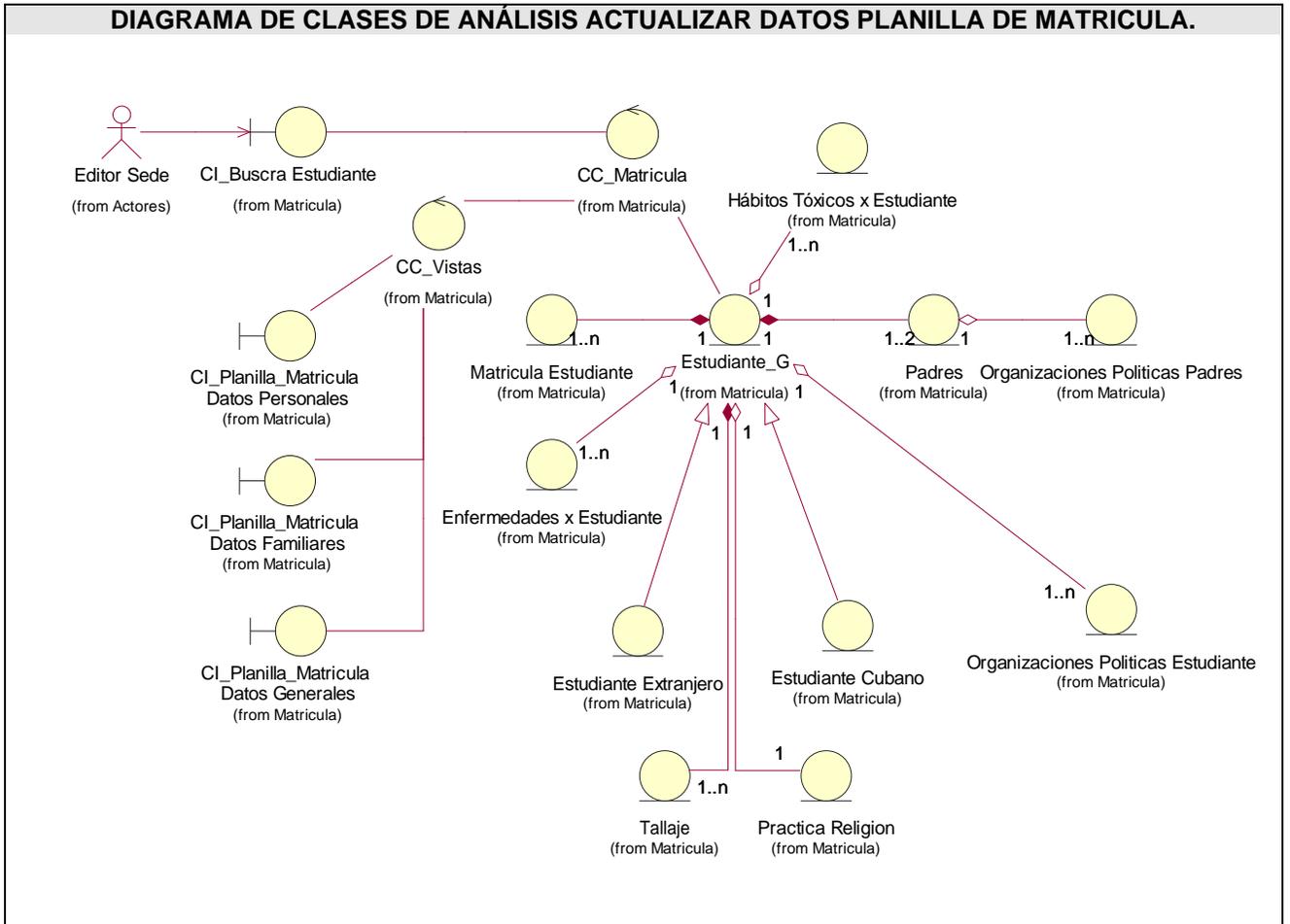


Fig. 6 Diagrama de clases de análisis CU Actualizar Datos Planilla de Matrícula.

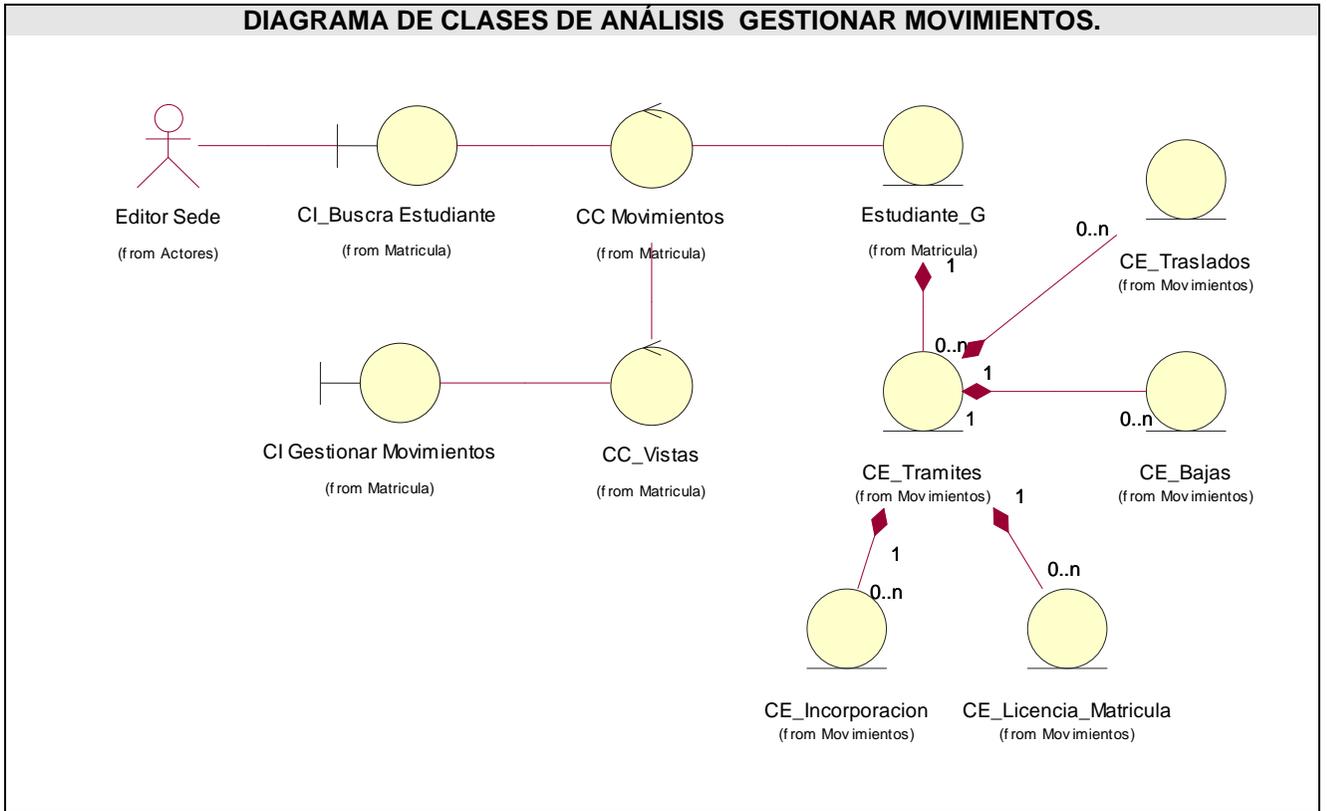


Fig. 7 Diagrama de clases de análisis CU Gestionar Movimientos.

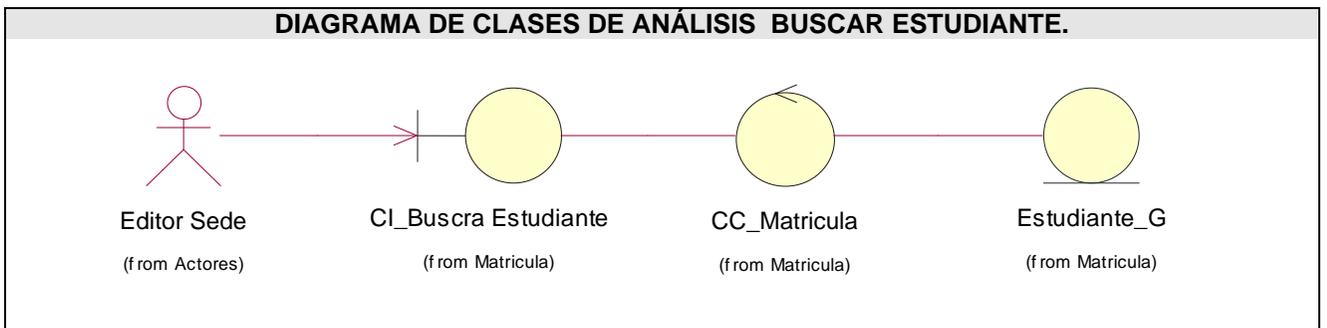


Fig. 8 Diagrama de clases de análisis CU Buscar Estudiante.

3.2 Diseño.

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

3.2.1. Diagramas de clases de diseño.

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

Paquete de Administración.

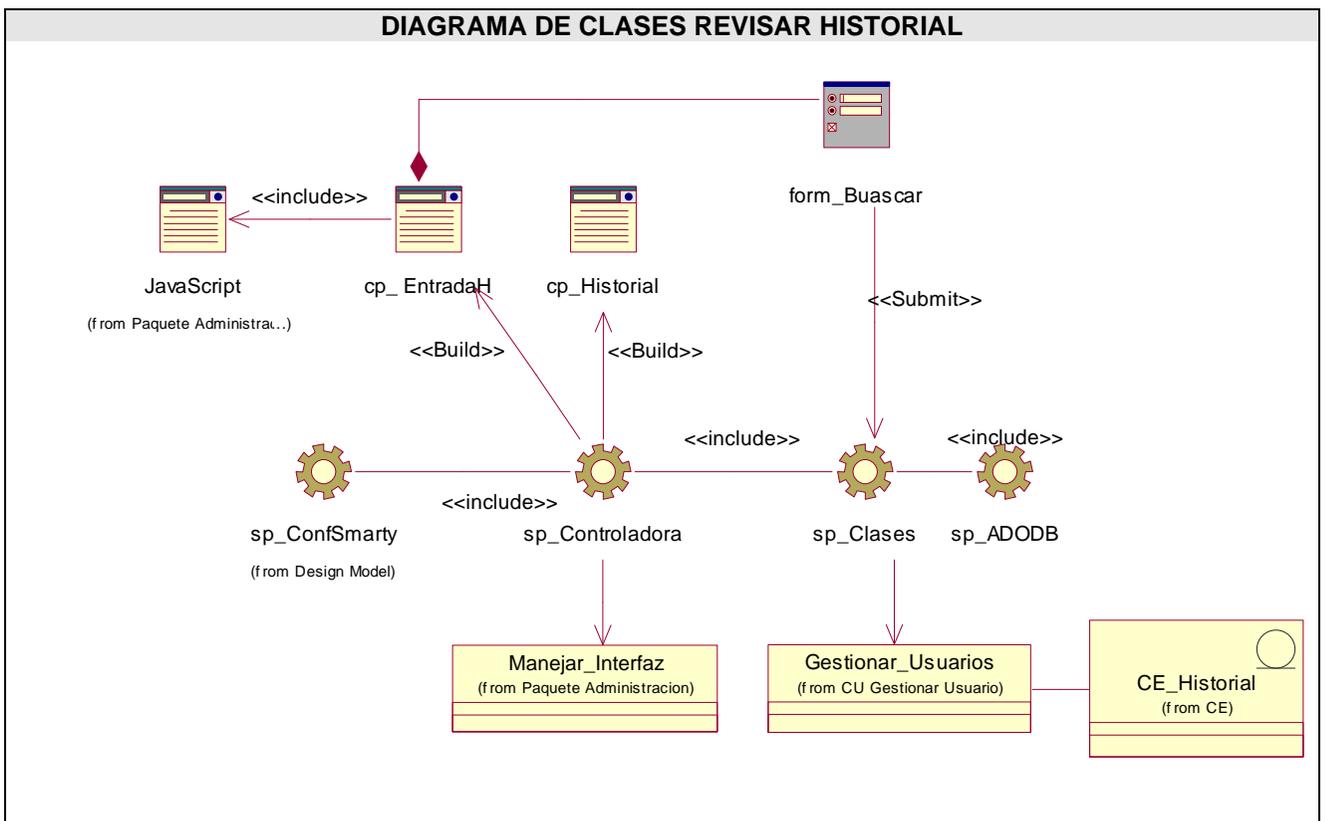


Fig. 9 Diagrama de clases de diseño (CU – A2)Revisar Historial.

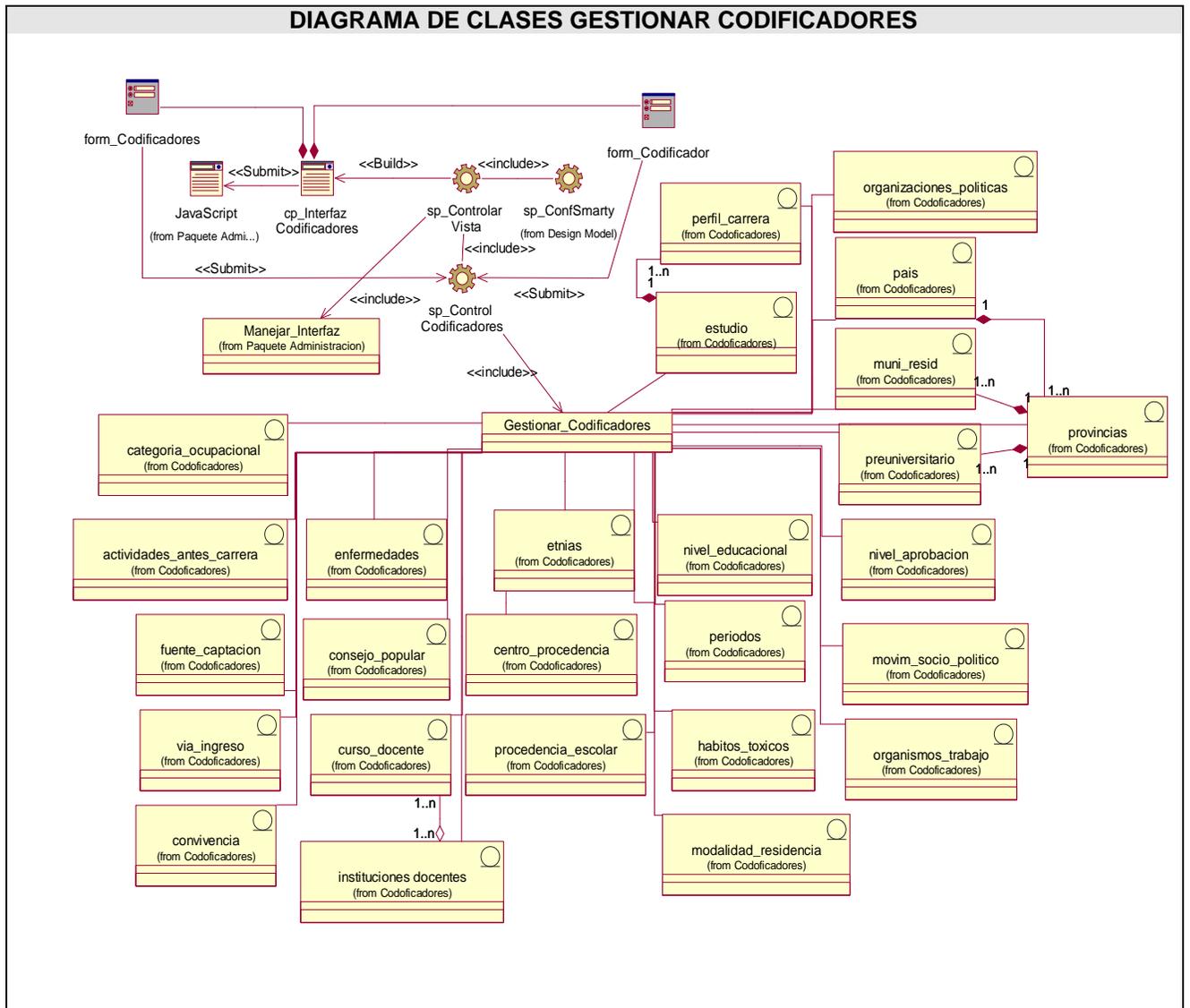


Fig. 10 Diagrama de clases de diseño (CU – A3)Gestionar codificadores.

Paquete de Matrícula.

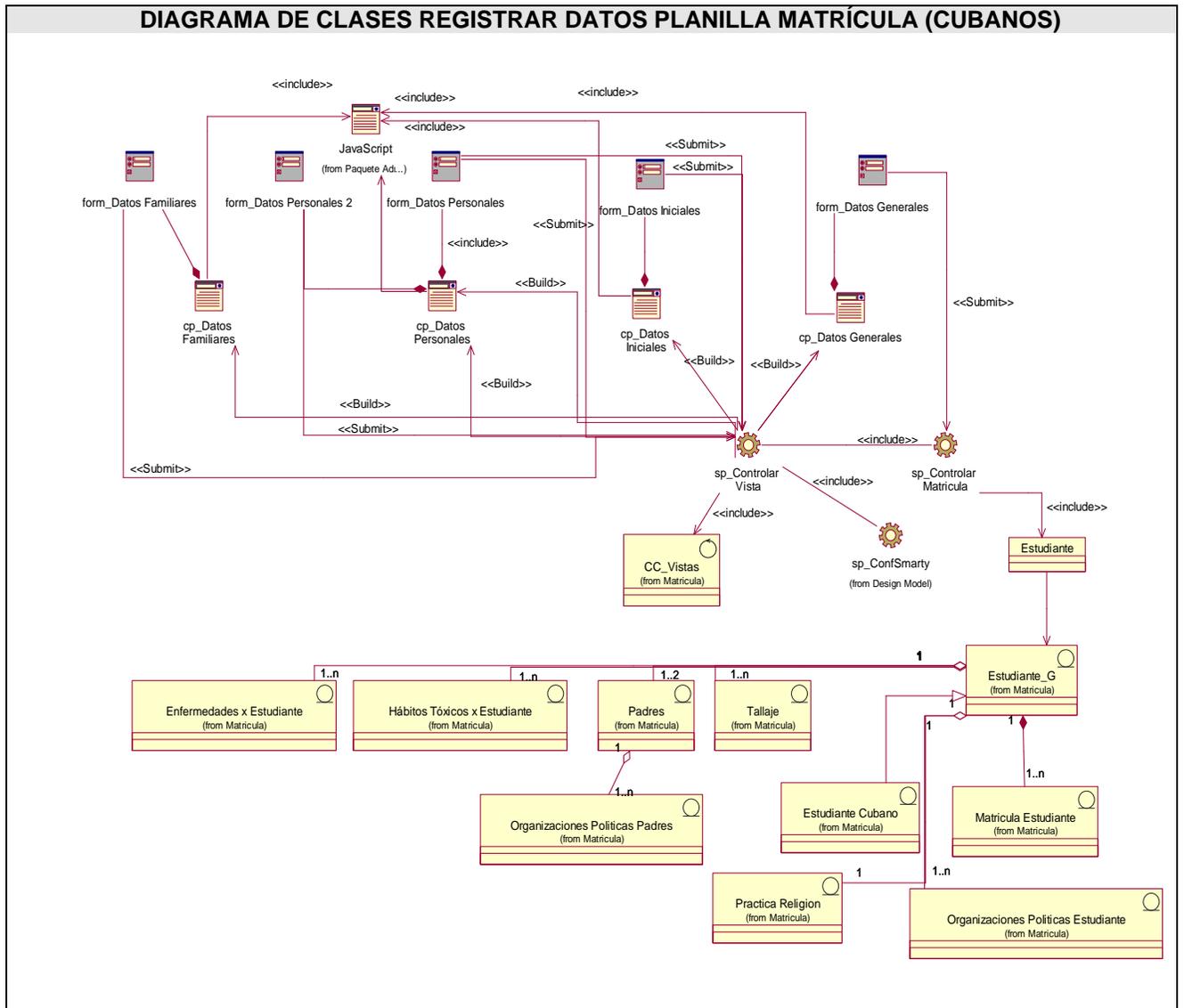


Fig. 11 Diagrama de clases de diseño (CU – M1) Matricular estudiante cubano.

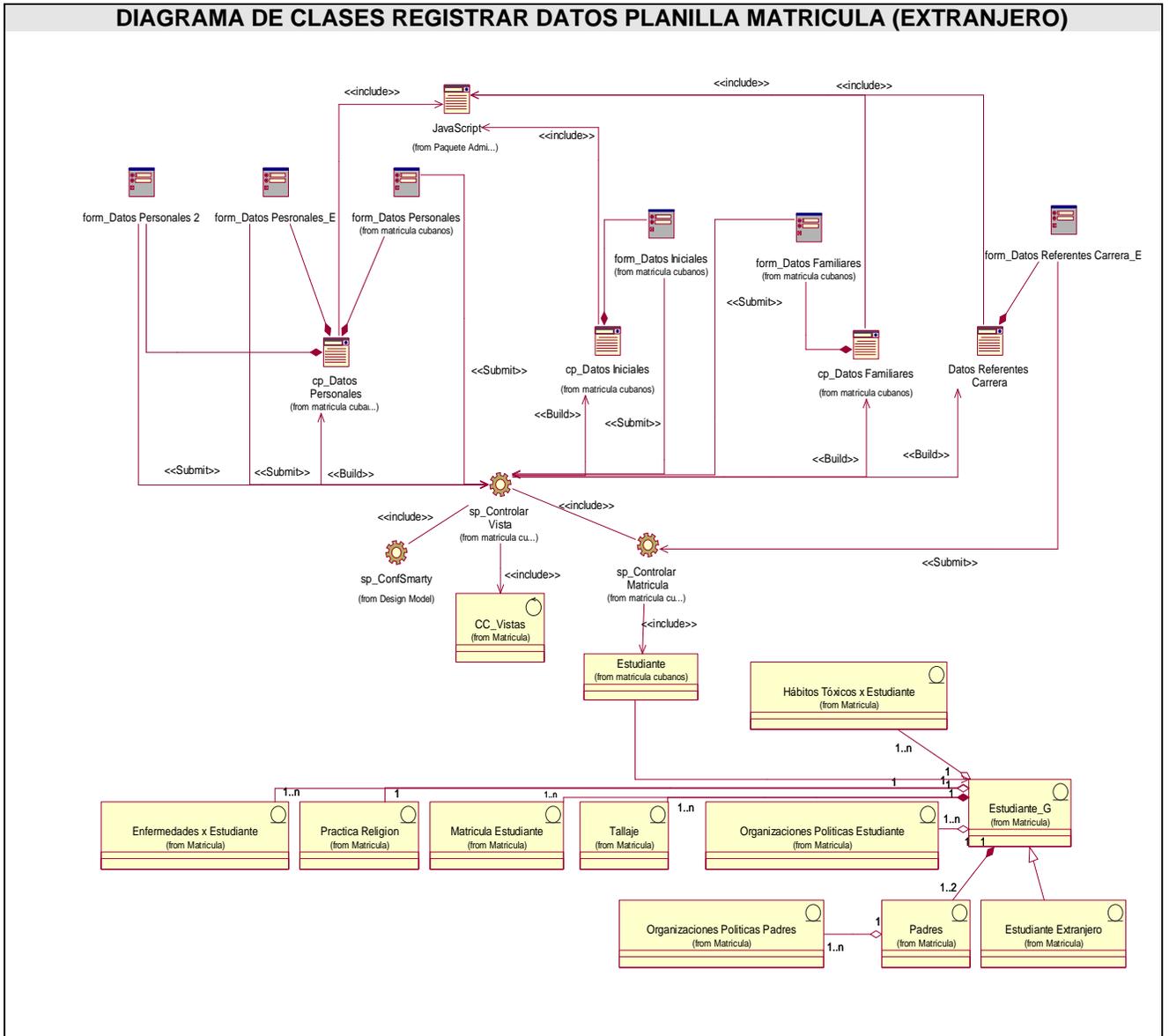


Fig. 12 Diagrama de clases de diseño (CU – M1) Matricular estudiante extranjero.

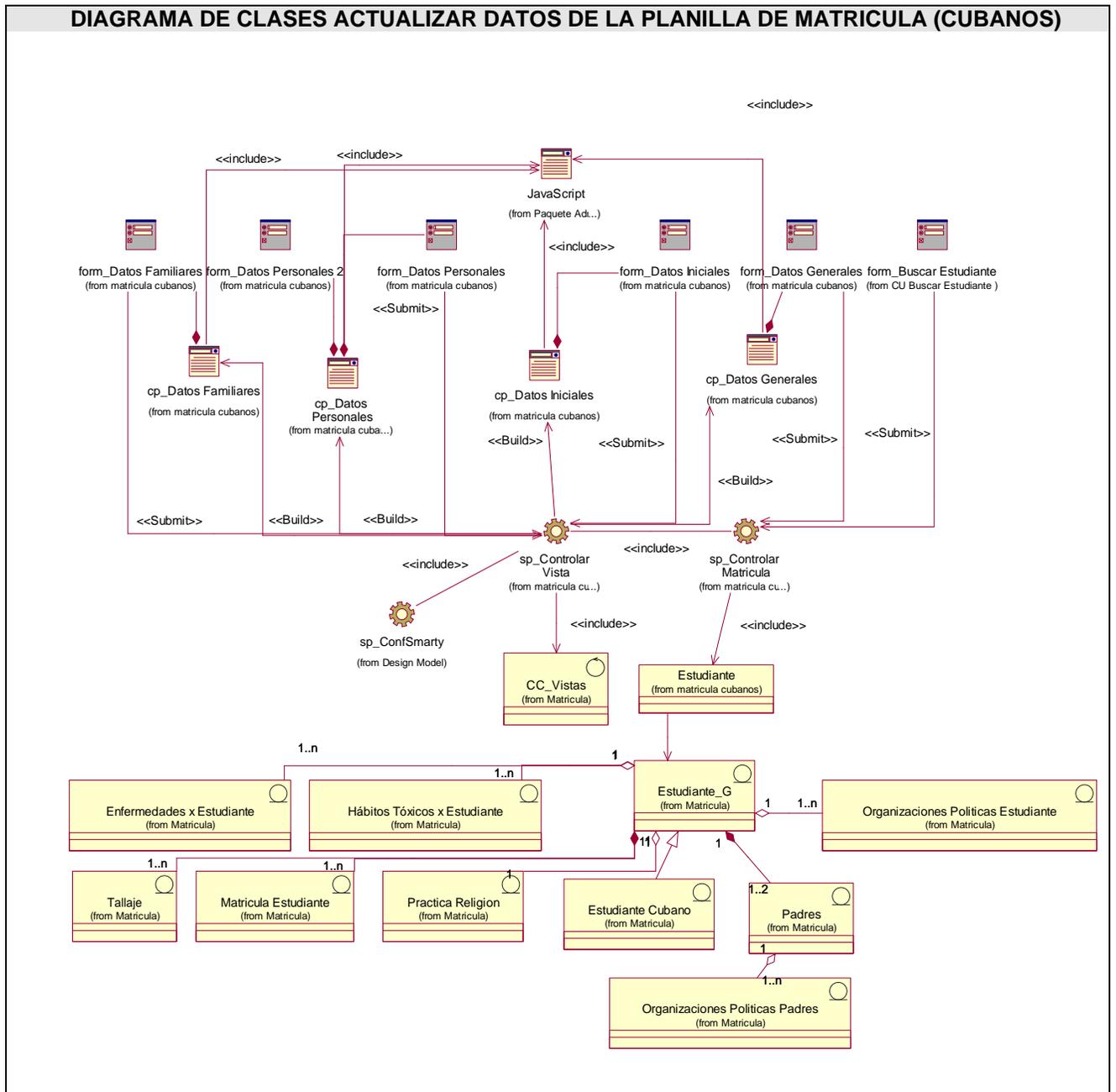


Fig. 13 Diagrama de clases de diseño (CU – M2) Actualizar datos matricula estudiante cubano.

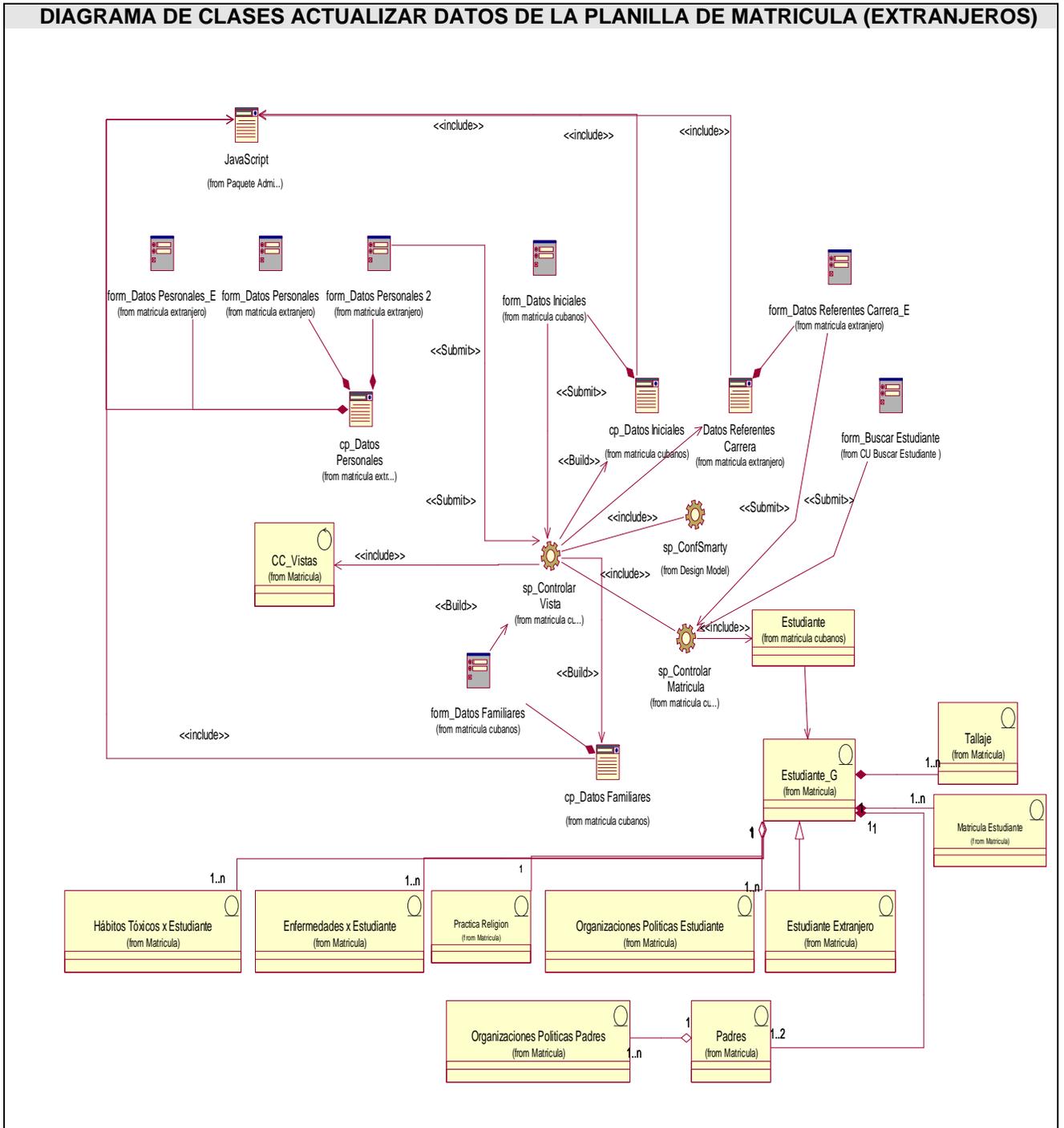


Fig. 14 Diagrama de clases de diseño (CU – M2) Actualizar datos matricula estudiante extranjero.

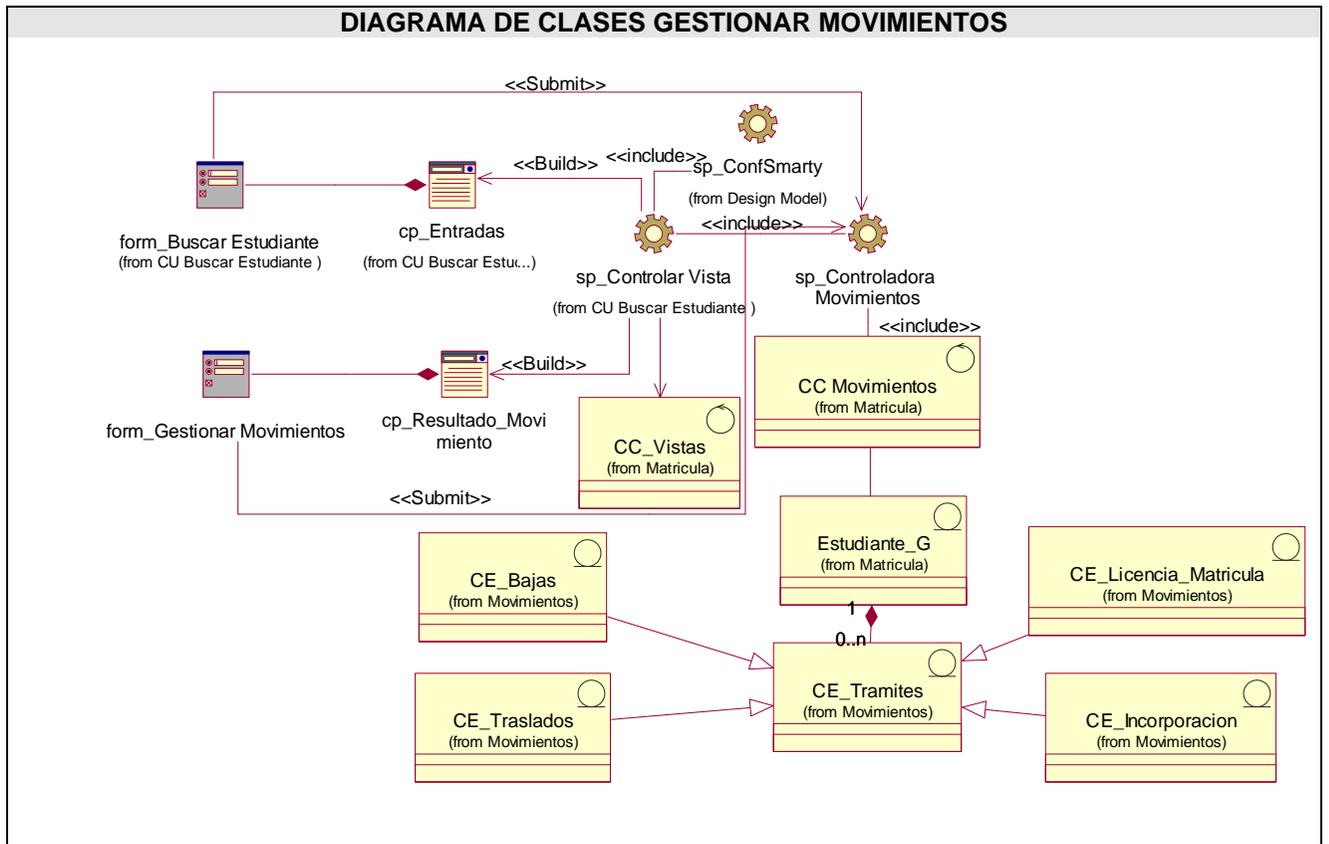


Fig. 15 Diagrama de clases de diseño (CU – M3) Gestionar movimientos.

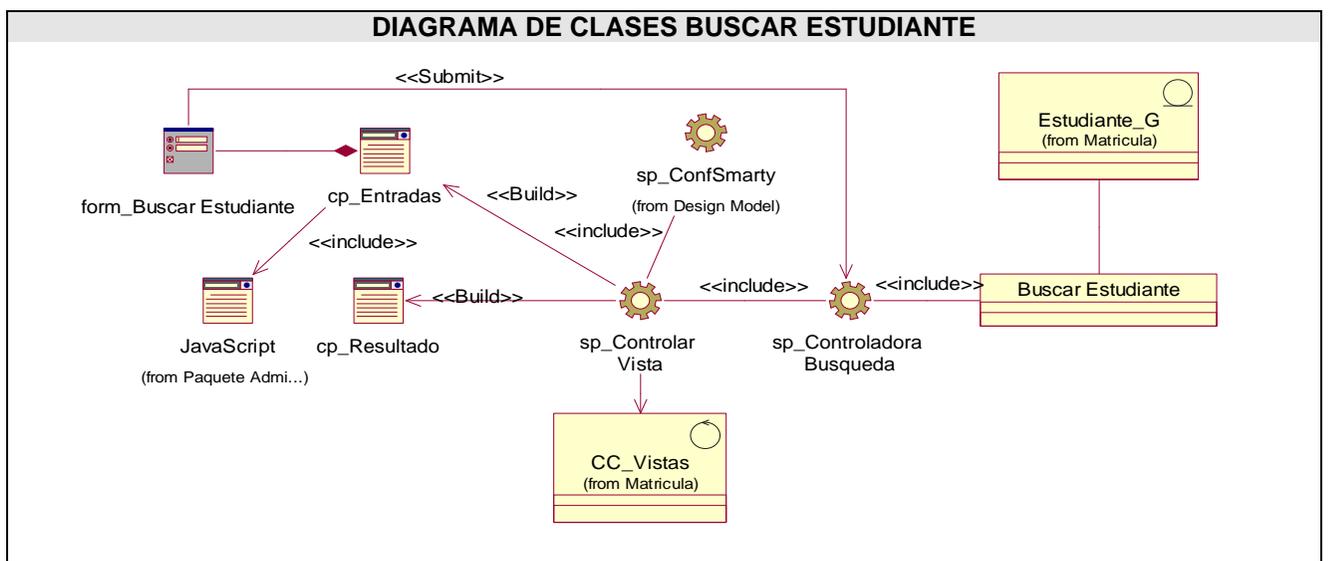


Fig. 16 Diagrama de clases de diseño (CU – M4) Buscar estudiante.

3.2.2 Descripción de las clases.

Nombre: CE_Historial (CU – A2)	
Tipo de clase: Entidad.	
Atributo	Tipo
Fecha	Date
Nombre	String
IP	String
Hora	Date
Login	Strip
Operación	String
Tabla	String
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Perfil_carrera. (CU – A3)	
Tipo de clase: Entidad.	
Atributo	Tipo
Nombre	String
ID_cod_estudios	Integer
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Estudio. (CU – A3)	
Tipo de clase: Entidad.	
Atributo	Tipo
Nombre	String
Descripción	String
Duración	String
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Actividad entes de la carrera. (CU – A3)	
Tipo de clase: Entidad.	
Atributo	Tipo
Nombre	String
ID	Integer
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Categoría ocupacional. (CU – A3)	
Tipo de clase: Entidad.	
Atributo	Tipo
Nombre	String
ID	Integer
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Centro procedencia. (CU – A3)	
Tipo de clase: Entidad.	
Atributo	Tipo
Nombre	String
ID	Integer
ID municipio	Integer
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Consejo popular. (CU – A3)	
Tipo de clase: Entidad.	
Atributo	Tipo
Nombre	String
ID	Integer
ID municipio	Integer
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Etnias. (CU – A3)	
Tipo de clase: Entidad.	
Atributo	Tipo
Nombre	String
ID	Integer
Responsabilidad:	
Nombre:	

Descripción:	
--------------	--

Nombre: Convivencia. (CU – A3)	
Tipo de clase: Entidad.	
Atributo	Tipo
Nombre pariente	String
ID	Integer
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Curso docente. (CU – A3)	
Tipo de clase: Entidad.	
Atributo	Tipo
Nombre curso	String
ID	Integer
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Nivel_educacional. (CU – A3)	
Tipo de clase: Entidad.	
Atributo	Tipo
Nombre	String
ID	Integer
Responsabilidad:	

Nombre:	
Descripción:	

Nombre: Fuente_captación. (CU – A3)	
Tipo de clase: Entidad.	
Atributo	Tipo
Nombre	String
ID	Integer
Descripción	String
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Hábitos_tóxicos. (CU – A3)	
Tipo de clase: Entidad.	
Atributo	Tipo
Nombre	String
ID	Integer
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Modalidad de residencia. (CU – A3)	
Tipo de clase: Entidad.	
Atributo	Tipo

Nombre	String
ID	Integer
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Instituciones_docentes. (CU – A3)	
Tipo de clase: Entidad.	
Atributo	Tipo
Nombre	String
ID	Integer
Instituto	Bolean
ID municipio	Integer
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Estudiante_g. (CU – M1)	
Tipo de clase: Entidad.	
Atributo	Tipo
Id	String
Nombre	String
Segundo_nombre	String
Primer_apellido	String
Segundo_apellido	String
Fecha_nacimiento	String
Sexo	String

Color_piel	String
Es_creyente	Bolean
ID_religion	Integer
Estado_civil	String
No_hijos	Integer
Calle_residencia	String
No_residencia	String
Entre_calle	String
Ciudad_o_pueblo	String
Telefono	String
ID_consejo_popular	Integer
Fecha_matricula	String
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Estudiante_Cubano. (CU – M1)	
Tipo de clase: Entidad.	
Atributo	Tipo
Id	String
Condiciones_vivienda	String
Solvencia_economica	String
Índice_academico	String
Valor_escalafón	String
Zona	String
Concepto_ingrso	String
Situacion_defensa	String
Centro_procedencia	String

Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Estudiante_Extranjero. (CU – M1)	
Tipo de clase: Entidad.	
Atributo	Tipo
Id	String
Observaciones	String
Tipo_ingreso	String
ID_ modalidad_residencia	Integer
No_pasaporte	String
Fecha_vencimiento_ pasaporte	Date
Es_indigena	Boolean
Correo_electronico	String
No_cedula	String
ID_pais	Integer
ID_nacionalidad	Integer
ID_etnia	Integer
ID_pais	Integer
ID_nacionalidad	Integer
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Enfermedades_estudiante. (CU – M1)	
Tipo de clase: Entidad.	
Atributo	Tipo
ID_enfermedad	Integer
ID_estudiante	Integer
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Hábitos_toxicos_estudiante. (CU – M1)	
Tipo de clase: Entidad.	
Atributo	Tipo
ID_habito_toxico	Integer
ID_estudiante	Integer
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Padres. (CU – M1)	
Tipo de clase: Entidad.	
Atributo	Tipo
ID_estudiante	Integer
ID	Integer
ID_ocupacion	Integer
ID_nivel_educacion	Integer
Nombre	String
Segundo_nombre	String

Primer_apellido	String
Segundo_apellido	String
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Organizaciones Políticas Padres. (CU – M1)	
Tipo de clase: Entidad.	
Atributo	Tipo
ID_padre	Integer
ID_organizaciones_politicas	Integer
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Talla. (CU – M1)	
Tipo de clase: Entidad.	
Atributo	Tipo
Blusa	String
Falda	String
Pantalón	Integer
Zapato	String
Camisa	String
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Organizaciones Políticas Estudiante. (CU – M1)	
Tipo de clase: Entidad.	
Atributo	Tipo
ID_organizaciones_politicas	Integer
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Matricula. (CU – M1)	
Tipo de clase: Entidad.	
Atributo	Tipo
ID	Integer
ID_Estudio	Integer
Fecha_matricula	Date
Año_academico	Date
ID_Brigada	String
Tipo_curso	Date
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Practica religión. (CU – M1)	
Tipo de clase: Entidad.	
Atributo	Tipo
ID	Integer

Nombre institución	String
Nombre ministerio	String
Nombre corriente religiosa	String
Nombre movimiento	String
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Tramites. (CU – M3)	
Tipo de clase: Entidad.	
Atributo	Tipo
ID	String
Tramite completado	Boolean
Tipo tramite	String
Año carrera	Integer
Responsabilidad:	
Nombre:	Realizar_Tramite(ID_estudiante, ID_matricula, ID_tramite)
Descripción:	Se encarga de realizar los tramites de movimientos de un estudiante.

Nombre: Bajas. (CU – M3)	
Tipo de clase: Entidad.	
Atributo	Tipo
Tipo baja	String
Causa	String
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Traslados. (CU – M3)	
Tipo de clase: Entidad.	
Atributo	Tipo
Tipo traslado	String
Causa	String
Centro docente origen	String
Centro docente destino	String
Provincia origen	String
Provincia destino	String
Municipio origen	String
Municipio destino	String
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Incorporación. (CU – M3)	
Tipo de clase: Entidad.	
Atributo	Tipo
Causa	String
Fecha solicitud	Date
Fecha aprobación	Date
ID provincia	Integer
ID municipio	Integer
ID nivel de aprobación	Integer
Centro Docente	
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Licencia matricula. (CU – M3)	
Tipo de clase: Entidad.	
Atributo	Tipo
Causa	String
Fecha solicitud	Date
Fecha aprobación	Date
Tipo Licencia	String
Responsabilidad:	
Nombre:	
Descripción:	

Nombre: Manejar_Interfaz	
Tipo de clase: Controladora.	
Atributo	Tipo
Responsabilidad:	
Nombre:	
Descripción:	Se encarga crear y manejar las interfaces del sistema.

Nombre: Gestionar_Codificadores (CU – M3)	
Tipo de clase: Controladora.	
Atributo	Tipo
Nombre	String
ID	Integer
Descripción	String

ID_Codificador	Integer
Responsabilidad:	
Nombre:	Insertar(nombre, descripcion, id_codificador)
Descripción:	Se encarga de adicionar elementos a los codificadores existentes en la base de datos del sistema, pasándole como parámetro el nombre del valor, la descripción de lo que significa este en el codificador y por último el identificador del codificador sobre el que se realizará la operación.
Nombre:	Modificar(id, nombre, descripcion, id_codificador)
Descripción:	Se encarga de actualizar los valores existentes en los codificadores de la base de datos del sistema, para lo cual se le pasan como parámetros el identificador del elemento a modificar o actualizar, los nuevos valores actualizados ya sea el nombre o la descripción y por último el identificador del codificador sobre el cual se realizara la operación.
Nombre:	Eliminar(id, id_codificador)
Descripción:	Se encarga de eliminar los valores de los codificadores, para lo cual se le pasa como parámetro el identificador del valor que se desea eliminar y el identificador del codificador sobre el cual se desea realizar la operación.

Nombre: CC_Vista	
Tipo de clase: Controladora.	
Atributo	Tipo
Responsabilidad:	
Nombre:	
Descripción:	Se encarga crear y manejar las interfaces del sistema relacionadas con el paquete de matrícula.

Nombre: Estudiante. (CU – M1) (CU – M2)	
Tipo de clase: Controladora.	
Atributo	Tipo
Id	String
Nombre	String
Segundo_nombre	String
Primer_apellido	String
Segundo_apellido	String
Fecha_nacimiento	String
Sexo	String
Color_piel	String
Es_creyente	Boolean
ID_religion	Integer
Estado_civil	String
No_hijos	Integer
Calle_residencia	String
No_residencia	String
Entre_calle	String
Ciudad_o_pueblo	String
Telefono	String
ID_consejo_popular	Integer
ID_procedencia_escolar	Integer
Solvencia_economica	String
Indice_academico	Integer
Valor_escalafon	Integer
ID_via_ingreso	Integer
ID_ocupacion	Integer
ID_situacion_escolar	Integer
ID_centro_procedencia	Integer

ID_cod_escuela	Integer
ID_estudios	Integer
Situacion_defensa	String
Fecha_matricula	Date
Observaciones	String
Tipo_ingreso	String
ID_modalidad_residencia	Integer
No_pasaporte	String
Fecha_vencimiento_pasaporte	Date
Es_indigena	Boolean
Correo_electronico	String
No_cedula	String
ID_pais	Integer
ID_nacionalidad	Integer
ID_etnia	Integer
Responsabilidad:	
Nombre:	ADD Estudiante_Cubano(Id, Nombre, Segundo_nombre, Primer_apellido, Segundo_apellido, Fecha_nacimiento, Sexo, Color_piel, Es_creyente, ID_religion, Estado_civil, No_hijos, No_hijos, Calle_residencia, No_residencia, Entre_calle, Ciudad_o_pueblo, Telefono, ID_consejo_popular, ID_procedencia_escolar, Solvencia_economica, Indice_academico, Valor_escalafon, ID_via_ingreso, ID_ocupacion, ID_situacion_escolar, ID_centro_procedencia, ID_cod_escuela, ID_estudios, Situacion_defensa, Fecha_matricula)
Descripción:	Se encarga de insertar un nuevo estudiante cubano en el sistema para lo cual se le pasa como parámetros los datos del estudiante.

Nombre:	ADD Estudiante_Extranjero(Id, Nombre, Segundo_nombre, Primer_apellido, Segundo_apellido, Fecha_nacimiento, Sexo, Color_piel, Es_creyente, ID_religion, Estado_civil, No_hijos, No_hijos, Calle_residencia, No_residencia, Entre_calle, Ciudad_o_pueblo, Telefono, ID_consejo_popular, ID_procedencia_escolar, Solvencia_economica, ID_via_ingreso, ID_ocupacion, ID_situacion_escolar, ID_estudios, Fecha_matricula, Tipo_ingreso, Observaciones, ID_modalidad_residencia, No_pasaporte, Fecha_vencimiento_pasaporte, Es_indigena, Correo_electronico, No_cedula, ID_pais, ID_nacionalidad, ID_etnia)
Descripción:	Se encarga de insertar un nuevo estudiante extranjero en el sistema para lo cual se le pasa como parámetros los datos del estudiante.
Nombre:	Actualizar_Estudiante_Cubano(Id, Nombre, Segundo_nombre, Primer_apellido, Segundo_apellido, Fecha_nacimiento, Sexo, Color_piel, Es_creyente, ID_religion, Estado_civil, No_hijos, No_hijos, Calle_residencia, No_residencia, Entre_calle, Ciudad_o_pueblo, Telefono, ID_consejo_popular, ID_procedencia_escolar, Solvencia_economica, Indice_academico, Valor_escalafon, ID_via_ingreso, ID_ocupacion, ID_situacion_escolar, ID_centro_procedencia, ID_cod_escuela, ID_estudios, Situacion_defensa, Fecha_matricula)
Descripción:	Se encarga de actualizar los datos de un estudiante cubano en el sistema para lo cual se le pasa como parámetros los datos que se le decena actualizar al estudiante, en el caso que no se le actualicen ningún dato persistirán los anteriores.
Nombre:	Actualizar_Estudiante_Extranjero(Id, Nombre, Segundo_nombre, Primer_apellido, Segundo_apellido, Fecha_nacimiento, Sexo, Color_piel, Es_creyente, ID_religion, Estado_civil, No_hijos, No_hijos, Calle_residencia, No_residencia, Entre_calle, Ciudad_o_pueblo, Telefono, ID_consejo_popular, ID_procedencia_escolar, Solvencia_economica, ID_via_ingreso,

	ID_ocupacion, ID_situacion_escolar, ID_estudios, Fecha_matricula, Tipo_ingreso, Observaciones, ID_modalidad_residencia, No_pasaporte, Fecha_vencimiento_pasaporte, Es_indigena, Correo_electronico, No_cedula, ID_pais, ID_nacionalidad, ID_etnia)
Descripción:	Se encarga de actualizar los datos de un estudiante extranjero en el sistema para lo cual se le pasa como parámetros los datos que se le decena actualizar al estudiante, en el caso que no se le actualicen ningún dato persistirán los anteriores.

Nombre: Movimientos. (CU – M3)	
Tipo de clase: Controladora.	
Atributo	Tipo
ID_estudiante	String
ID_matricula	Integer
ID_tramite	Integer
Responsabilidad:	
Nombre:	Realizar_Tramite(ID_estudiante, ID_matricula, ID_tramite)
Descripción:	Se encarga de realizar los tramites de movimientos de un estudiante.

Nombre: Buscar Estudiante. (CU – M2) (CU – M3)	
Tipo de clase: Controladora.	
Atributo	Tipo
Id estudiante	String
Nombre	String
Brigada	String
Sexo	String
Responsabilidad:	

Nombre:	Buscar_Estudiante(Id_estudiante, Nombre, Brigada, Sexo)
Descripción:	Se encarga de la búsqueda de uno o varios estudiante especificado por cualquiera de los datos anteriores.

Nombre: EntradaH. (CU – A2)	
Tipo de clase: Interfaz.	
Atributo	Tipo
Nombre usuario	Satring
Fecha	String
Responsabilidad:	
Nombre:	Validar_Entradas(Nombre usuario, Fecha)
Descripción:	Indica el proceso de revisar historial verificando que la entrada de datos sea correcta.
Nombre: Historial. (CU – A2)	
Tipo de clase: Interfaz.	
Atributo	Tipo
Responsabilidad:	
Nombre:	Muestra_Informacion()
Descripción:	Muestra el resultado de de la consulta sobre la revisión de uno o varios usuarios.

Nombre: Interfaz_Codificadores (CU – A3)	
Tipo de clase: Interfaz.	
Atributo	Tipo
actividades_antes_carrera	Satring
carreras	Satring
categoria_ocupacional	Satring

centro_procedencia	Satring
consejo_popular	Satring
convivencia	Satring
curso_docente	Satring
curso_x_institucion docente	Satring
enfermedades	Satring
estudio	Satring
etnias	Satring
fuelle captacion	Satring
mivimiento socio politicos	Satring
habitos toxicos	Satring
instituciones docentes	Satring
modalidad residencia	Satring
munisipio_residencia	Satring
provincia residencia	Satring
nivel aprobacion	Satring
nivel educacional	Satring
organismos trabajo	Satring
organizaciones politicas	Satring
pais	Satring
perfil carrera	Satring
periodos	Satring
preuniversitarios	Satring
procedencia escolar	Satring
provincias	Satring
via_ingreso	Satring
Responsabilidad:	
Nombre:	Muestra_Codificadores()

Descripción:	Muestra los codificadores del sistema.
Nombre:	Entrada_Valores()
Descripción:	Indica el proceso adicionar un valor a un codificador seleccionado validando que la entrada de los datos sea correcta.

Nombre: Datos_Iniciales (CU – M1)	
Tipo de clase: Interfaz.	
Atributo	Tipo
Carrera	String
Perfil	String
Tipo_curso	String
Tipo_estudiante	String
Responsabilidad:	
Nombre:	Entrada_Valores()
Descripción:	Indica el proceso de adicionar un estudiante al sistema verificando que la entrada de los datos sea correcta.

Nombre: Datos Personales (CU – M1)	
Tipo de clase: Interfaz.	
Atributo	Tipo
Nombre	String
2do_Nombre	String
1er_Apellido	String
2do_Apellido	String
Fecha nacimiento	Date
Sexo	String

Color de la piel	String
No CI	String
Creencias Religiosas	String
talla pantalon	String
talla falda	String
talla calzado	String
talla camisa	String
talla blusa	String
Organizacion politica	String
Convivencia	String
Procedencia social	String
Actividad que realizaba	String
Direccion particular	String
No de Hijos	Integer
Estado civil	String
Rango edades hijos	Integer
Convivencia	String
Actividad_ realizaba	String
No cedula	String
No pasaporte	String
Fecha vencimiento	Date
Pais procedencia	String
Provincia procedencia	String
municipio procedencia	String
Dirección	String
Email	String
Responsabilidad:	
Nombre:	Entrada_Valores()
Descripción:	Indica la acción de adicionar los datos personales de un estudiante verificando la

	entrada de los mismos y mostrando el formulario en dependencia del tipo de estudiante que se esta adicionando.
--	--

Nombre: Datos_Familiares (CU – M1)	
Tipo de clase: Interfaz.	
Atributo	Tipo
Nombre_padre	String
2do_Nombre_padre	String
1er_Apellido_padre	String
2do_Apellido_padre	String
Organizacion politica_padre	String
Nivel_educacional_padre	String
Vive_padre	String
Categoria ocupacional_padre	String
Nombre_madre	String
2do_Nombre_madre	String
1er_Apellido_madre	String
2do_Apellido_madre	String
Organizacion politica_madre	String
Nivel_educacional_madre	String
Vive_madre	String
Categoria ocupacional_madre	String
Responsabilidad:	
Nombre:	Entrada_ValoresDF()

Descripción:	Indica la acción de adicionar los datos familiares del estudiante al sistema verificando la entrada de los mismos.
--------------	--

Nombre: Datos Referentes Carrera_E(CU – M1)	
Tipo de clase: Interfaz.	
Atributo	Tipo
Matriculadoanteriormente	Boolean
Fecha	Date
Procedencia escolar	String
Tipo procedencia	String
Captacion carrera	String
Observaciones	String
Responsabilidad:	
Nombre:	Entrada_ValoresRC()
Descripción:	Indica la acción de adicionar los datos referentes a la carrera para un estudiante extranjero verificando la entrada de los mismos.

Nombre: Datos Generales_C (CU – M1)	
Tipo de clase: Interfaz.	
Atributo	Tipo
Matriculadoanteriormente	Boolean
Fecha_ingrso	Date
Procedencia escolar	String
Tipo procedencia	String
Observaciones	String
Indice académico	Float
Valor escalafón	Float

Responsabilidad:	
Nombre:	Entrada_ValoresDG()
Descripción:	Indica la acción de adicionar los datos generales del estudiante extranjero al sistema verificando la entrada de los mismos.

Nombre: Buscar_Estudiante(CU – M2)	
Tipo de clase: Interfaz.	
Atributo	Tipo
Nombre	String
Sexo	String
Brigada	String
Responsabilidad:	
Nombre:	Entrada_Valores()
Descripción:	Indica la acción de buscar un estudiante en específico o una brigada docente completa, verificando la entrada de los datos al formulario.

Nombre: Resultado_Movimiento. (CU – M3)	
Tipo de clase: Interfaz.	
Atributo	Tipo
Tipo _Trámite	String
Tipo _Traslado	String
Tipo _Baja	String
Causas	String
Tiempo_solicitado	String
Tiempo_aprobado	String
Responsabilidad:	
Nombre:	Entrada_Valores()

Descripción:	Indica la acción de solicitud de movimientos ya sea cualquiera de los tipos existentes, y verifica la corrección de los datos entrados.
--------------	---

3.2.3 Diagramas de Interacción.

Los diagramas de interacción son diagramas dinámicos que muestran como colaboran los objetos que intervienen en la realización de los casos de uso. Existen dos tipos de estos diagramas que son diagramas de secuencias y diagramas de colaboración. En el trabajo se desarrollaron los diagramas de secuencia que muestran la secuencia ordenada de las operaciones entre objetos, ver [Anexo 3](#).

3.3 Diseño de la BD.

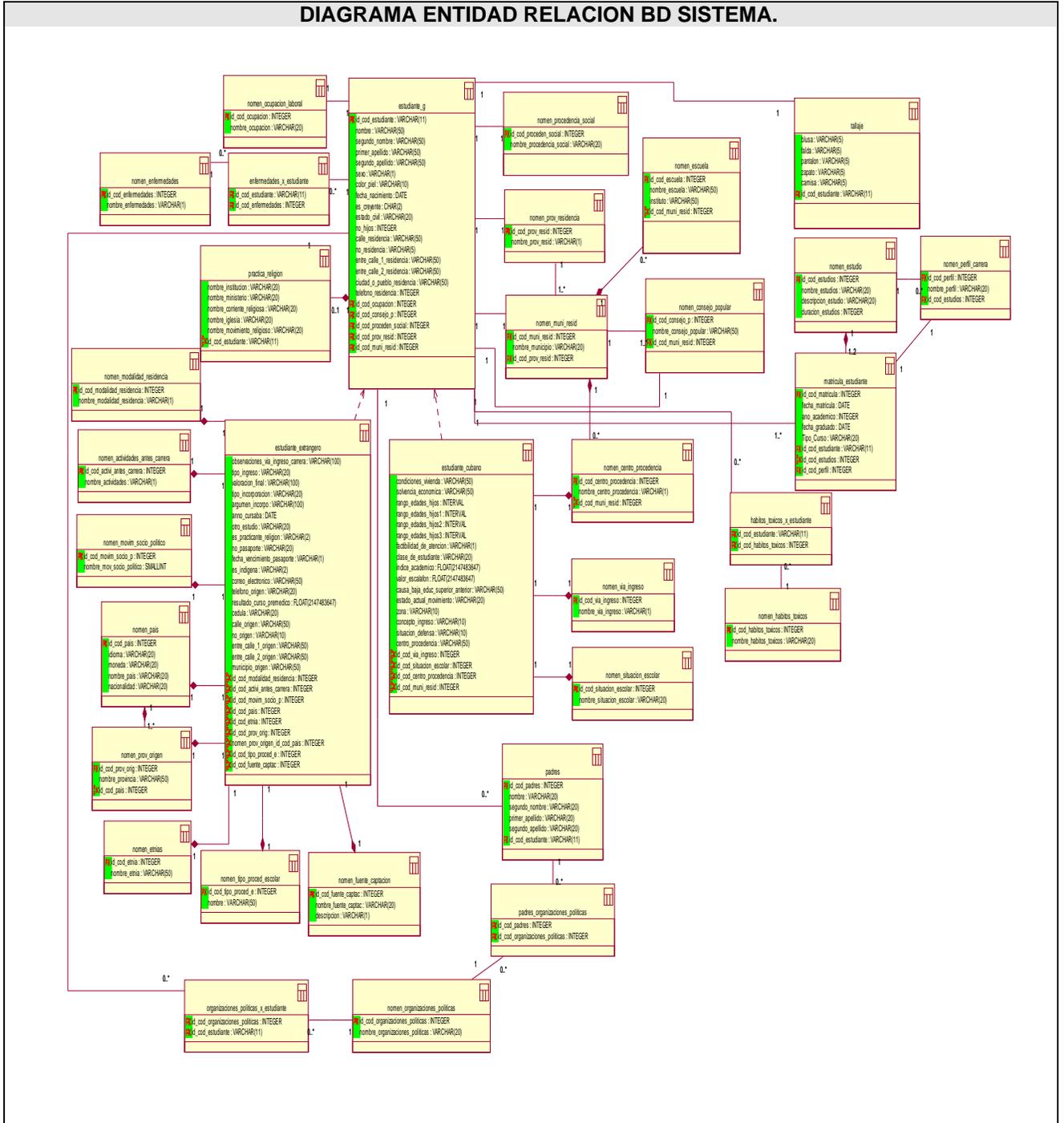


Fig. 16 Diagrama de entidad relación de la BD

3.3.1 Descripción de las tablas.

Nombre: estudiante_g		
Descripción: Esta tabla contiene la generalización de los datos de los estudiantes extranjeros y cubanos.		
Atributo	Tipo	Descripción
id_cod_estudiante	Varchar	Representa el identificador o No.CI del estudiante.
Nombre	Varchar	Representa el nombre del estudiante.
segundo_nombre	Varchar	Representa el segundo nombre del estudiante en caso de que tenga nombre compuesto.
primer_apellido	Varchar	Representa el 1er apellido del estudiante.
segundo_apellido	Varchar	Representa el 2do apellido del estudiante.
Sexo	Char	Representa el sexo del estudiante.
color_piel	Varchar	Representa el color de la piel del estudiante.
fecha_nacimiento	Date	Representa la fecha de nacimiento del estudiante.
es_creyente	Char	Contiene si el estudiante es creyente de alguna religión, se recoge Si o No.
estado_civil	Varchar	Representa el estado civil del estudiante.
no_hijos	Integer	Representa el número de hijos del estudiante.
calle_residencia	Varchar	Representa el nombre de la calle donde reside el estudiante.
no_residencia	Varchar	Representa el número de la residencia del estudiante.
entre_calle_1_residencia	Varchar	Representa el nombre de las entrecalles de la dirección del estudiante.
entre_calle_2_residencia	Varchar	Representa el nombre de las entrecalles de

		la dirección del estudiante.
ciudad_o_pueblo_residencia	Varchar	Representa el nombre de la ciudad o pueblo donde vive el estudiante.
telefono_residencia	Integer	Representa el número telefónico de la residencia del estudiante.

Nombre: estudiante_extranjero		
Descripción: Esta tabla contiene la especialización de los datos de los estudiantes extranjeros.		
Atributo	Tipo	Descripción
observaciones_via_ingreso_carrera	Varchar	Representa la información adicional que se tiene del estudiante recogida por el organismo que lo captó para la carrera en su país.
tipo_ingreso	Varchar	Representa el tipo de ingreso del estudiante.
valoracion_final	Varchar	Representa la valoración que se tenga sobre el estudiante.
Otro_estudio	Varchar	Representa si el estudiante ha cursado algún estudio anteriormente en la educación superior.
Anno_cursaba	Date	Representa el año que cursaba en la educación superior en caso de haber estado estudiando otra carrera.
tipo_incorporacion	Varchar	Representa el tipo de incorporación a la carrera que se encuentra estudiando.
argumen_incorpo	Varchar	Representa el argumento de la incorporación del estudiante a la carrera.
es_practicante_religion	Varchar	Representa si el estudiante practica

		religión.
no_pasaporte	Varchar	Representa el número de pasaporte del estudiante.
fecha_vencimiento_pasaporte	Date	Representa la fecha de vencimiento del pasaporte del estudiante.
es_indigena	Varchar	Guarda si el estudiante es indígena o no.
correo_electronico	Varchar	Representa el correo electrónico del estudiante.
telefono_origen	Varchar	Representa el número telefónico del estudiante en su país de procedencia.
resultado_curso_premedico	Integer	Representa la nota del curso premédico del estudiante.
Cedula	Varchar	Representa el No. de cedula o No. CI del estudiante en su país.
calle_origen	Varchar	Representa la dirección del estudiante en su país.
no_origen	Varchar	Representa No. de la casa del estudiante en su país.
entre_calle_1_origen	Varchar	Representa las entrecalles de la dirección del estudiante en su país.
entre_calle_2_origen	Varchar	Representa las entrecalles de la dirección del estudiante en su país.
municipio_origen	Varchar	Representa el nombre del municipio o ciudad del estudiante en su país.
id_cod_pais	Integer	Representa el código del país de procedencia del estudiante. Este atributo es una llave foránea absorbida de la tabla nomen_pais.
id_cod_etnia	Integer	Representa el código de la etnia del

		estudiante en caso que este sea indígena. Este atributo es una llave foránea absorbida de la tabla nomen_etnias.
id_cod_prov_orig	Integer	Representa el código de la provincia de procedencia del estudiante. Este atributo es una llave foránea absorbida de la tabla nomen_provincias.
id_cod_tipo_proced_e	Integer	Representa el código del tipo de procedencia del estudiante. Este atributo es una llave foránea absorbida de la tabla nomen_procedencia_social.
id_cod_fuente_captac	Integer	Representa el código de la fuente de captación del estudiante. Este atributo es una llave foránea absorbida de la tabla nomen_fuente_captacion.

Nombre: estudiante_cubano		
Descripción: Esta tabla contiene la especialización de los datos de los estudiantes cubanos.		
Atributo	Tipo	Descripción
condiciones_vivienda	Varchar	Representa las condiciones de la vivienda del estudiante.
solvencia_economica	Varchar	Representa la solvencia económica del estudiante.
rango_edades_hijos	Integer	Representa el rango de edades de los hijos del estudiante, que está dado por 0-5.
rango_edades_hijos1	Integer	Representa el rango de edades de los

		hijos del estudiante que está dado por 5-10.
rango_edades_hijos2	Integer	Representa el rango de edades de los hijos del estudiante que está dado por 10-15.
rango_edades_hijos3	Integer	Representa el rango de edades de los hijos del estudiante que está dado por 15 +
factibilidad_de_atencion	Varchar	Representa la factibilidad de atención del estudiante a sus hijos en caso de que tenga.
clase_de_estudiante	Varchar	Representa la clase del estudiante en cuanto a su forma de estudio.
indice_academico	Float	Representa el índice académico con que el estudiante entro a la carrera.
valor_escalafon	Float	Representa el valor del escalafón con que el estudiante obtuvo la carrera.
Zona	Varchar	Representa el calificativo del tipo de zona donde vive el estudiante este puede ser Rural, Urbana etc.
concepto_ingreso	Varchar	Representa el concepto del ingreso del estudiante a la educación superior o por que vía obtuvo la carrera.
situacion_defensa	Varchar	Representa la situación de la defensa del estudiante.
id_cod_via_ingreso	Integer	Representa el código de la vía de ingreso del estudiante. Este atributo es llave foránea que absorbe la tabla de la tabla nomen_via_ingreso.
id_cod_situacion_escolar	Integer	Representa el código de la situación

		escolar del estudiante. Este atributo es llave foránea que absorbe la tabla de la tabla nomen_situacion_escolar.
id_cod_centro_procedencia	Integer	Representa el código del centro de procedencia del estudiante. Este atributo es llave foránea que absorbe la tabla de la tabla nomen_centro_procedencia.
id_cod_muni_resid	Integer	Representa el código del municipio de residencia del estudiante. Este atributo es llave foránea que absorbe la tabla de la tabla nomen_muni_residencia.

Nombre: enfermedades_x_estudiante		
Descripción: Esta tabla contiene el identificador del estudiante y lo relaciona con el identificador de enfermedades si es que el estudiante tiene alguna enfermedad.		
Atributo	Tipo	Descripción
id_cod_estudiante	Varchar	Representa el identificador o No.CI del estudiante. Este atributo es llave foránea que la absorbe de la tabla estudiante_g.
id_cod_enfermedades	Integer	Representa el identificador de la enfermedad. Este atributo es llave foránea que la absorbe de la tabla nomen_enfermedades.

Nombre: habitos_toxicos_x_estudiante		
Descripción: Esta tabla contiene el identificador del estudiante y lo relaciona con el identificador de hábitos tóxicos si es que el estudiante tiene algún habito toxico de los existentes en el codificador nomen_habitos_toxicos.		

Atributo	Tipo	Descripción
id_cod_estudiante	Varchar	Representa el identificador o No.CI del estudiante. Este atributo es llave foránea que la absorbe de la tabla estudiante_g.
id_cod_habitos_toxicos	Integer	Representa el identificador del hábito tóxico. Este atributo es llave foránea que la absorbe de la tabla nomen_habitos_toxicos.

Nombre: matricula_estudiante		
Descripción: Esta tabla contiene la información referente a la matrícula del estudiante.		
Atributo	Tipo	Descripción
id_cod_matricula	Integer	Representa el identificador de la matrícula.
fecha_matricula	Date	Representa la fecha en la que el estudiante matricula.
ano_academico	Integer	Representa el año académico en que el estudiante matricula.
fecha_graduado	Date	Representa la fecha en la que el estudiante se debe graduar.
Tipo_Curso	Varchar	Representa el tipo de curso que el estudiante matricula, este puede ser, Regular diurno, Por encuentro o Curso por trabajadores.
id_cod_estudiante	Varchar	Representa el identificador o No.CI del estudiante. Este atributo es llave foránea que la absorbe de la tabla estudiante_g.
id_cod_estudios	Integer	Representa el identificador del estudio que matricula el estudiante. Este atributo es llave foránea que la absorbe de la tabla nomen_estudios.

id_cod_perfil	Integer	Representa el identificador del perfil que matricula el estudiante. Este atributo es llave foránea que la absorbe de la tabla nomen_perfil.
---------------	---------	---

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

Nombre: nomen_centro_procedencia		
Descripción: Esta tabla contiene la información referente al codificador de los centros de procedencia de los cuales un estudiante obtuvo la carrera.		
Atributo	Tipo	Descripción
id_cod_centro_procedencia	Integer	Representa el identificador del centro.
nombre_centro_procedencia	Varchar	Representa el nombre del centro.
id_cod_muni_resid	Integer	Representa el código del municipio donde se encuentra el centro. Este atributo es llave foránea que absorbe de la tabla nomen_muni_residencia.

Nombre: nomen_consejo_popular		
Descripción: Esta tabla contiene la información referente al codificador de consejos populares.		
Atributo	Tipo	Descripción

id_cod_consejo_p	Integer	Representa el identificador del consejo popular.
nombre_consejo_popular	Varchar	Representa el nombre del consejo popular.
id_cod_muni_resid	Integer	Representa el código del municipio donde se encuentra el consejo popular. Este atributo es llave foránea que absorbe de la tabla nomen_muni_residencia.

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

Nombre: nomen_escuela		
Descripción: Esta tabla contiene la información referente al codificador de los centros docentes donde se estudian las carreras del área de salud.		
Atributo	Tipo	Descripción
id_cod_escuela	Integer	Representa el identificador del centro docente.
nombre_escuela	Varchar	Representa el nombre del centro docente.
Instituto	Varchar	Representa el nombre del instituto en caso de que el centro docente pertenezca a un instituto de medicina.
id_cod_muni_resid	Integer	Representa el código del municipio donde se encuentra el centro docente. Este atributo es llave foránea que absorbe de la tabla

		nomen_muni_residencia.
Nombre: nomen_estudio		
Descripción: Esta tabla contiene la información referente al codificador de las carreras que se estudian en medicina.		
Atributo	Tipo	Descripción
id_cod_estudios	Integer	Representa el identificador de la carrera.
nombre_estudios	Varchar	Representa el nombre de la carrera.
descripcion_estudio	Varchar	Representa una breve descripción de la carrera.
duracion_estudios	Integer	Representa el tiempo de duración la carrera.

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

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

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

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

Nombre: nomen_movim_socio_politico		
Descripción: Esta tabla contiene la información referente al codificador de movimientos sociopolíticos del extranjero.		
Atributo	Tipo	Descripción
id_cod_movim_socio_p	Integer	Representa el identificador de los movimientos sociopolíticos.
nombre_mov_socio_politico	Varchar	Representa el nombre de los movimientos sociopolíticos.

Nombre: nomen_muni_resid		
Descripción: Esta tabla contiene la información referente al codificador de los municipios de Cuba.		
Atributo	Tipo	Descripción
id_cod_muni_resid	Integer	Representa el identificador del municipio.
nombre_municipio	Varchar	Representa el nombre del municipio.
id_cod_prov_resid	Integer	Representa el código de la provincia a la cual pertenece el municipio. Este atributo es llave foránea que absorbe de la tabla nomen_prov_residencia.

Nombre: nomen_ocupacion_laboral		
Descripción: Esta tabla contiene la información referente al codificador de ocupación laboral.		
Atributo	Tipo	Descripción
id_cod_ocupacion	Integer	Representa el identificador de la ocupación laboral.
nombre_ocupacion	Varchar	Representa el nombre de la ocupación laboral.

Nombre: nomen_organizaciones_politicas		
Descripción: Esta tabla contiene la información referente al codificador de organizaciones políticas.		
Atributo	Tipo	Descripción
id_cod_organizaciones_politicas	Integer	Representa el identificador de la organización política.
nombre_organizaciones_politicas	Varchar	Representa el nombre de la organización política.

Nombre: nomen_pais		
Descripción: Esta tabla contiene la información referente al codificador de países.		
Atributo	Tipo	Descripción
id_cod_pais	Integer	Representa el identificador del país.
idioma	Varchar	Representa el idioma del país.
Moneda	Varchar	Representa la moneda del país.
nombre_pais	Varchar	Representa el nombre del país.
nacionalidad	Varchar	Representa el nombre de la nacionalidad de los ciudadanos nacidos en el país.

Nombre: nomen_perfil_carrera		
Descripción: Esta tabla contiene la información referente al codificador de perfil de la las carreras de tecnologías de la salud.		
Atributo	Tipo	Descripción
id_cod_perfil	Integer	Representa el identificador del perfil.
nombre_perfil	Varchar	Representa el nombre del perfil.
id_cod_estudios	Integer	Representa el código de la carrera a la cual pertenece el perfil . Este atributo es llave foránea que absorbe de la tabla nomen_estudio.

Nombre: nomen_procedencia_social		
Descripción: Esta tabla contiene la información referente al codificador de procedencia social.		
Atributo	Tipo	Descripción
id_cod_proceden_social	Integer	Representa el identificador de la procedencia

		social.
nombre_procedencia_social	Varchar	Representa el nombre de la procedencia social.

Nombre: nomen_prov_origen		
Descripción: Esta tabla contiene la información referente al codificador de las provincias o estados de los países.		
Atributo	Tipo	Descripción
id_cod_prov_orig	Integer	Representa el identificador de la procedencia social.
nombre_provincia	Varchar	Representa el nombre de la procedencia social.
id_cod_pais	Integer	Representa el código del país al cual pertenece la provincia. Este atributo es llave foránea que absorbe de la tabla nomen_pais.

Nombre: nomen_prov_residencia		
Descripción: Esta tabla contiene la información referente al codificador de las provincias de Cuba.		
Atributo	Tipo	Descripción
id_cod_prov_resid	Integer	Representa el identificador de la provincia.
nombre_prov_resid	Varchar	Representa el nombre de la provincia.

Nombre: nomen_situacion_escolar		
Descripción: Esta tabla contiene la información referente al codificador situación escolar del estudiante.		

Atributo	Tipo	Descripción
id_cod_situacion_escolar	Integer	Representa el identificador de la situación escolar.
nombre_situacion_escolar	Varchar	Representa el nombre de la situación escolar.

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

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

Nombre: organizaciones_politicas_x_estudiante		
Descripción: Esta tabla representa la relación entre estudiantes y organizaciones políticas.		
Atributo	Tipo	Descripción
id_cod_organizaciones_pol	Integer	Representa el código de la organización

iticas		política a la que pertenece un estudiante. Este atributo es llave foránea que absorbe de la tabla nomen_organizaciones_politicas.
id_cod_estudiante	Integer	Representa el código del estudiante. Este atributo es llave foránea que absorbe de la tabla nomen_pais.

Nombre: padres		
Descripción: Esta tabla contiene la información de los padres del estudiante		
Atributo	Tipo	Descripción
id_cod_padres	Integer	Representa el identificador del padre.
Nombre	Varchar	Representa el nombre del padre.
segundo_nombre	Varchar	Representa el segundo nombre del padre en caso de que tenga nombre compuesto.
primer_apellido	Varchar	Representa el 1er apellido del padre.
segundo_apellido	Varchar	Representa el 2do apellido del padre.
id_cod_estudiante	Varchar	Representa el identificador o No.CI del estudiante. Este atributo es llave foránea que absorbe de la tabla estudiante_g.

Nombre: padres_organizaciones_politicas		
Descripción: Esta tabla contiene la información de las organizaciones políticas a las cuales pertenecen los padres del estudiante.		
Atributo	Tipo	Descripción
id_cod_padres	Integer	Representa el código del padre. Este atributo es llave foránea que absorbe de la

		tabla padres.
id_cod_organizaciones_politicas	Integer	Representa el código de la organización política a que pertenece el padre. Este atributo es llave foránea que absorbe de la tabla nomen_organizaciones_politicas.

Nombre: practica_religion		
Descripción: Esta tabla contiene la información referente a la creencias religiosas del estudiante		
Atributo	Tipo	Descripción
nombre_institucion	Varchar	Representa el nombre de la institución religiosa.
nombre_ministerio	Varchar	Representa el nombre del ministerio religioso.
nombre_corriente_religiosa	Varchar	Representa el nombre de la corriente religiosa.
nombre_iglesia	Varchar	Representa el nombre de la iglesia a la que asiste el estudiante.
nombre_movimiento_religioso	Varchar	Representa el nombre del movimiento religioso a que pertenezca el estudiante.
id_cod_estudiante	Varchar	Representa el identificador o No.CI del estudiante. Este atributo es llave foránea que la absorbe de la tabla estudiante_g.

Nombre: tallaje		
Descripción: Esta tabla contiene la información de las tallas del vestuario del estudiante		
Atributo	Tipo	Descripción
Blusa	Varchar	Representa la talla.

Falda	Varchar	Representa la talla.
Pantalón	Varchar	Representa la talla.
Zapato	Varchar	Representa la talla.
Camisa	Varchar	Representa la talla.
id_cod_estudiante	Varchar	Representa el identificador o No.CI del estudiante. Este atributo es llave foránea que la absorbe de la tabla estudiante_g.

3.4 Definiciones de diseño.

El diseño de la interfaz es otro de los puntos fundamentales a tratar a la hora de la presentación de la aplicación teniendo en cuenta que esa es la capa de presentación al usuario y por lo tanto debe ser lo más amigable y comprensible posible, por lo que la interfaz debe resultar coherente, sensata, necesaria y comprensible para el usuario corriente.

En este caso la aplicación está dirigida a profesionales de la salud que en ocasiones tienen un conocimiento mínimo del uso de la computadora, teniendo esto en cuenta, los esfuerzos de diseño están orientados a lograr una interfaz clara y fácil de usar, ya que a través de la misma es que se interactúa con los recursos que se ponen a disposición del usuario en el ordenador.

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

- El equilibrio en la organización de la información, por ejemplo, que todas las páginas que muestran información siempre la mostrarán en el mismo orden.
- La optimización de la cantidad de elementos en la pantalla, ayudando al fácil manejo y mejor comprensión de la información mostrada en pantalla.
- La unidad, donde cada elemento de la pantalla se diseñará siguiendo un patrón de tamaño, colores y formas, utilizando para ello hojas de estilos.
- Anticipar lo que el usuario quiere y necesita, brindándole toda la información y herramientas necesarias para cada paso del proceso.
- Lograr una autonomía, la aplicación y el entorno pertenecen al usuario, pero esto no significa que se abandonen las reglas ofreciéndole cierta amplitud de maniobrabilidad.
- Mantener un nivel alto de consistencia a través de toda la aplicación, reutilizando los criterios con que se diseñaron los íconos, diálogos, formularios, mensajes informativos, etc.

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

3.5 Tratamiento de errores.

Al tratamiento de errores en el sistema se le concede una especial importancia debido a que es un problema que a menudo no se tiene en cuenta en el desarrollo de las aplicaciones, lo que puede traer consigo que se proporcione más información de la necesaria al usuario. Esta circunstancia es muy común en las páginas de error de las aplicaciones, que suelen ser páginas por defecto que ofrecen información detallada acerca del error, información que puede ser aprovechada por el cliente para un uso malintencionado.

Debido a esto el tratamiento de errores en la aplicación está presente del lado del cliente, utilizando código Java Script combinado con el uso de expresiones regulares que brindan una fortaleza garantizando rapidez de ejecución del sistema ya que se garantiza que la información recibida por el servidor esté correcta, evitando la entrada incorrecta de la información. Del lado del servidor se le da un tratamiento a los errores de forma tal que las operaciones de inserción, eliminación y/o modificación de registros en la base de datos se realicen de forma correcta, estableciendo mecanismos que visualicen la información para evitar en la medida de lo posible la introducción manual de los datos y minimizar los errores que pueda tener la aplicación, siendo válida toda la información. En el caso de la inserción se implementarán funciones que validen la entrada de datos y en caso de existir errores, se mostrarán mensajes que aclaren la inserción o modificación errónea de algún dato, los mensajes emitidos por causas de infracción serán fácilmente configurables. Otros errores pueden ser generados por el Gestor de Base de Datos, estos se capturan antes de mostrarse al usuario y una vez tratados por el sistema, son mostrados al usuario de una manera entendible para él.

3.6 Seguridad.

El tema de la seguridad y protección del sistema es un punto elemental a tener en cuenta por las características de la información que se almacena y la relación congruente que existirá entre este sistema y los ya existentes en el Ministerio de Salud Pública. Con este objetivo el sistema cuenta con

un módulo de administración que se encarga de manejar la lógica del negocio tratado, implementando un modelo de autenticación, autorización y auditoría para los usuarios. Logrando un alto grado de:

- **Confiabledad:** La información manejada por el sistema está protegida de acceso no autorizado.
- **Integridad:** La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes.
- **Disponibilidad:** Los usuarios autorizados tendrán acceso a la información en todo momento.

En lo que respecta a la prevención de inyecciones de código malicioso en el sistema, que permita mostrar información a intrusos, se evita este tipo de ataque utilizando funciones propias de PHP combinadas con expresiones regulares y validaciones implementadas por los desarrolladores.

3.7 Estándar de codificación.

Las reglas o estándares de codificación son fundamentales para el desarrollo de cualquier software, ya que con el tiempo, el código bien escrito y comentado y que además guarde siempre el mismo estilo permite el reconocimiento de las funciones y la lectura comprensiva del código. Para definir el estilo de codificación a seguir en la aplicación se utilizó la notación estándar establecida para aplicaciones desarrolladas en PHP .

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

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

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

Se comentará entre etiquetas `/**/` y `//` la funcionalidad de las clases y funciones definidas en el código justificando su uso para una mayor comprensión del código. Además el nombre debe ser lo más descriptivo posible y se debe evitar el uso de abreviaturas.

3.8 Concepción de la ayuda.

Debido a que, como se ha reflejado anteriormente, los usuarios del sistema generalmente tienen poca experiencia en el uso de aplicaciones de este tipo, se confeccionará una ayuda que explicará de forma detallada cómo funciona el sistema para cada uno de los niveles de acceso. Cada página muestra cómo realizar aquellas operaciones que estén relacionadas con la posición donde se encuentre el usuario en dicho momento.

El sistema cuenta con un módulo de mensajería que permitirá a los usuarios realizar cualquier tipo de preguntas y sugerencias a los administradores, lo cual facilitará la comunicación entre los usuarios del sistema, de esta forma la ayuda resulta de gran ventaja, ya que contribuye a la resolución de problemas en el software, la gestión de cambios y configuraciones y la actualización y el mantenimiento del producto.

Conclusiones.

Con el desarrollo de este capítulo se ha logrado modelar todo el sistema que ha sido objeto de estudio durante el transcurso del trabajo. Se realizó la etapa de análisis y diseño, obteniendo diagramas de clases del análisis, donde se representó un diagrama de clases inicial, mostrando las relaciones que se establecen entre las clases interfaz, controladoras y entidad. En la etapa de diseño quedaron definidos los diagramas de interacción, los que se elaboraron a partir de artefactos de la etapa de análisis. Esto permitió asignar responsabilidades a las clases. También se obtiene el diagrama de clases del diseño, que por ser una aplicación Web, se utilizaron estereotipos destinados para ello.

Además se realizaron los diagramas de diseño de la base de datos dando una explicación detallada de cada una de sus tablas y campos y se explican cuestiones del diseño de la interfaz, tratamiento de los errores, seguridad del sistema y la concepción de la ayuda para operar el sistema.

CAPÍTULO 4: IMPLEMENTACIÓN.

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

4.1 Diagrama de despliegue.

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

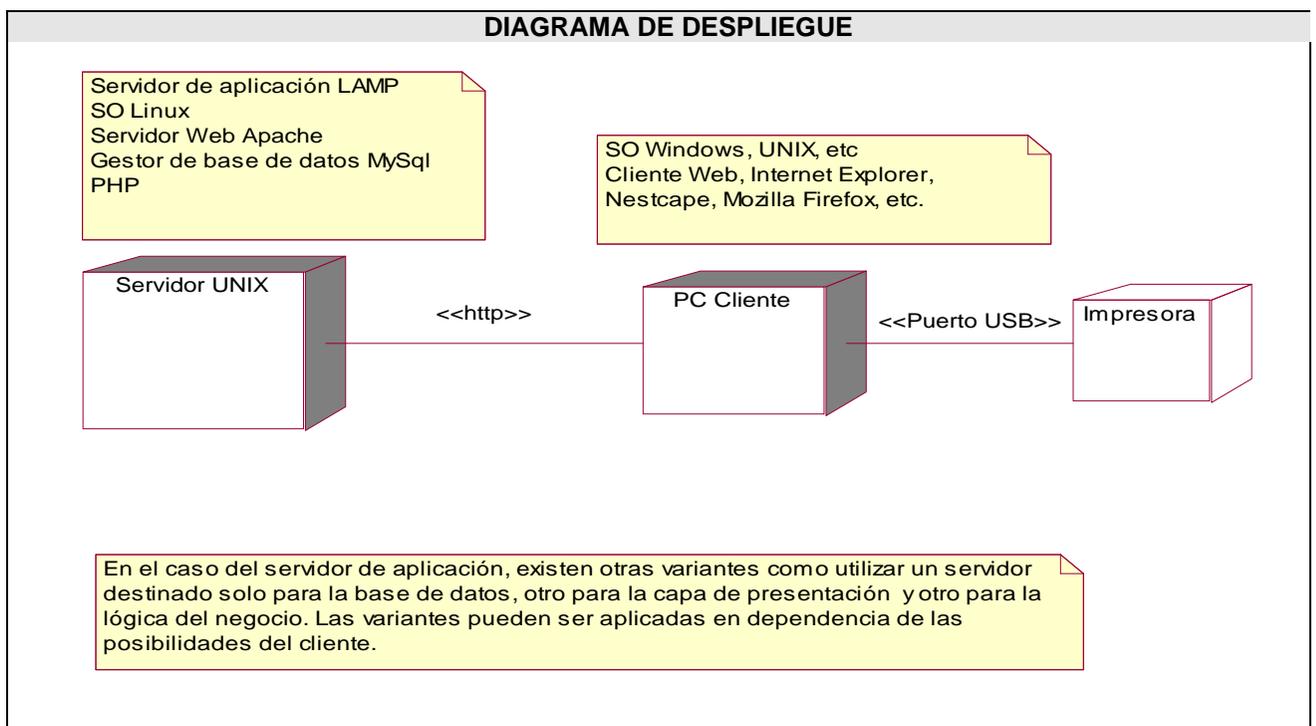


Fig. 18 Diagrama de despliegue.

4.2 Diagrama de componentes.

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. Con el objetivo de lograr un mayor entendimiento se realizó un diagrama de componentes general de la aplicación y posteriormente se dividió en subsistemas de implementación, usados para estructurar el modelo de implementación y dividirlos en pequeñas partes que pueden ser integradas y probadas de forma separada, con el objetivo de lograr un mayor entendimiento.

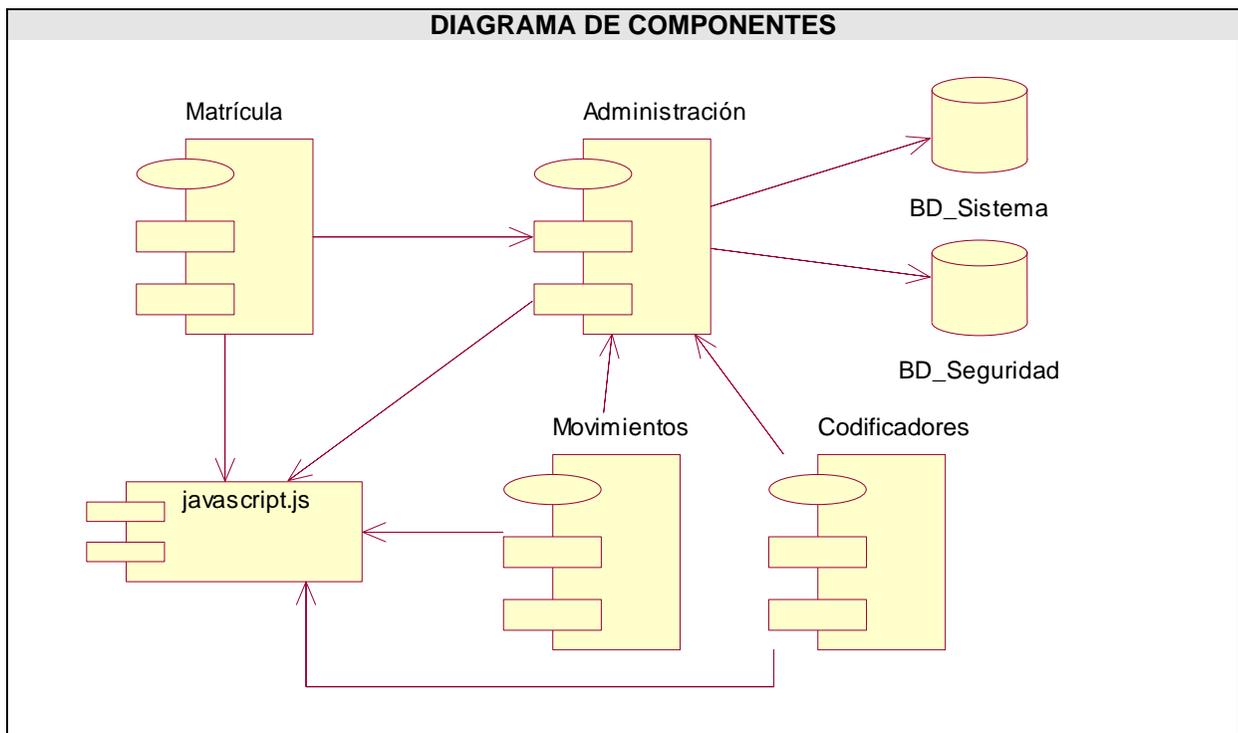


Fig. 19 Diagrama de componentes.

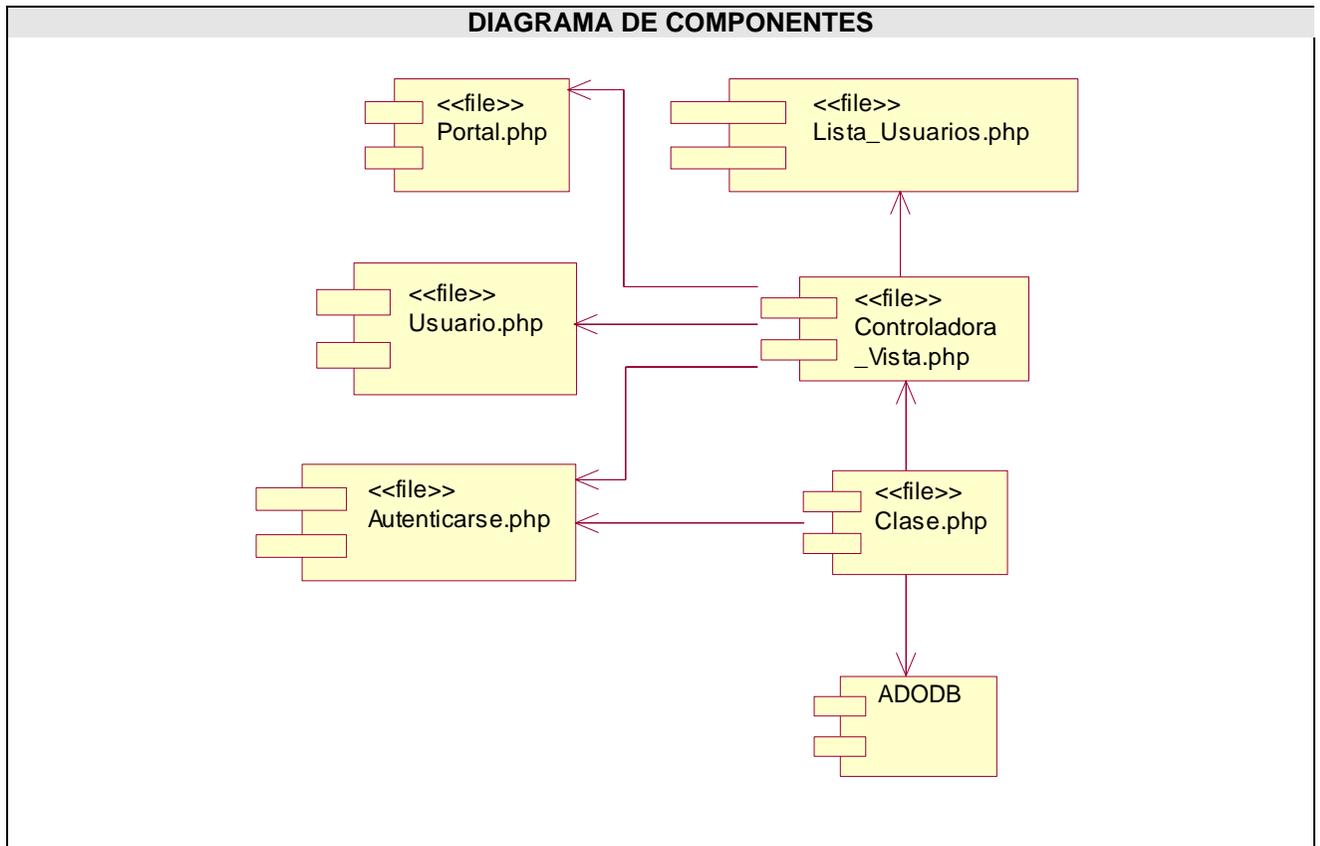


Fig. 20 Diagrama de componentes Administración.

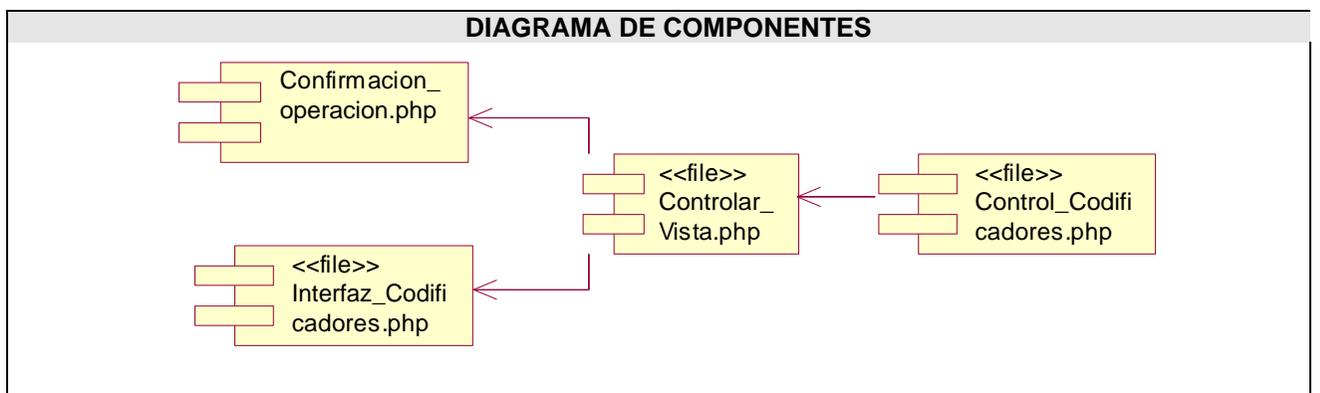


Fig. 21 Diagrama de componentes Codificadores.

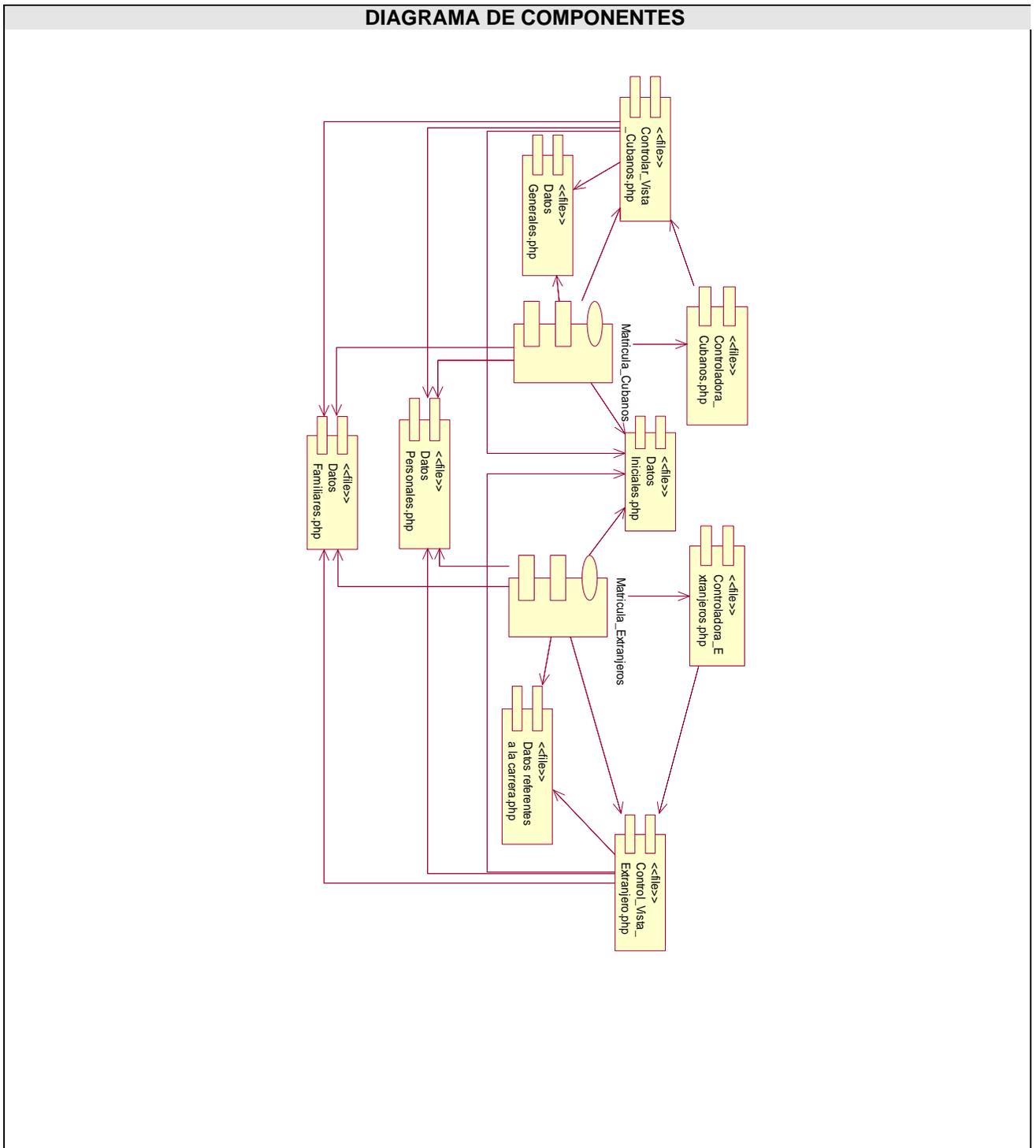


Fig. 22 Diagrama de componentes Matrícula.

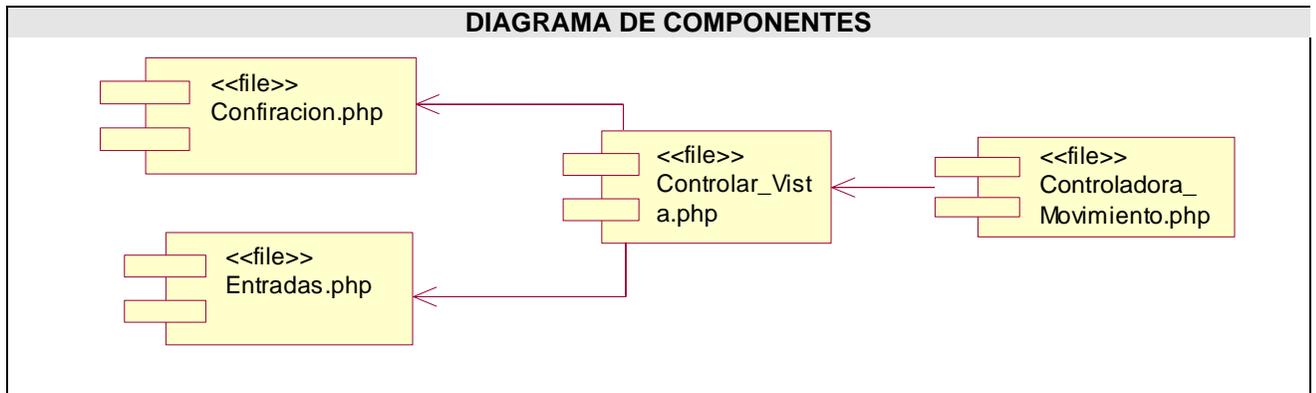


Fig. 23 Diagrama de componentes Movimientos.

Conclusiones.

En este capítulo se desarrolló el flujo de trabajo de implementación donde se muestran los diagramas de despliegue que modelan el sistema desarrollado, los diagramas de componentes, mostrando la organización del código fuente de cada uno de ellos.

CONCLUSIONES.

Una vez concluida la investigación sobre el proceso de formación de recursos humanos en el MINSAP y el desarrollo del Sistema de Información en el Proceso de Formación de Recursos Humanos en Salud, se ha dado cumplimiento a los objetivos planteados obteniéndose los resultados que a continuación se mencionan:

- Se analizaron los aspectos teóricos conceptuales planteados por el Ministerio de Salud Pública relacionados con el proceso de formación de recursos humanos obteniendo una concepción del negocio y los requisitos para el desarrollo de la solución.
- Se realizó un estudio de las principales tendencias y tecnologías informáticas actuales para el desarrollo de la aplicación definiendo como entorno de desarrollo la tecnología LAMP (SO *Linux*, Servidor Web *Apache*, Gestor de BD *MySQL* y lenguaje de programación *PHP*) utilizando el motor de plantilla *Smarty* y tecnología *AJAX*; además se definió una arquitectura en tres capas según el modelo Cliente Servidor, implementando el patrón Modelo Vista Controlador . Se define RUP (*Proceso de desarrollo unificado*) como metodología de desarrollo.
- Se realizó el diseño e implementación del sistema que permite controlar la gestión de la información relacionada con la matrícula y movimientos de estudiantes en el proceso de formación de recursos humanos en el MINSAP.

RECOMENDACIONES.

Como recomendaciones al trabajo se tienen:

- A la dirección de la Facultad que se continúe con el desarrollo del proyecto, pues el mismo es de vital importancia para el proceso de formación de recursos humanos para la salud que desarrolla el país actualmente como uno de los programas priorizados.
- A la dirección del proyecto que se le dé cumplimiento a la implementación de los casos de uso del segundo ciclo de desarrollo y lograr una integración con otros sistemas de gestión existentes en el MINSAP como el Registro de Unidades de Salud, Registro de Profesionales de Salud, etc.

BIBLIOGRAFÍA.

- 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.
- FRANCO NAVARRO, J. A. *UML en acción. Modelando Aplicaciones Web*, 2005.
- GALLEGO VÁZQUEZ, J. A. *Desarrollo Web con PHP y MySQL*. S.A, A. M., 2003.
- GILFILLAN, I. *La Biblia MySQL.*, 2002.
- HANSEN, G. W. H., JAMES V. . *Diseño y Administración de Bases de Datos* HALL., P.
- LARMAN, C. *UML y Patrones*. Prentice Hall Iberoamericana. 1999. p.
- Manual de Procedimientos de Secretarías Generales*
- Manual MySQL Server 5*. 2006. [Disponible en: <http://www.lawebdelprogramador.com>
- Modelo de matrícula*. MES.
- 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.
- 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.
- ZELDMAN, J. *Diseño con estándares Web* S.A., A. M.

REFERENCIAS BIBLIOGRÁFICAS.

1. *Aplicación web*. Disponible en: http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web
2. SANZ, M. Á. *A, B, C de Internet*, 13/02/2002.
3. *Internet* Disponible en: <http://es.wikipedia.org/wiki/Internet>
4. *Funcionamiento de la Web*. Wikilearning Disponible en:
http://www.wikilearning.com/funcionamiento_de_la_web-wkccp-3443-6.htm
5. CLEMENTS, P. *Coming attractions in Software Architecture*, 1996.
6. KICILLOF, C. R.-N. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*, 2004.
Disponible en: <http://www.microsoft.com/spanish/msdn/arquitectura>
7. Ídem a la referencia 6.
8. Ídem a la referencia 6.
9. VALLE, J. G. *Arquitectura Cliente/Servidor*, 2005.
10. *Servidor web*. 2007. [Disponible en: http://es.wikipedia.org/wiki/Servidor_web
11. *Servidor HTTP Apache*. 2007. [Disponible en:
http://es.wikipedia.org/wiki/Servidor_HTTP_Apache
12. Ídem a la referencia 6.
13. Ídem a la referencia 11.
14. RODRÍGUEZ, J. T. Q. *Sistemas Distribuidos*
15. *Sistema de gestión de base de datos*. 2007. [Disponible en:
http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_base_de_datos
16. *Sistema gestores de BD*. 2007. [Disponible en:
http://es.wikipedia.org/wiki/Sistemas_gestores_de_bases_de_datos#Objetivos
17. *HTML*. 2007. [Disponible en: <http://es.wikipedia.org/wiki/HTML>
18. *JavaScript*. 2007. [Disponible en: <http://es.wikipedia.org/wiki/JavaScript>
19. *AJAX*. 2007. [Disponible en: <http://es.wikipedia.org/wiki/AJAX>
20. *PHP*. 2007. [Disponible en: <http://es.wikipedia.org/wiki/PHP>
21. *Smarty*. 2007. [Disponible en: <http://es.wikipedia.org/wiki/Smarty>
22. *Rational Suite*. 2005. [Disponible en: <http://www.vico.org/TallerRationalRose.pdf>
23. *Control de versiones*. 2007. [Disponible en: http://es.wikipedia.org/wiki/Control_de_versiones

24. TortoiseSVN. 2007. [Disponible en: <http://es.wikipedia.org/wiki/TortoiseSVN>]

ANEXOS #1: MODELO DE NEGOCIO.

Actores del Negocio.

Actores.	Justificación.
Estudiante.	Interesado en que se le procesen los datos personales para matricular como estudiante.
Profesor.	Interesado en procesar los resultados docentes alcanzados por el estudiante.
Solicitante.	Interesado en hacer una solicitud de baja o licencia de matrícula.
Dirección de la Institución.	Interesado en hacer una solicitud de baja de un estudiante por motivos académicos o por indisciplinas.

Trabajadores del Negocio.

Trabajadores.	Justificación.
Secretario Docente.	Encargado de realizar los procesos en la secretaría docente.
Controlador de Expediente.	Encargado de llevar el control de los expedientes en la secretaría docente.

Diagrama de Casos de Uso del Negocio.

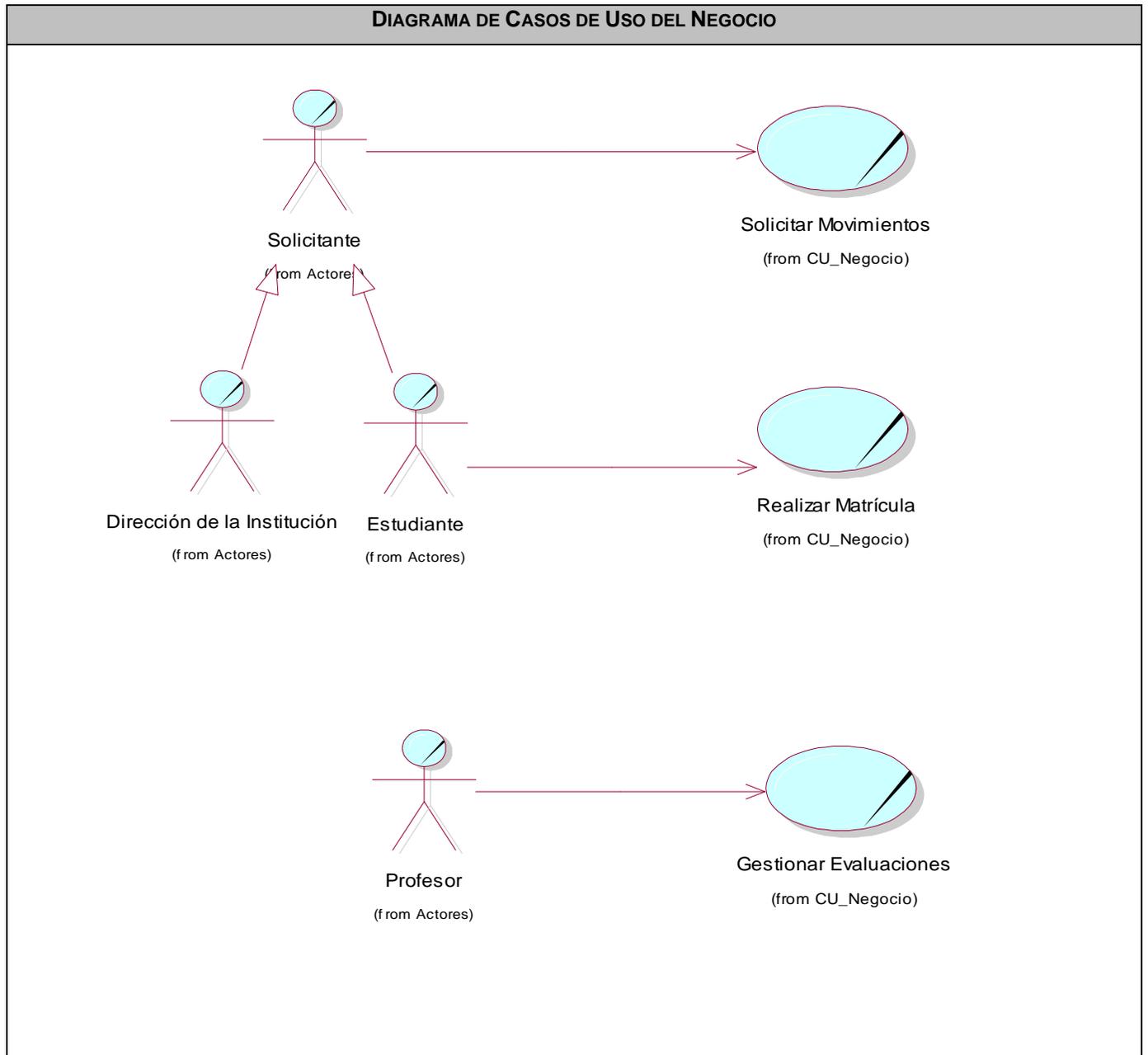


Fig. 1 Diagrama de Casos de Uso del Negocio.

Especificación de los Casos de Uso.

Caso de Uso:	Realizar Matrícula.	
Actores:	Estudiante (inicia).	
Trabajadores:	Secretario Docente.	
Resumen:	El caso de uso se inicia cuando el estudiante se presenta en la secretaría docente de la institución para realizar la matrícula. El secretario docente le entrega una planilla donde se recoge un grupo de datos y la documentación necesaria. El caso de uso termina cuando el secretario docente archiva los datos y crea el expediente académico.	
Precondiciones:	El estudiante ha obtenido la carrera y presenta la documentación necesaria.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1 El estudiante se presenta en la secretaría docente para matricularse.	2 El secretario docente revisa en los listados oficiales si aparece el nombre del aspirante a matricular.	
	3 Solicita al aspirante los documentos establecidos para efectuar la matrícula.	
4 El estudiante entrega la documentación solicitada.	5 El secretario docente verifica que el estudiante tenga la documentación requerida y que la misma esté correcta.	
7 El estudiante responde el cuestionario de preguntas.	6 El secretario docente entrevista al estudiante para hacer la captación de datos y llenar la planilla de matrícula.	

	8 El secretario docente archiva la planilla de matrícula del estudiante y crea el expediente académico del mismo.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	2.1 Si el estudiante no se encuentra en los listados oficiales el Secretario Docente le informa que no puede matricular. 5.1 Si el estudiante no tiene la documentación completa el secretario docente le informa que no puede ser matriculado.
Poscondiciones	El estudiante se encuentra matriculado.
Mejoras	Agilizar el proceso de matrícula y de gestión de la información en el proceso de formación de recursos humanos.
Prioridad	Alta

Caso de Uso:	Gestionar Evaluaciones
Actores:	Profesor (inicia).
Trabajadores:	Secretario Docente.
Resumen:	El caso de uso se inicia cuando el profesor de la institución necesita entregar las evaluaciones finales de los estudiantes a la secretaría docente para que sean archivadas. En la secretaría docente se le entrega al profesor un modelo donde registrará las notas del grupo o brigada docente al que le da clases. Este caso de uso concluye cuando el secretario docente archiva los resultados docentes del estudiante entregados por el profesor.
Precondiciones:	El estudiante se encuentra matriculado en la carrera y se ha examinado.

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1 El profesor solicita en la secretaría docente el acta de examen e informe de evaluación final del estudiante.	2 El secretario docente le entrega el acta de examen y evaluación final. 4 El secretario docente archiva el acta de examen e informe de evaluación final.
3 El profesor llena el acta con las evaluaciones de los estudiantes del grupo al que le imparte clases y lo entrega al secretario docente.	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
Poscondiciones	El profesor tenga las evaluaciones de los estudiantes.
Mejoras	Conocer de forma rápida y precisa el estado académico de un estudiante o de los estudiantes de un grupo o brigada docente.
Prioridad	Media.

Caso de Uso:	Solicitar Movimiento.
Actores:	Solicitante (Inicia).
Trabajadores:	Secretario Docente.
Resumen:	El caso de uso se inicia cuando el estudiante o la dirección de la institución hacen una solicitud de trámite. Posteriormente el secretario docente informa la solicitud a los diferentes niveles. Este caso de uso concluye cuando el secretario docente informa al solicitante del trámite la aprobación o no del mismo.

Precondiciones:	Se realice solicitud.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1 El Solicitante presenta la solicitud del trámite.	2 Recepciona la solicitud. 3 Informa la solicitud del estudiante al Controlador de Expedientes, para pasar el expediente al archivo pasivo. 4 Informa la solicitud del estudiante al Gestor Económico para la actualización de los listados y la comunica al Departamento Docente correspondiente y al Departamento Económico del centro. 5 Informa al estudiante de la aprobación o no de su solicitud de trámite.	
Flujos Alternos		
Acción del Actor	Respuesta del Negocio	
Poscondiciones	Procesamiento de la solicitud.	
Mejoras	Agilizar el proceso de movimientos, traslados y licencia de matrícula tanto internos como entre centros docentes.	
Prioridad	Alta.	

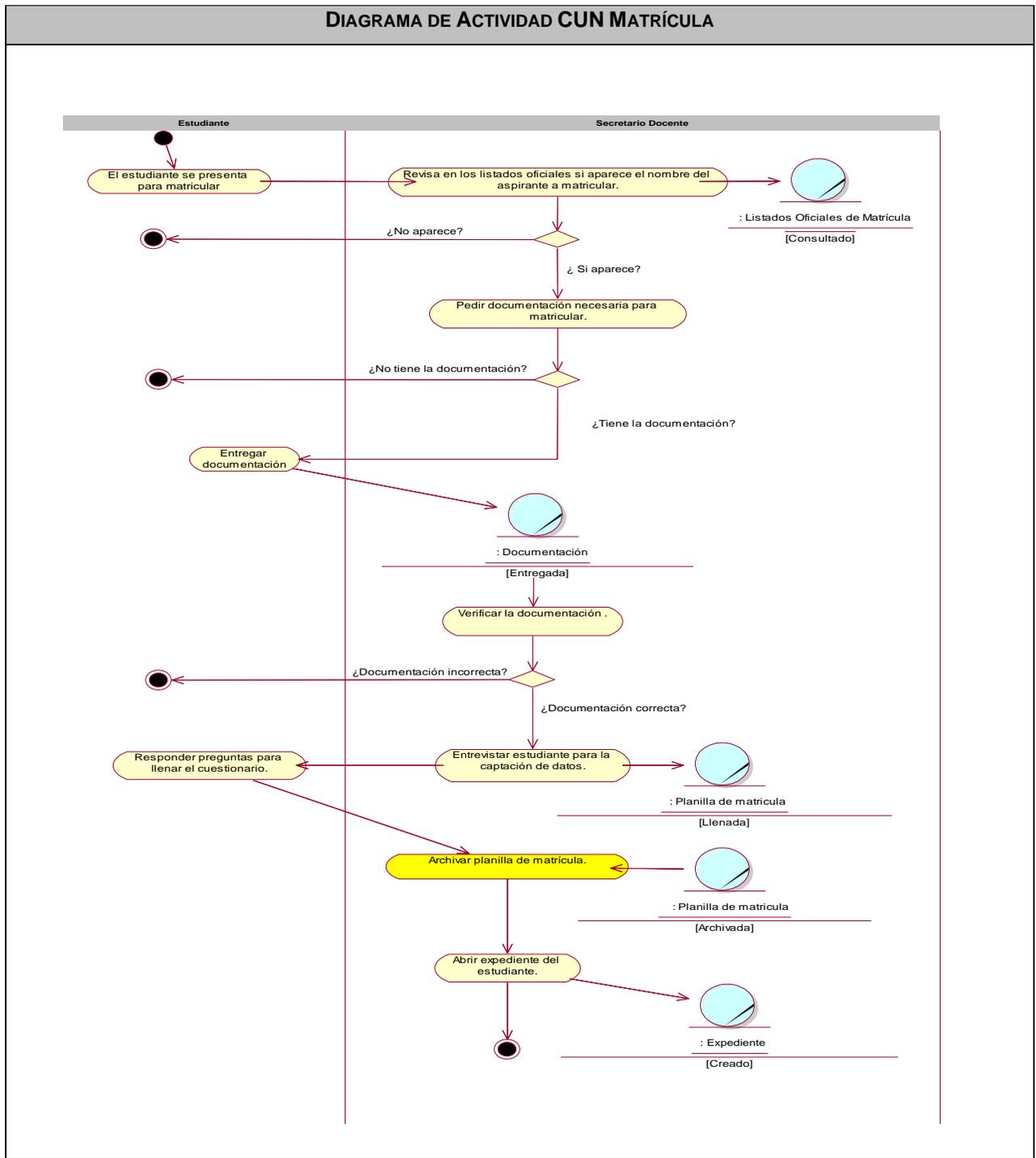


Fig. 2 Diagrama de Actividad Caso de Uso Matricula.

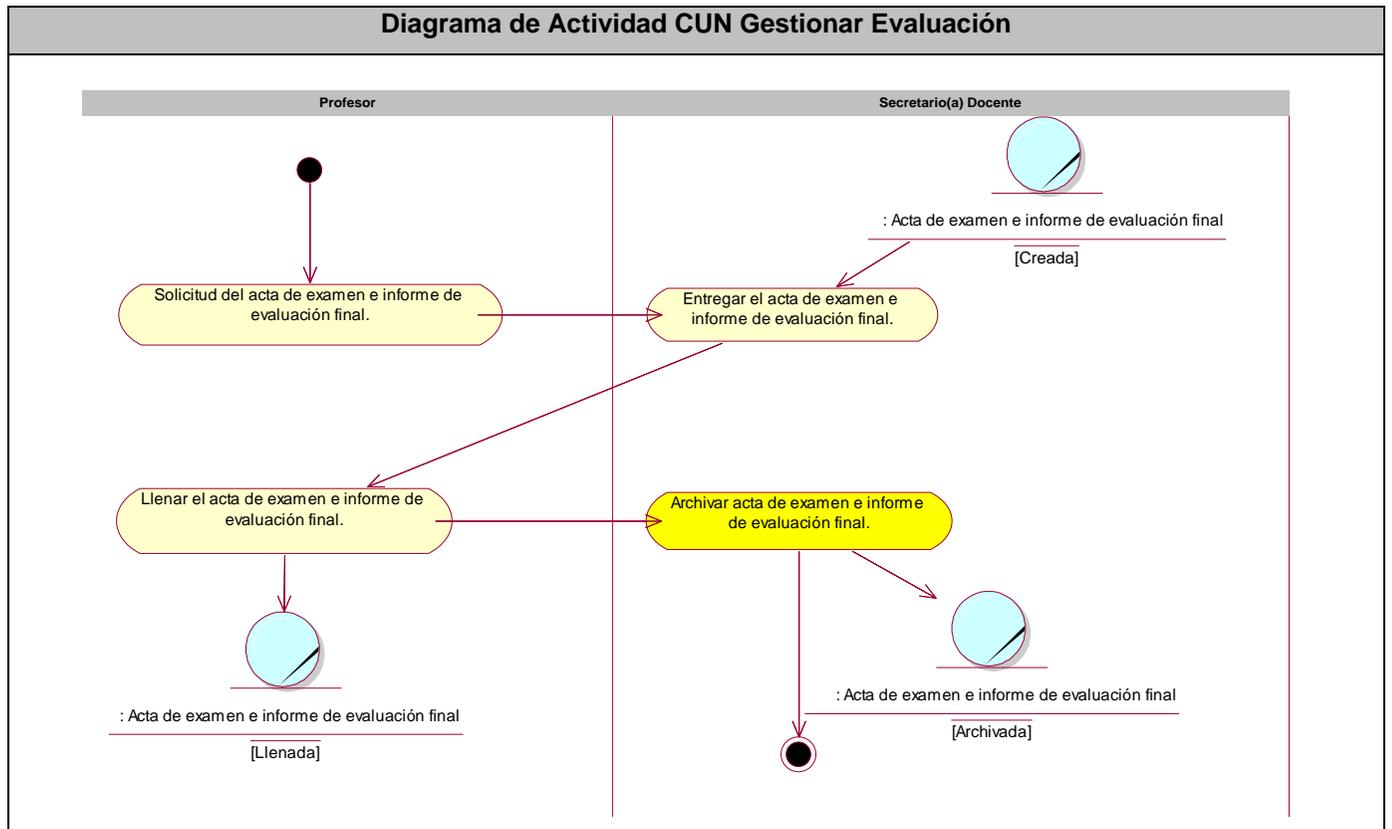


Fig. 3 Diagrama de Actividad Caso de Uso Gestionar Evaluación.

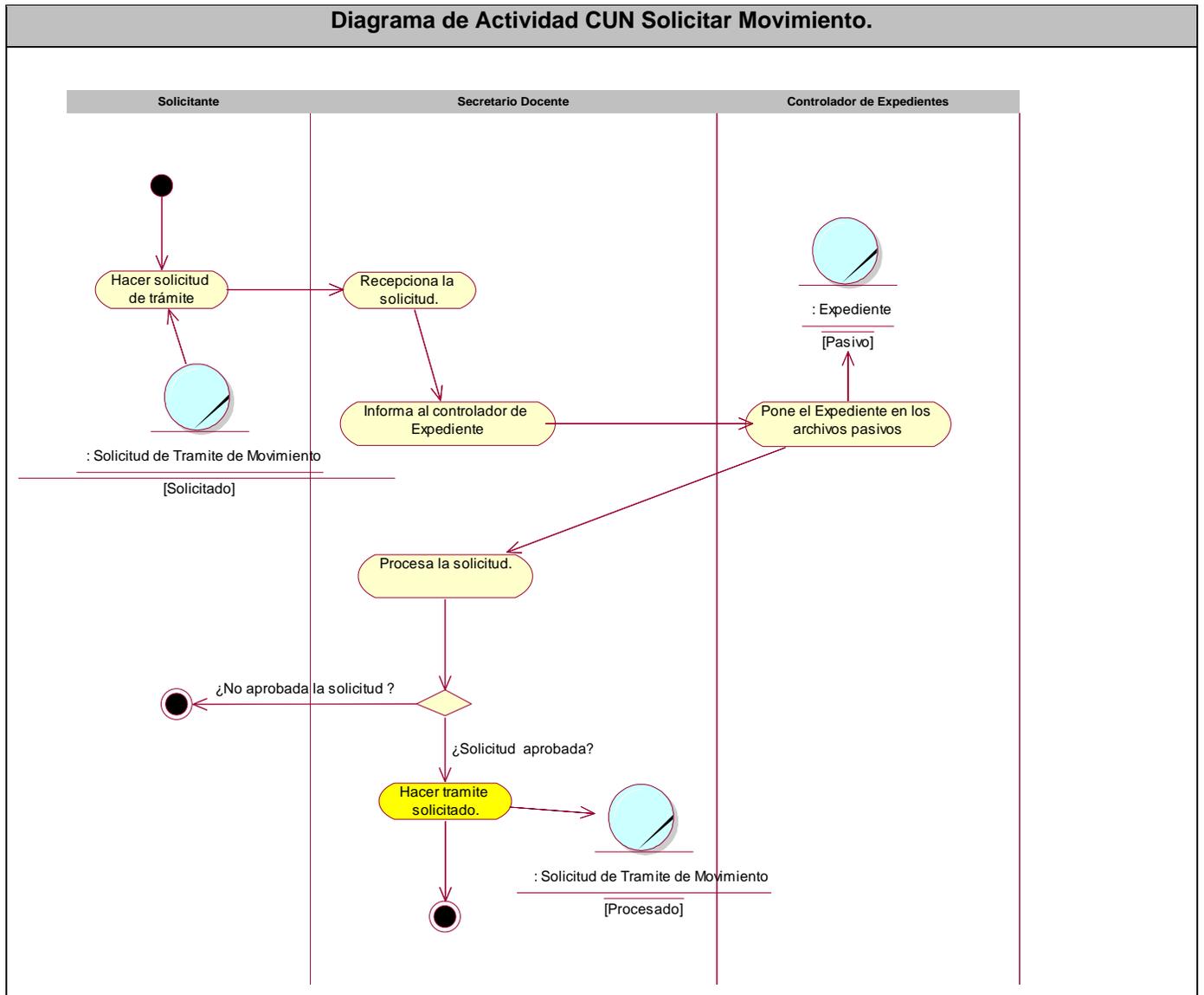


Fig. 4 Diagrama de Actividad Caso de Uso Solicitar Movimiento.

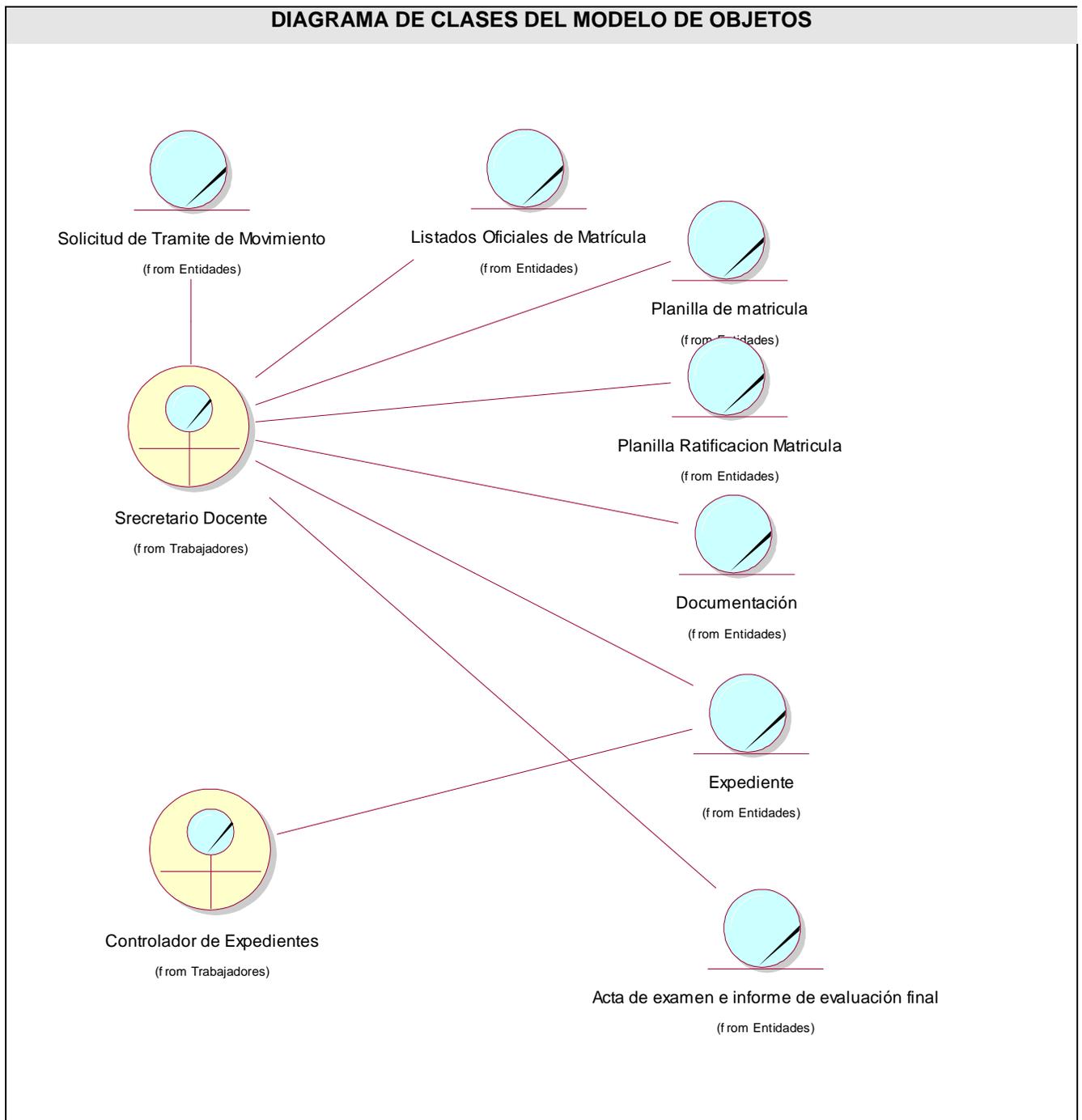


Fig. 5 Diagrama de Modelo de Objetos del Negocio.

ANEXOS #2: MODELO DE SISTEMA.

Definición de los 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 Sistema de Gestión de Información en el Proceso de Formación de Recursos Humanos en Salud.

Actores del Sistema

Los actores representan terceros fuera del sistema que intercambian información con éste. En la Tabla 3.1 se muestran los actores del sistema.

Actores	Justificación
Administrador Nacional.	Controla la configuración del sistema, define los usuarios del mismo a nivel provincial, así como los niveles de acceso de estos, además gestiona los codificadores.
Administrador Provincial.	Controla la configuración del sistema, define los usuarios del mismo a nivel de institución docente.
Editor de Sede	Secretario docente de los centros de estudios, es el responsable de ingresar nuevos estudiantes al sistema. Solo tiene acceso a la información de su centro.
Visualizador Nacional	Accede a los informes que brinda el sistema (Listados y resúmenes)
Visualizador Provincial	Accede a los informes que brinda el sistema (Listados y

	resúmenes) a nivel provincial.
Usuario	Actor abstracto que generaliza los roles para la autenticación al sistema.

Casos de Uso del Sistema.

Dado el número de casos de uso y actores, se introducen paquetes en el modelo de casos de uso para organizar los elementos en grupos y hacerlo más comprensibles.

En la Fig.1 se muestra el diagrama de casos de uso por paquetes (Actores; Administración; Matrícula; Gestión Académica; Informes).

El paquete Actores (Fig. 2) contiene los actores del sistema.

El paquete Administración (Fig. 3) agrupa los casos de usos relacionados con operaciones de administración y gestión de datos de los codificadores que se usan en el sistema.

El paquete Matrícula (Fig. 4) contiene los casos de uso relacionados con el proceso de la matrícula.

El paquete Gestión Académica (Fig. 5) contiene los casos de uso relacionados con las actividades docentes durante la Carrera.

El paquete Informes (Fig. 6) contiene los casos de uso relacionados con los informes que brinda el sistema (resúmenes y listados).

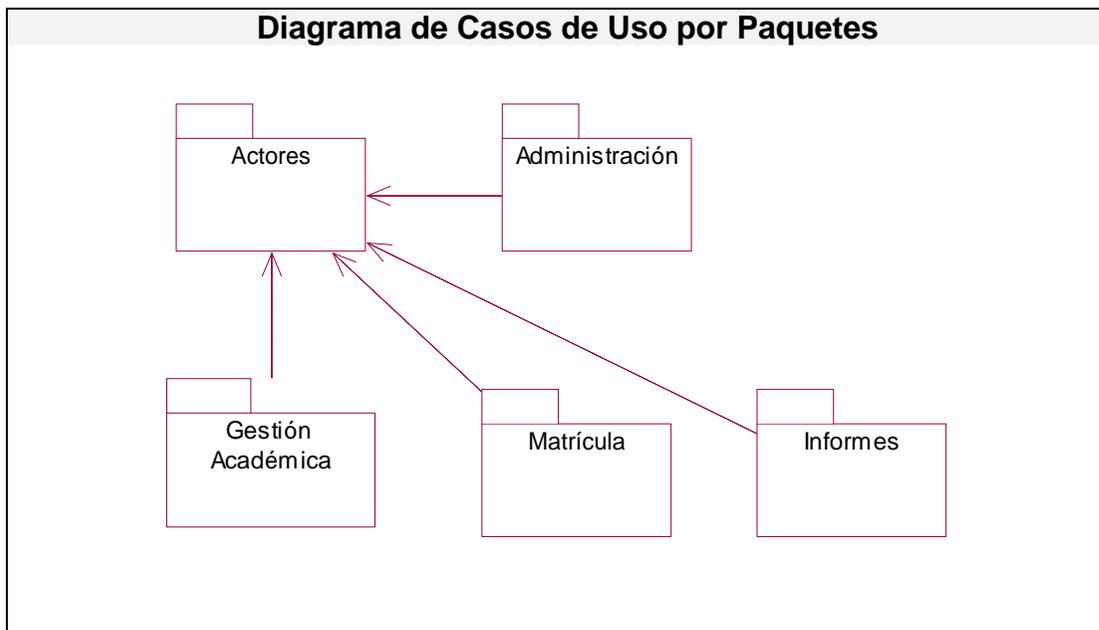


Fig. 1 Paquetes de Casos de Uso. Relación entre Paquetes.

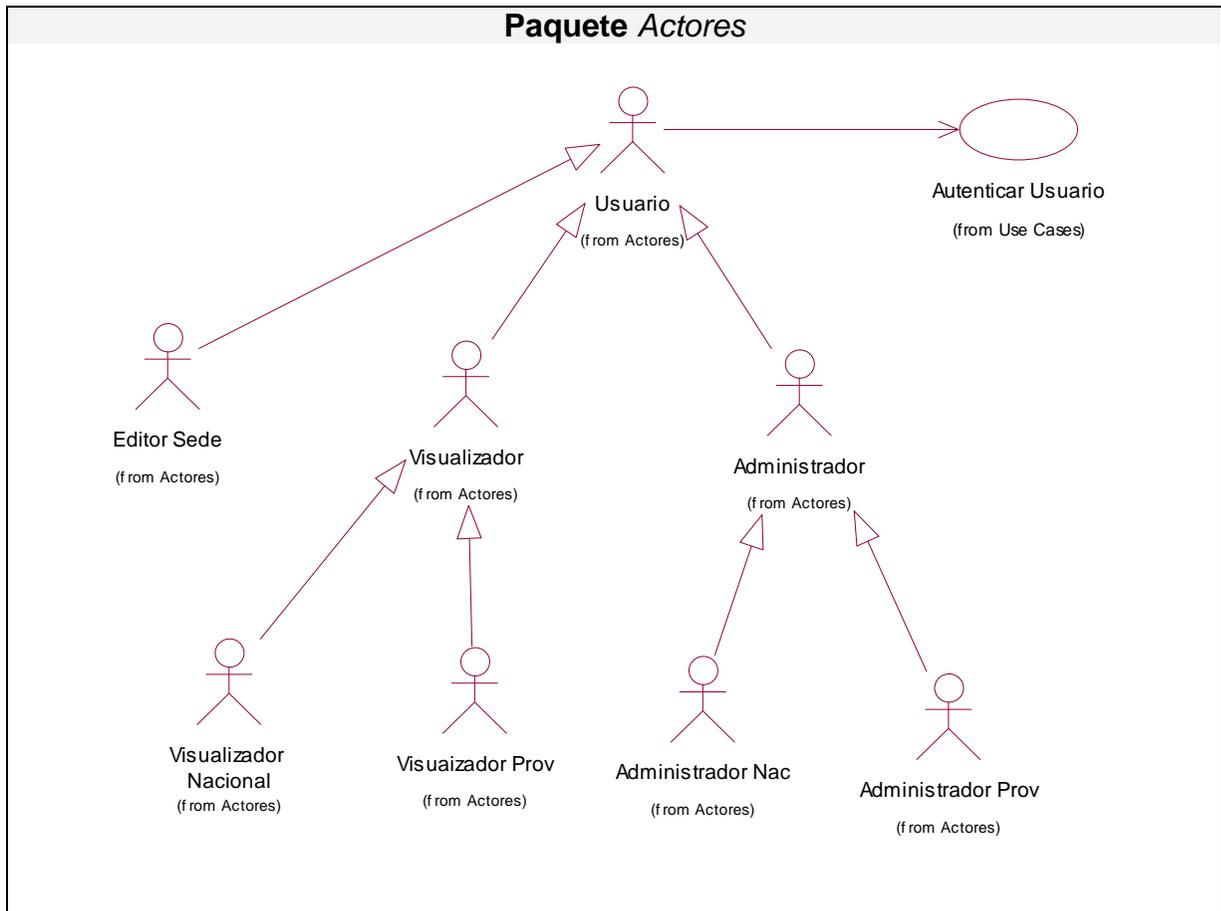


Fig. 2 Paquete "Actores" del Sistema.

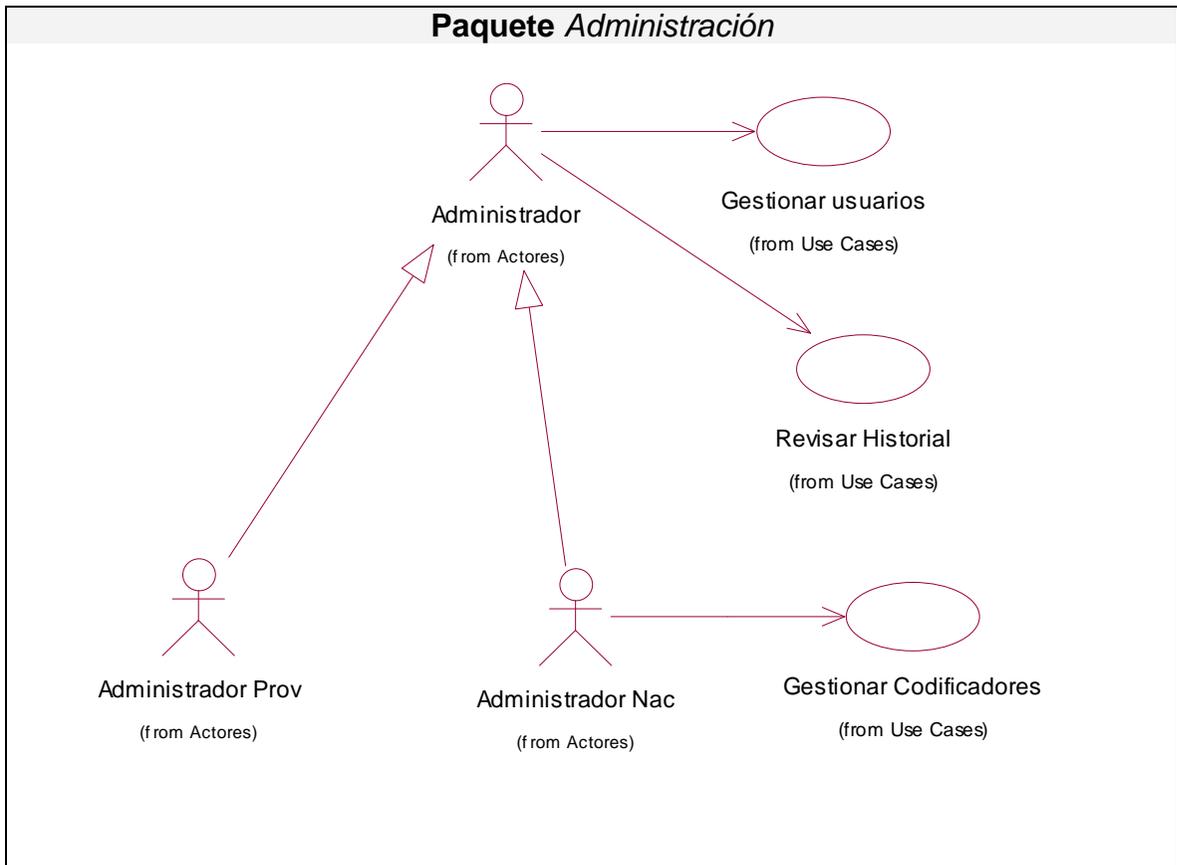


Fig. 3 Paquete "Administración". Diagrama de Casos de Uso.

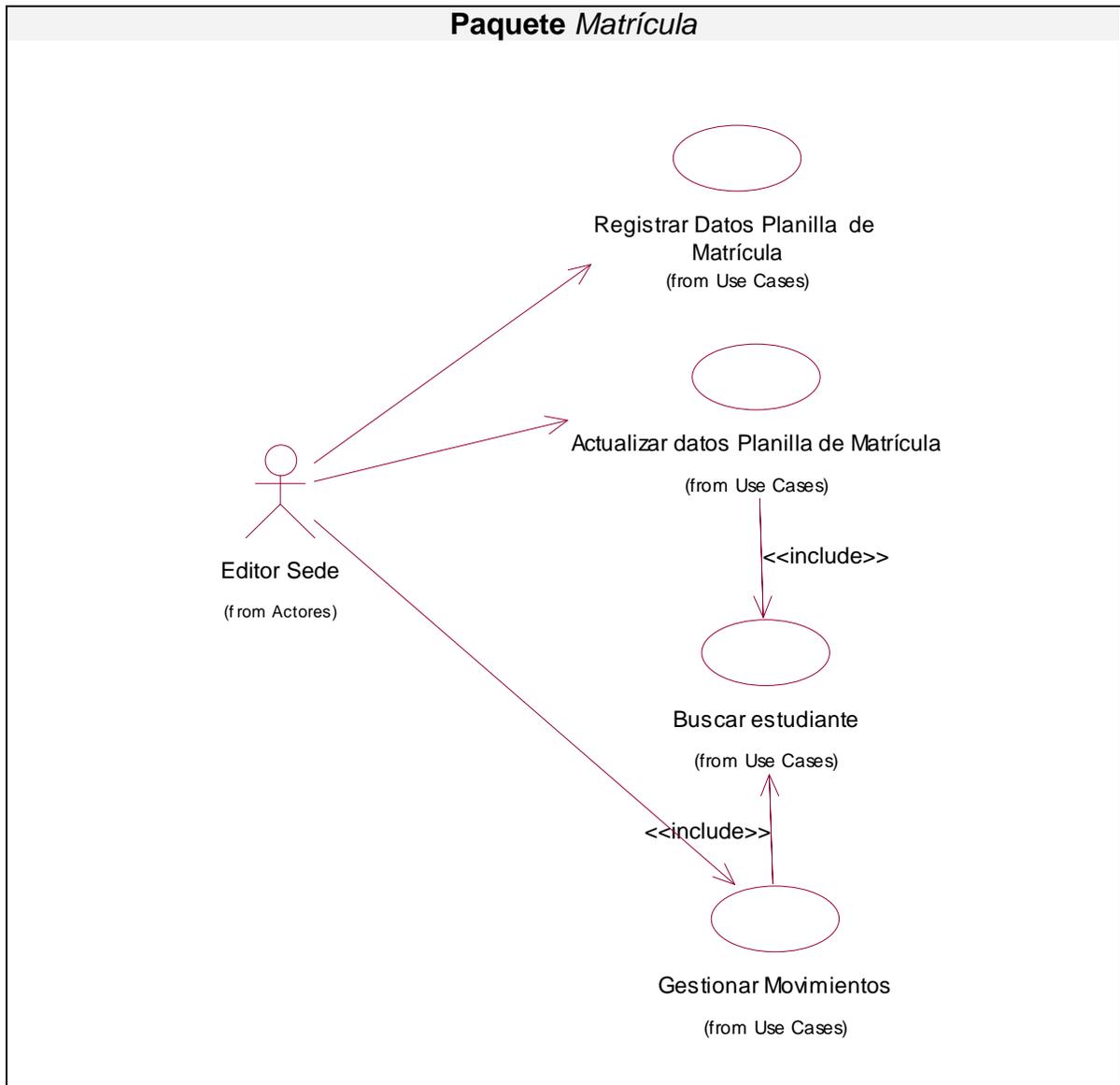


Fig. 4 Paquete "Matrícula". Diagrama de Casos de Uso.

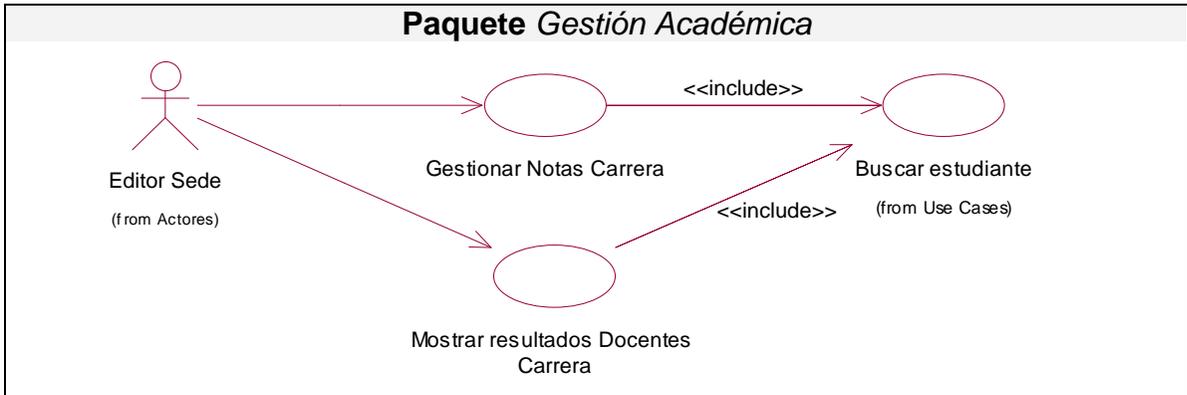


Fig. 5 Paquete "Gestión Académica". Diagrama de Casos de Uso.

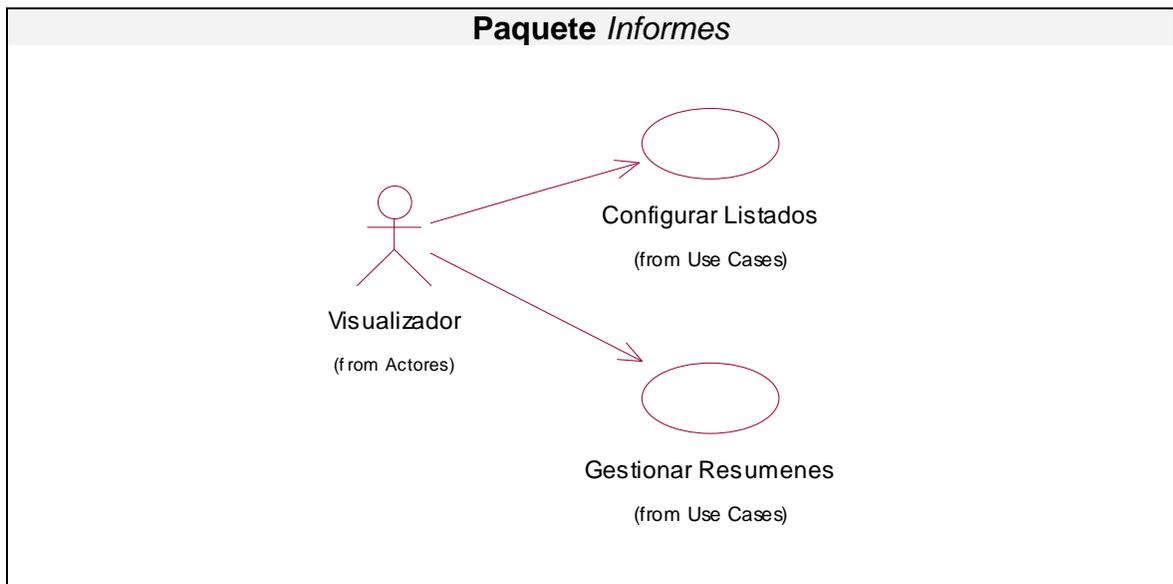


Fig. 6 Paquete "Informes". Diagrama de Casos de Uso.

Listado de casos de uso.

CU - A	Autenticar Usuario.
Actor	Usuario.
Descripción	Este caso de uso se encarga de direccionar al usuario después de autenticado en el sistema, hacia las funcionalidades que le corresponde al mismo como usuario del sistema.
Referencia	R8.

CU – A1	Gestionar Usuarios
Actor	Administrador.
Descripción	Este caso de uso se encarga de todo el negocio relacionado con gestionar usuarios del sistema, así como mostrar listado de usuarios, modificar datos de un usuario registrado, cambiar contraseña, modificar datos personales y otros, eliminar un usuario existente, registrar un nuevo usuario, asignar rol y nivel de acceso al nuevo usuario.
Referencia	R6.

CU – A2	Revisar Historial
Actor	Administrador.
Descripción	Este caso de uso se encarga de obtener un historial de accesos al sistema de todos los usuarios o de uno en particular según el criterio de búsqueda especificado por el administrador, muestra datos como accesos por usuario y fecha de acceso, páginas a las que accedió, fecha, hora e IP de la máquina desde la cual realizó el acceso y hasta el tipo de operación que realizó.
Referencia	R7

CU – A3	Gestionar Codificadores.
Actor	Administrador.
Descripción	Este caso de uso se encarga de gestionar datos de codificadores del sistema

	que son tablas de la base de datos que contienen información codificada para los usuarios tales como provincias, municipios, etc. A través de este se puede registrar, modificar y eliminar datos de un codificador y además listar todos los codificadores existentes.
Referencia	R9

CU – M1	Registrar datos Planilla de Matrícula.
Actor	Editor Sede.
Descripción	Este caso de uso se encarga de registrar los datos del estudiante recogidos en la planilla de matrícula (<i>datos generales, personales, familiares, de la carrera</i>).
Referencia	R1

CU – M2	Actualizar datos de la Planilla de Matrícula.
Actor	Editor Sede.
Descripción	Este caso de uso se encarga de actualizar la información registrada del estudiante en la planilla de matrícula.
Referencia	R2, R13

CU – M3	Gestionar Movimientos.
Actor	Editor Sede.
Descripción	Este caso de uso se encarga de todo el negocio relacionado con los movimientos que realice un estudiante durante su formación: bajas, traslados, incorporaciones, licencias de matrícula, etc.
Referencia	R13, R14

CU – M4	Buscar Estudiante.
Actor	Editor Sede.
Descripción	Este caso de uso se encarga de gestionar la búsqueda de cualquier estudiante o brigada docente según los distintos criterios de búsqueda existentes en el sistema.
Referencia	R3, R4

CU – GA1	Gestionar Notas Carrera
Actor	Editor Sede.
Descripción	Este caso de uso se encarga de gestionar el negocio relacionado con registrar, modificar los resultados docentes de un estudiante o de un grupo docente.
Referencia	R9.

CU – GA2	Mostrar Resultados Docentes
Actor	Editor Sede.
Descripción	Este caso de uso se encarga de visualizar los resultados docentes de un estudiante o de un grupo docente.
Referencia	R10

CU – V1	Confeccionar Listados
Actor	Visualizador.
Descripción	Este caso de uso se encarga de visualizar los listados estadísticos a los diferentes niveles de los usuarios visualizadores. El usuario debe indicar alguno de los criterios de búsqueda que brinda el sistema para conformar el listado.
Referencia	R15

CU – V2	Gestionar Resúmenes.
Actor	Visualizador.
Descripción	Este caso de uso se encarga de gestionar todos los resúmenes que el usuario necesite haciendo cualquier tipo de cruces de variables.
Referencia	R16, R17, R18, R19, R20, R21, R22, R23, R24, R25, R26, R27, R28, R29

Casos de uso por ciclos de desarrollo.

El sistema será desarrollado en ciclos de desarrollo dentro de los cuales se priorizaron los casos de uso de acuerdo a las necesidades del cliente.

Cód	Nombre de caso de uso	Paquete	Justificación de la selección.
CU - A	Autenticar Usuario.	Administración.	Debido a que el sistema estará publicado en la red es de prioridad máxima la seguridad, teniendo en cuenta una valoración adecuada de los riesgos y las consecuencias que pueda traer la pérdida o alteración de información producto a un intruso. Es debido a esto que se le concede una alta importancia a este caso de uso y por lo que se decide desarrollarlo en el primer ciclo de desarrollo.
CU – A1	Gestionar Usuarios	Administración.	Producto a que el negocio que se analiza desarrolla una política de asignación de roles los cuales tienen un nivel de acceso diferente a cada una de las partes o módulos del sistema, constituye una prioridad para la funcionalidad del mismo la gestión de los usuarios que interactuarán con él, ya que esto está contenido en la lógica del negocio del mismo. Producto a la fundamentación anterior este caso de uso se desarrollará en el primer ciclo de desarrollo.

CU – A2	Revisar Historial	Administración.	Debido a la importancia de la información que se manejará en el sistema es importante conocer las operaciones que realizan los usuarios del mismo en cualquier momento por lo cual el presente caso de uso es desarrollado en el primer ciclo de desarrollo .
CU – A3	Gestionar Codificadores.	Administración.	Producto a la gran cantidad de codificadores que contiene el sistema y el constante cambio y actualización de los mismos debido a la dinámica del proceso que implementa el sistema este caso de uso es de vital importancia desarrollarlo en el primer ciclo de desarrollo .
CU – M1	Registrar datos Planilla de Matrícula.	Matrícula.	Producto a que la matrícula es la primera fase de entrada de información al sistema, ya que un estudiante para comenzar a estudiar una carrera en cualquier institución docente lo primero que tiene que hacer es formar parte de la matrícula del mismo. Es de suma importancia que este caso de uso sea desarrollado en el primer ciclo de desarrollo .
CU – M2	Actualizar datos de la Planilla de Matrícula.	Matrícula.	Una vez insertada la información del estudiante en el sistema, es de vital importancia la actualización de los

			datos del mismo por lo que este caso de uso es desarrollado en el primer ciclo de desarrollo.
CU – M3	Gestionar Movimientos.	Matrícula.	Debido a que un estudiante puede ser trasladado o movido ya sea internamente en el centro docente o para otro centro docente, es de vital importancia que este trámite sea automatizado para agilizar este proceso, es por esto que este caso de uso es desarrollado en el primer ciclo de desarrollo.
CU – M4	Buscar Estudiante.	Matrícula.	Producto a que este caso de uso está asociado a los casos de uso Gestionar Movimientos y Actualizar datos de la Planilla de Matrícula es de vital importancia desarrollarlo en el primer ciclo de desarrollo.
CU – GA1	Gestionar Notas Carrera	Gestión Académica.	Este caso de uso debido a que no constituye una prioridad inmediata del usuario se desarrollará en un segundo ciclo de desarrollo.
CU – GA2	Mostrar Resultados Docentes	Gestión Académica.	Este caso de uso debido a que no constituye una prioridad inmediata del usuario se desarrollará en un segundo ciclo de desarrollo.
CU – V1	Confeccionar Listados	Informes.	Este caso de uso debido a que no constituye una prioridad inmediata del usuario se desarrollará en un segundo ciclo de desarrollo.

CU – V2	Gestionar Resúmenes.	Informes.	Este caso de uso debido a que no constituye una prioridad inmediata del usuario se desarrollará en un segundo ciclo de desarrollo.
---------	----------------------	-----------	---

Casos de uso expandidos.

Descripción Caso de Uso “Autenticar Usuario”.

Caso de Uso:	Autenticar Usuario.		
Actores:	Usuario (inicia).		
Propósito:	Gestionar el proceso de autenticarse en el sistema.		
Resumen:	El Caso de Uso se inicia cuando un usuario necesita acceder al sistema para lo cual tiene que autenticarse, este caso de uso finaliza una vez que el usuario logró autenticarse en el sistema correctamente.		
Referencia:	R7		
CU asociados:			
Precondiciones:	El usuario está registrado en el sistema.		
Flujo Normal de Eventos			
Acción del Actor		Respuesta del Sistema	
1 El usuario escribe en el formulario nombre de usuario y contraseña correspondiente para acceder al sistema.		2 El sistema comprueba la validez de los datos y da acceso al usuario.	
Flujos Alternos			
2.1 Si los datos no son correctos el sistema muestra mensajes de error.			
Poscondiciones:	El usuario accede al sistema.		
Prioridad:	Crítico.		

Descripción Caso de Uso “Gestionar Usuarios”.

Caso de Uso:	Gestionar Usuarios.
Actores:	Administrador (inicia), Administrador Nacional, Administrador Provincial
Propósito:	Gestionar los usuarios del sistema.
Resumen:	El Caso de Uso se inicia cuando el administrador necesita crear un usuario del sistema, o ver el listado de usuarios del sistema y editar o eliminar alguno existente. Finaliza el caso de uso una vez realizadas alguna de las acciones anteriores.
Referencia:	R6.
CU asociados:	
Precondiciones:	Administrador Nacional o Administrador Provincial autenticado satisfactoriamente por el sistema.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 El Administrador selecciona una de las siguientes opciones del menú: “Listado de usuarios” “Nuevo usuario”	1.1 El sistema ejecuta una de las siguientes acciones: a) Si desea ver el listado de usuarios del sistema, editar o eliminar alguno existente, ir a la sección “Listado de usuarios”. b) Si desea crear un nuevo usuario del sistema ir a la sección “Nuevo usuario”.
Sección “Listado de usuarios”	
	2.1 El sistema muestra una tabla con los usuarios, esta contiene nombre y apellidos, usuario y el (los) rol(es) que tiene asignado(s). Además la opción de editar o eliminar un

<p>3. Para realizar la edición o eliminación de un usuario, el administrador debe marcar primero el usuario y luego pulsar el botón Editar o Eliminar según la acción que desee realizar.</p>	<p>usuario mediante los botones “Editar” y “Eliminar”.</p> <p>3.1 El sistema ejecuta una de las siguientes acciones:</p> <p>a) Si desea editar un usuario, ir a la sección “Editar usuario”.</p> <p>b) Si desea eliminar un usuario ir a la sección “Eliminar usuario”.</p>
<p>Sección “Editar Usuario”</p>	
<p>5. El Administrador acepta continuar la operación.</p> <p>6. El Administrador modifica los datos y pulsa el botón “Editar”.</p>	<p>4.1 Una vez pulsado el botón Editar, el sistema emite un mensaje de confirmación de la operación.</p> <p>5.1 El sistema muestra la página con el formulario para modificar los datos.</p> <p>6.1 El sistema actualiza los cambios realizados y regresa a la página “Listado de usuarios” donde se muestran los nuevos datos.</p>
<p>Sección “Eliminar Usuario”</p>	
<p>8. El Administrador acepta continuar la operación.</p>	<p>7.1 Una vez pulsado el botón Eliminar, el sistema emite un mensaje de confirmación de la operación.</p> <p>8.1 El sistema elimina el usuario y regresa a la página “Listado de usuarios”, con este eliminado del listado.</p>
<p>Sección “Nuevo usuario”</p>	

<p>9. El Administrador debe llenar el formulario sin dejar campos vacíos, debe escribir usuario, nombre y apellidos de este, contraseña y confirmación de la misma, rol(es) que desempeñará.</p> <p>Los administradores nacionales podrán crear usuarios con las siguientes características (los usuarios pueden tener varios roles según la tarea que realicen, los roles pueden ser: administrador nacional, administrador provincial, visualizador nacional, visualizador provincial, editor de sede; en caso de que sea este último se selecciona provincia, municipio, facultad o centro al que pertenecerá).</p> <p>Los administradores provinciales podrán crear usuarios con las siguientes características (los usuarios pueden tener varios roles según la tarea que realicen, los roles pueden ser: administrador provincial, visualizador provincial, editor de sede, en caso de que sea este último se selecciona municipio, facultad o centro al que pertenecerá).</p>	<p>9.1 El sistema crea el nuevo usuario asignándole el rol y los niveles de acceso que tendrá. Muestra la página “Listado de usuarios” con el nuevo usuario incluido.</p>
Flujos Alternos	
<p>9.1 Si el usuario que se va a insertar ya existe, el sistema envía un mensaje con el error ocurrido.</p>	
Poscondiciones:	Se crea un nuevo usuario o se modifican datos de un usuario existente.
Prioridad:	Crítico.

Descripción Caso de Uso “Revisar Historial”.

Caso de Uso:	Revisar Historial.	
Actores:	Administrador Nacional (inicia)	
Propósito:	Tener registrado una traza de la navegación y operaciones de los usuarios del sistema.	
Resumen:	El Caso de Uso se inicia cuando el administrador desea revisar las operaciones y la navegación de los usuarios que han accedido al sistema,. Se podrá obtener las páginas a las que ha accedido, la fecha, hora y el IP de la máquina desde la cual accedió y además la operación que realizó. Finaliza el caso de uso una vez obtenida la información de los accesos.	
Referencia:	R7.	
CU asociados:		
Precondiciones:	Administrador nacional o Administrador Provincial autenticado satisfactoriamente por el sistema.	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. El Administrador selecciona la opción “Historial” del menú principal de trabajo.	1.1 El sistema muestra la página Historial para realizar la búsqueda de accesos.
	2. El Administrador escribe el usuario y selecciona la fecha de acceso, luego pulsa el botón “Buscar”.	2.1 El sistema muestra una tabla con paginación que contiene el usuario, todos los accesos de este en la fecha seleccionada, la hora y el IP de la máquina desde donde accedió.
Flujos Alternos		
Poscondiciones:	Se obtuvo historial de accesos y operaciones del usuario.	
Prioridad:	Crítico	

Descripción Caso de Uso “Gestionar Codificadores”.

Caso de Uso:	Gestionar Codificadores.	
Actores:	Administrador (inicia)	
Propósito:	Actualizar, modificar y eliminar datos de los codificadores del sistema.	
Resumen:	El Caso de Uso se inicia cuando el administrador desea actualizar (adicionar, editar, eliminar) los datos de algún codificador. Finaliza el caso de uso una vez realizada la actualización.	
Referencia:	R9.	
CU asociados:		
Precondiciones:	Administrador nacional autenticado satisfactoriamente por el sistema.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Administrador selecciona la opción “Nomencladores” del menú principal de trabajo.	1.1 El sistema muestra la página Nomencladores que contiene un listado de los codificadores del sistema.	
2. El Administrador indica el codificador que desee actualizar y pulsa el botón “Editar”	2.1 El sistema muestra la página con los datos del codificador seleccionado (campos para introducir datos del codificador y listado de todos los datos de este) y las opciones de adicionar datos, editar o eliminar datos existentes, a través de los botones “Adicionar”, “Editar” y “Eliminar” respectivamente. <ul style="list-style-type: none"> a) Si desea adicionar datos al codificador ir a sección “Adicionar”. b) Si desea editar (modificar) datos al codificador ir a sección “Editar”. c) Si desea eliminar datos al codificador ir a sección “Eliminar”. 	

Sección “Adicionar”	
3. El administrador llena los campos con los nuevos datos del codificador y pulsa el botón “Adicionar”.	3.1 El sistema inserta los nuevos datos e inmediatamente los muestra en el listado de datos del codificador que aparece en la página.
Sección “Editar”	
4. El administrador debe marcar la fila correspondiente a los datos que desee modificar.	4.1 El sistema carga estos datos en los campos de edición.
5. El administrador realiza los cambios pertinentes y pulsa el botón “Editar”.	5.1 El sistema muestra un mensaje de confirmación de la operación a realizar.
6. El administrador acepta la confirmación	6.1 El sistema actualiza los datos e inmediatamente los muestra en el listado de datos del codificador que aparece en la página.
Sección “Eliminar”	
7. El administrador debe marcar la fila correspondiente a los datos que desee eliminar y pulsar el botón “Eliminar”.	7.1 El sistema muestra un mensaje de confirmación de la operación a realizar.
8. El administrador acepta la confirmación.	8.1 El sistema elimina los datos e inmediatamente se dejarán de mostrar en el listado de datos del codificador.
Flujos Alternos	
2. Si no seleccionó el codificador, al pulsar el botón el sistema emite un mensaje para que lo seleccione.	

Poscondiciones:	Se actualizan los datos del codificador.
Prioridad:	Crítico

Descripción Caso de Uso “Registrar Datos Planilla Matrícula”.

Caso de Uso:	Registrar Datos Planilla Matrícula.
Actores:	Editor Sede (inicia)
Propósito:	Registrar los datos de la planilla de matrícula del estudiante en el sistema.
Resumen:	El Caso de Uso se inicia cuando el editor necesita insertar un nuevo estudiante al sistema, se registran todos los datos del estudiante: datos generales, personales, familiares y de la carrera. Finaliza el caso de uso una vez registrados estos datos, el estudiante pasa a formar parte de la matrícula del centro.
Referencia:	R1
CU asociados:	
Precondiciones:	Editor de Sede autenticado satisfactoriamente por el sistema.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Editor selecciona la opción “Nuevo estudiante” y en dependencia del tipo de estudiante que vaya a matricular escogerá la opción de cubano o extranjero.	<p>1.1 El sistema muestra la página de Datos Iniciales para comenzar la captación de datos que se rige por una planilla oficial del Ministerio de Educación Superior (MES).</p> <p>a) Si desea matricular un estudiante cubano ir a la sección “Matricular estudiante cubano”.</p> <p>b) Si desea matricular un estudiante extranjero ir</p>

	a la sección "Matricular estudiante extranjero".
Sección "Matricular estudiante cubano"	
2. El Editor llena los campos. Pulsa el botón "Siguiete" para continuar el registro del estudiante.	2.1 El sistema muestra la siguiente página "Datos Personales" para llenar estos datos del estudiante.
3. El Editor llena los campos. Pulsa el botón "Siguiete" para continuar	3.1 El sistema muestra la siguiente página de datos personales a llenar del estudiante.
4. El Editor llena los campos. Pulsa el botón "Siguiete" para continuar.	4.1 El sistema muestra la siguiente página "Datos Familiares" para llenar estos datos del estudiante.
5. El Editor llena los campos. Pulsa el botón "Siguiete". para continuar	5.1 El sistema muestra la página "Datos Generales del Estudiante" para que se comience a insertar los datos.
6. El Editor llena los campos. Pulsa el botón "Guardar" para terminar la operación.	6.1 El sistema inserta el estudiante; muestra una página con los datos más generales del mismo.
7. El Editor pulsa el botón "Aceptar". para continuar	7.1 El sistema muestra la pagina "Datos Iniciales" para comenzar a matricular otro estudiante.
Sección "Matricular estudiante extranjero"	
2. El Editor llena los campos. Pulsa el botón "Siguiete" para continuar el registro del estudiante.	2.1 El sistema muestra la siguiente página "Datos Personales" para llenar estos datos del estudiante.
3. El Editor llena los campos. Pulsa el botón "Siguiete" para continuar	3.1 El sistema muestra la siguiente página de datos personales a llenar del estudiante.
4. El Editor llena los campos. Pulsa el botón "Siguiete" para continuar.	4.1 El sistema muestra la siguiente página "Datos Familiares" para llenar estos datos del estudiante.
5. El Editor llena los campos. Pulsa el botón "Siguiete" para continuar.	5.1 El sistema muestra la siguiente página "Datos Referentes a la Carrera" para llenar los datos del estudiante correspondientes a su situación docente.
6. El Editor llena los campos. Pulsa el botón	6.1 El sistema inserta el estudiante; muestra una

“Guardar” para terminar la operación.	página con los datos más generales del mismo.
7. El Editor pulsa el botón “Aceptar”. para continuar	7.1 El sistema muestra la pagina “Datos Iniciales” para comenzar a matricular otro estudiante.
Flujos Alternos	
1.1, 2.1, 3.1, 4.1, 5.1, 6.1 Si desea cancelar la operación pulsa el botón “Cancelar” y el sistema saldrá de la opción “Nuevo estudiante”.	
2.1, 3.1, 4.1, 5.1, 6.1 Si cometió algún error al llenar los campos de la página anterior puede regresar a esta pulsando el botón “Anterior”	
Poscondiciones:	Se registra un nuevo estudiante.
Prioridad:	Crítico

Descripción Caso de Uso “Actualizar Datos Planilla Matrícula”.

Caso de Uso:	Actualizar Datos Planilla Matrícula.
Actores:	Editor de Sede (inicia)
Propósito:	Actualizar o modificar los datos que se tienen registrados del estudiante en el sistema en caso de haber algún error.
Resumen:	El Caso de Uso se inicia cuando el editor necesita modificar algún dato de un estudiante registrado, accederá a las páginas de datos generales, personales, familiares y de la carrera según los datos que desee actualizar. Finaliza el caso de uso una vez actualizados los datos del estudiante.
Referencia:	R2
CU asociados:	Buscar Estudiante
Precondiciones:	Editor de Sede autenticado satisfactoriamente por el sistema. El estudiante debe estar registrado en el sistema.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 El editor selecciona según el tipo de	1.2 El sistema ejecuta una de las siguientes

<p>estudiante que sea si es extranjero o cubano la opción del menú correspondiente para cada uno de los casos, según los datos que desee actualizar del estudiante, una de las siguientes opciones del menú:</p> <p>Opciones para estudiantes extranjeros: “Datos Personales” “Datos Familiares” “Datos Referentes a la Carrera”</p> <p>Opciones para estudiantes cubanos: “Datos Personales” “Datos Familiares” “ Datos Generales del Estudiante ”</p>	<p>acciones en dependencia del tipo de estudiante que sea para el caso de los estudiantes extranjeros:</p> <p>a) Si desea actualizar alguno de los datos personales ir a la sección “Datos Personales”. b) Si desea actualizar alguno de los datos de sus familiares ir a la sección “Datos Familiares”. c) Si desea actualizar alguno de los datos de la carrera y procedencia escolar ir a la sección “Datos Referentes a la Carrera”.</p> <p>1.3 Para el caso de los estudiantes cubanos:</p> <p>a) Si desea actualizar alguno de los datos personales ir a la sección “Datos personales”. b) Si desea actualizar alguno de los datos familiares ir a la sección “Datos familiares”. c) Si desea actualizar alguno de los datos generales del estudiante ir a la sección “Datos Generales del Estudiante”.</p> <p>Para realizar la actualización se debe buscar primero al estudiante, el sistema muestra la página de búsqueda, se invoca al caso de uso Buscar Estudiante.</p>
<p>Sección “Datos Personales”</p>	
<p>2. El Editor modifica los campos. Pulsa el botón “Guardar” para realizar la actualización o el botón “Siguiente” si desea además modificar datos de la otra página de</p>	<p>2.1 El sistema actualiza los datos modificados y va a la página para realizar búsqueda.</p>

datos personales; en esta entonces pulsaría el botón “Guardar” para completar la operación o el botón “Anterior” si desea regresar a la primera página de datos personales.	
Sección “Datos Familiares”	
3. El Editor modifica los campos. Pulsa el botón “Guardar” para completar la actualización.	3.1 El sistema actualiza los datos modificados y va a la página para realizar búsqueda.
Sección “Datos Generales del Estudiante”	
4. El Editor modifica los campos. Pulsa el botón “Guardar” para completar la actualización.	4.1 El sistema actualiza los datos modificados y va a la página para realizar búsqueda.
Sección “Datos Referentes a la Carrera”	
5. El Editor modifica los campos. Pulsa el botón “Guardar” para completar la actualización.	5.1 El sistema actualiza los datos modificados y va a la página para realizar búsqueda.
Flujos Alternos	
2.1, 3.1, 4.1, 5.1 Si desea cancelar la actualización pulsa el botón “Cancelar”, el sistema no realizará los cambios. Se mostrará la página de búsqueda.	
2.1, 4.1, 5.1 Si cometió algún error al llenar los campos de la página anterior puede ir a esta pulsando el botón “Anterior”.	
Poscondiciones:	Se actualizan los datos del estudiante
Prioridad:	Alta.

Descripción Caso de Uso “Buscar Estudiante”

Caso de Uso:	Buscar Estudiante	
Actores:	Editor de Sede (inicia).	
Propósito:	Buscar por diferentes criterios de búsqueda a un estudiante determinado.	
Resumen:	El Caso de Uso se inicia cuando se desea actualizar la información de algún estudiante o realizar algún trámite de movimiento, para ello primeramente se debe buscar ese estudiante. Finaliza el caso de uso una vez que se encuentra el estudiante.	
Referencia:	R3.3, R3.4	
CU asociados:	Actualizar Datos Planilla Matrícula, Gestionar Movimientos	
Precondiciones:	Editor de sede autenticado satisfactoriamente por el sistema. El estudiante debe estar registrado en el sistema.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
<p>2. El editor de sede debe llenar alguno de los campos para buscar al estudiante: nombre, apellidos, sexo, o en caso de que conozca la brigada docente del mismo solo bastará con esta opción. Luego pulsa el botón “Buscar”.</p> <p>3.El editor de sede selecciona el estudiante que desea haciendo clic en el hipervínculo del Nombre y Apellidos.</p>	<p>1.1 El sistema muestra la página de búsqueda.</p> <p>2.1El sistema muestra un listado de los estudiantes que cumplan con los criterios de búsqueda, se muestra el Código del estudiante, Nombre y Apellidos y País.</p> <p>3.1 El sistema muestra la página correspondiente que invocó la búsqueda.</p>	

Flujos Alternos	
2.1. Si no se encuentran estudiantes con los criterios de búsqueda deseados, el sistema muestra un mensaje indicando que no existe ningún estudiante con estos datos.	
Poscondiciones:	Se busca el estudiante.
Prioridad:	Crítico

Descripción Caso de Uso “Gestionar Movimientos”

Caso de Uso:	Gestionar Movimientos.
Actores:	Editor sede (inicia).
Propósito:	Gestionar el proceso relacionado con las bajas y traslados de los estudiantes.
Resumen:	El caso de uso se inicia cuando el Editor Sede necesita realizar un trámite de movimiento a un estudiante ya sea un traslado, una baja, una licencia de matrícula o también la incorporación de un estudiante para lo cual primero debe buscar al estudiante. Finaliza el caso de uso una vez realizada alguna de las acciones anteriores.
Referencia:	R13, R14
CU asociados:	Buscar Estudiante
Precondiciones:	Editor Sede autenticado satisfactoriamente por el sistema. El estudiante debe estar registrado en el sistema.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 El Editor Sede selecciona la opción “Movimientos” del menú principal de trabajo.	Para realizar un movimiento a un estudiante se debe buscar primero al estudiante, el sistema muestra la página de búsqueda, se invoca al caso de uso Buscar Estudiante.

<p>2 El Editor Sede selecciona una de las siguientes opciones del tipo de trámite: “Traslado” “Baja” “Licencia de Matrícula”</p>	<p>1.1 El sistema muestra la página de movimientos con los datos generales del estudiante y un listbox para seleccionar el tipo de trámite.</p> <p>2.1 El sistema ejecuta una de las siguientes acciones:</p> <p>a) Si seleccionó “Traslado” ir a la sección “Traslado”.</p> <p>b) Si seleccionó “Baja ” ir a la sección “Baja”.</p> <p>c) Si seleccionó “Licencia de Matrícula” ir a la sección “Licencia de Matrícula”.</p> <p>d) Si desea realizar la operación de incorporación de un estudiante ir a la sección “Incorporación”</p>
<p>Sección “Traslado”</p>	
<p>3. El Editor Sede debe llenar todos los campos y pulsar el botón “Aceptar”.</p>	<p>2.1 El sistema muestra los campos a llenar:: el tipo de traslado, el municipio, la provincia, el centro, la facultad y las causas por las cuales se realiza este movimiento.</p> <p>3.1 El sistema ejecuta la acción y regresa a la página de búsqueda de un estudiante.</p>
<p>Sección “Baja”</p>	
	<p>4.1 El sistema muestra los tipos de baja a seleccionar (<i>temporal o definitiva</i>) y las causas de la misma.</p>

<p>5. El Editor Sede selecciona el tipo de baja y la causa y pulsa el botón "Aceptar".</p>	<p>5.1 El sistema realiza la baja del estudiante y regresa a la página de búsqueda de un estudiante.</p>
<p>Sección "Licencia de Matrícula"</p>	
<p>7. El Editor Sede llena todos los campos y pulsa el botón "Aceptar".</p>	<p>6.1 El sistema muestra los datos a llenar para este trámite: tiempo solicitado y tiempo aprobado de la licencia, y causa de la misma</p> <p>7.1 El sistema ejecuta la acción y regresa a la página de búsqueda de un estudiante.</p>
<p>Sección "Incorporación"</p>	
<p>9. El Editor Sede selecciona al estudiante para hacerle el trámite</p> <p>10. El Editor Sede selecciona la causa, pulsa el botón "Aceptar".</p> <p>11 El Editor Sede llena todos los campos</p>	<p>8 El sistema muestra el listado de estudiantes que esperan trámite de incorporación de otro centro o brigada.</p> <p>9.1 El sistema muestra los datos generales del estudiante y las posibles causas para la incorporación.</p> <p>10.1 El sistema muestra la siguiente página de datos a completar para este trámite: nivel de aprobación, fecha de solicitud, fecha de aprobación, fecha de incorporación.</p> <p>11.1 El sistema ejecuta la incorporación del estudiante.</p>
<p>Flujos Alternos</p>	
<p>Poscondiciones:</p>	<p>Se realiza el trámite de movimiento al estudiante.</p>
<p>Prioridad:</p>	<p>Alta</p>

ANEXO # 3: MODELO DE DISEÑO.

Paquete Administración.

Teniendo en cuenta que el presente trabajo será publicado en la biblioteca de la Universidad para el uso colectivo de todos los interesados, la información referente al módulo de administración del sistema que puede ser sensible para su funcionamiento no se publicará en el documento por motivos de seguridad. El interesado en acceder a esta información debe contactar con la dirección del proyecto en la facultad.

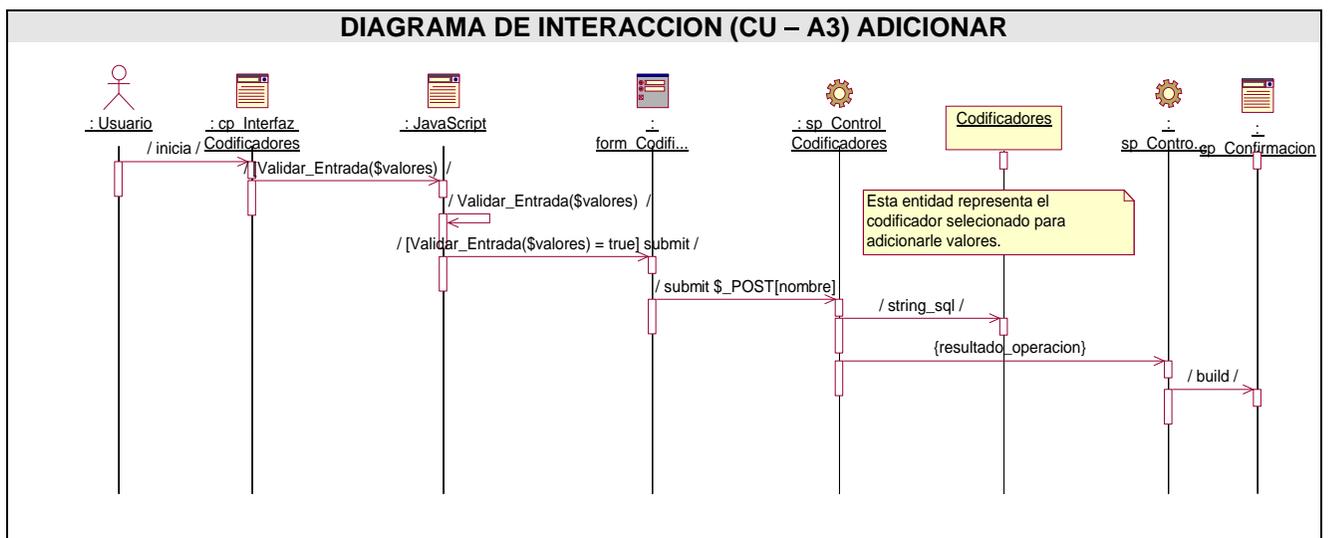


Fig. 1 Diagrama de interacción (CU- 3)Adicionar datos al codificador.

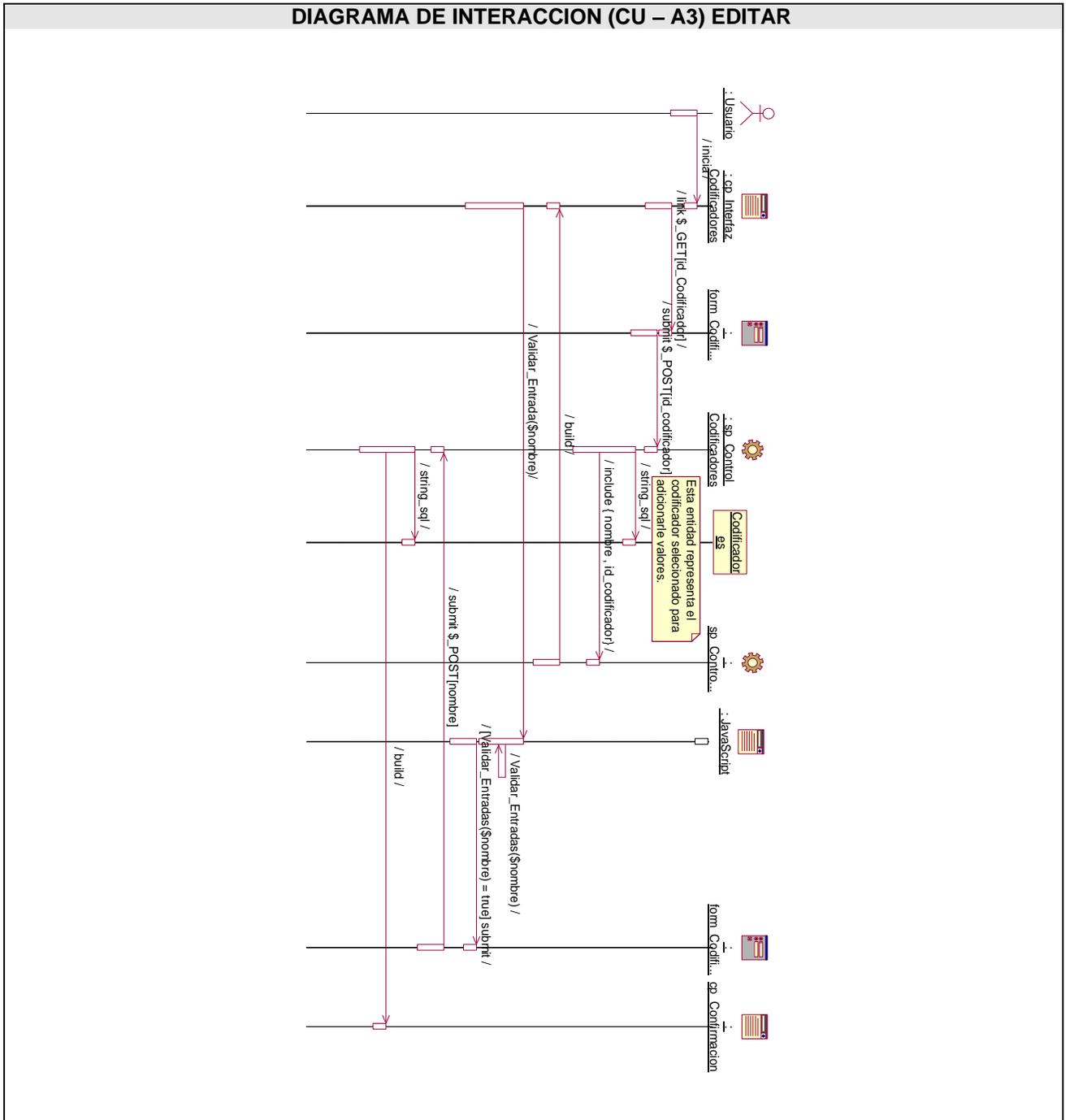
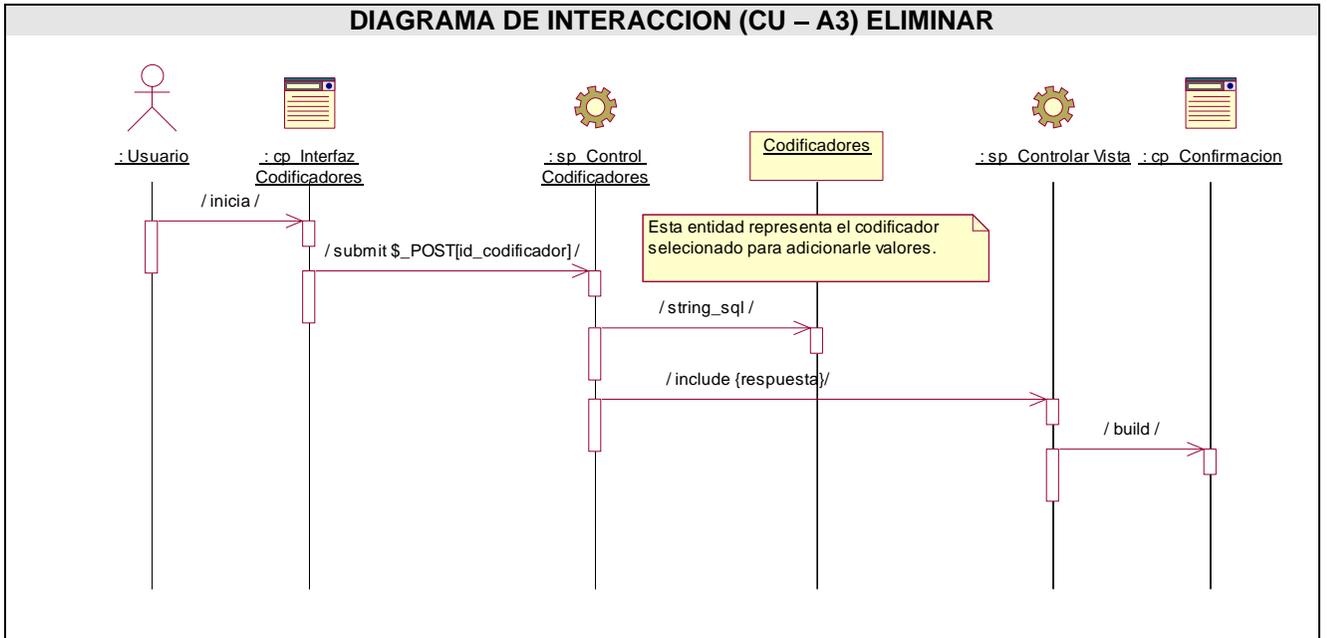


Fig. 2 Diagrama de interacción (CU- 3) Editar datos en el codificador.



Paquete Matrícula.

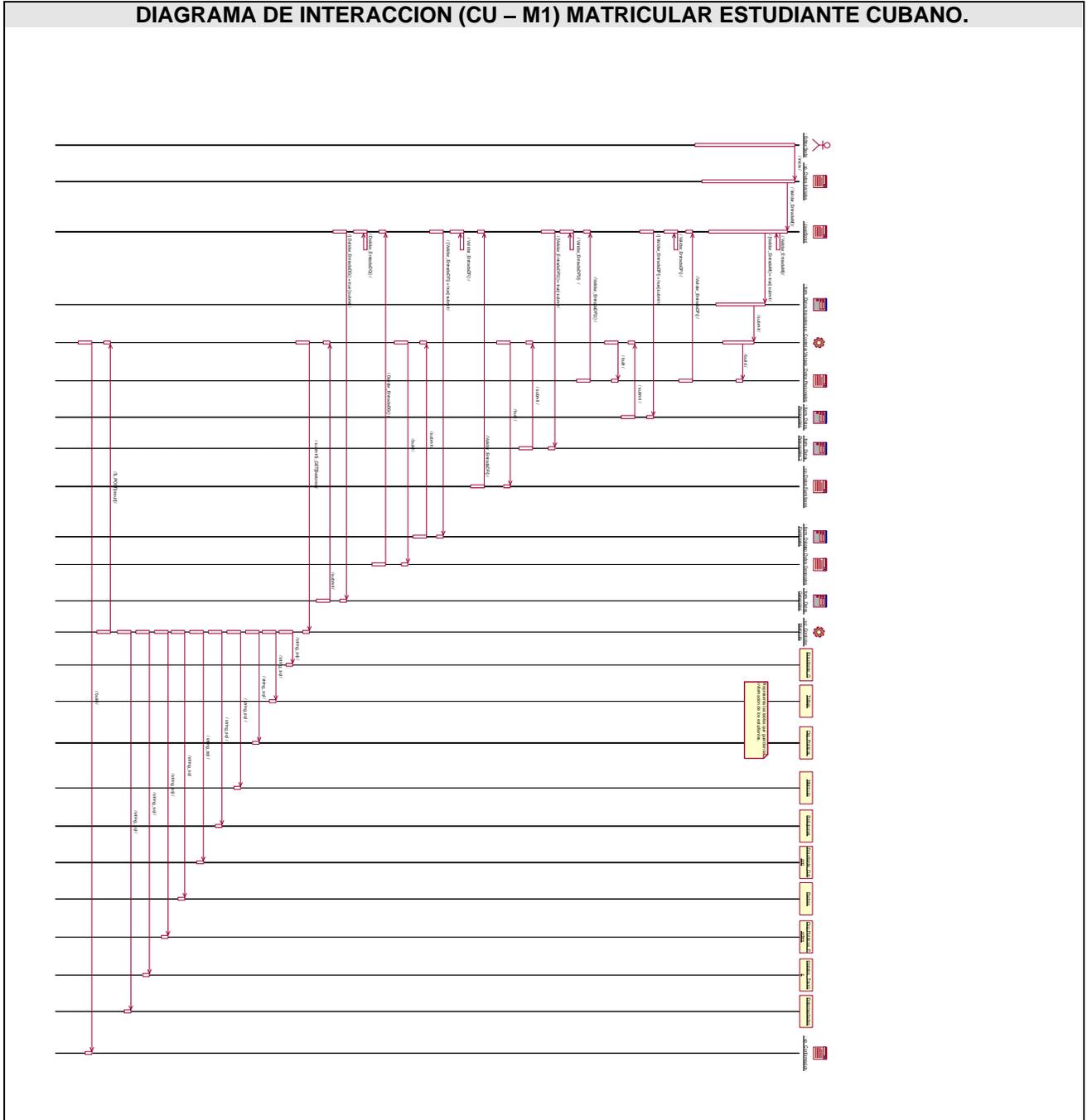


Fig. 4 Diagrama de interacción (CU- M1) Matricular estudiante cubano

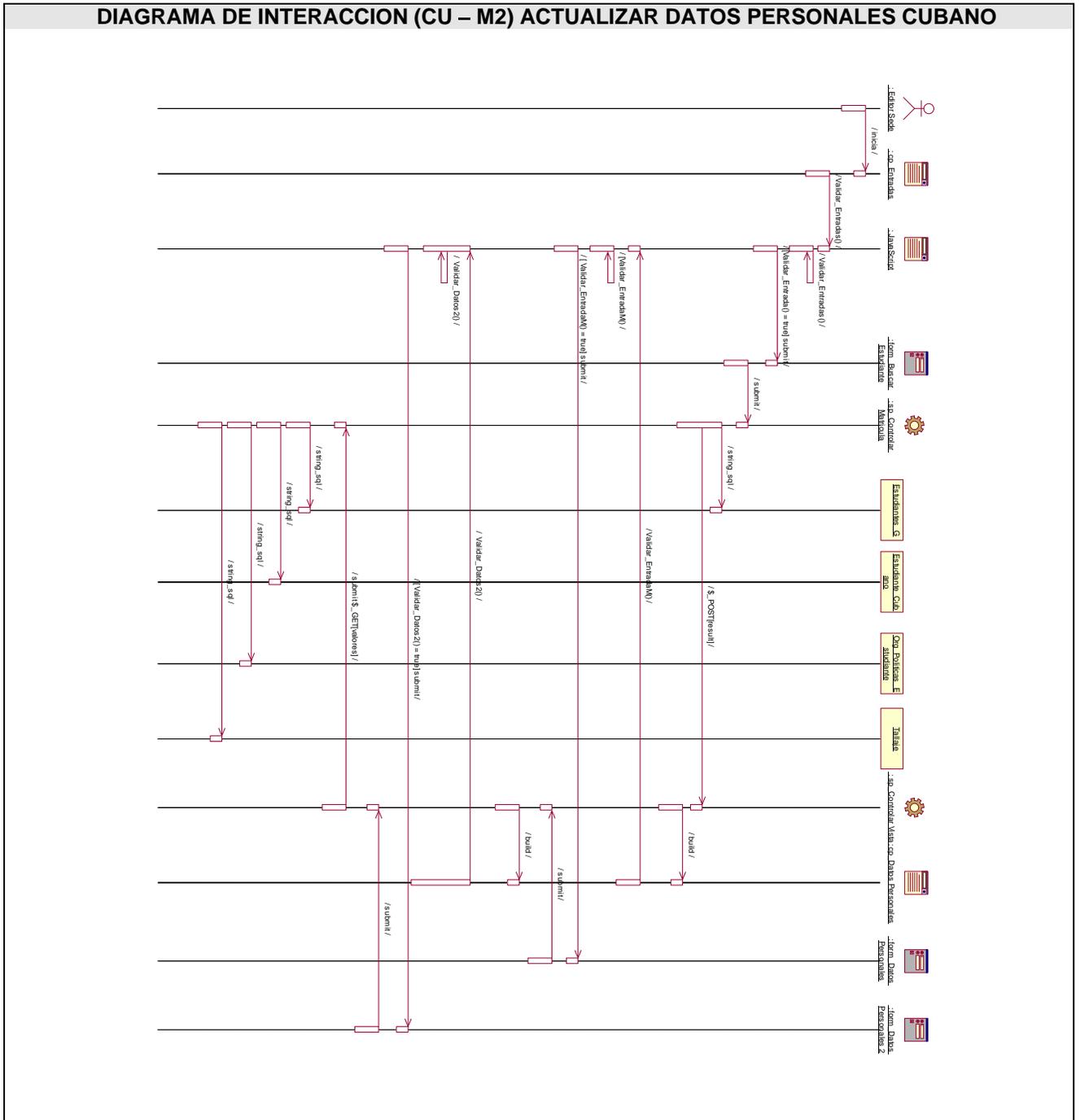


Fig. 7 Diagrama de interacción (CU- M2) Actualizar datos personales estudiante cubano.

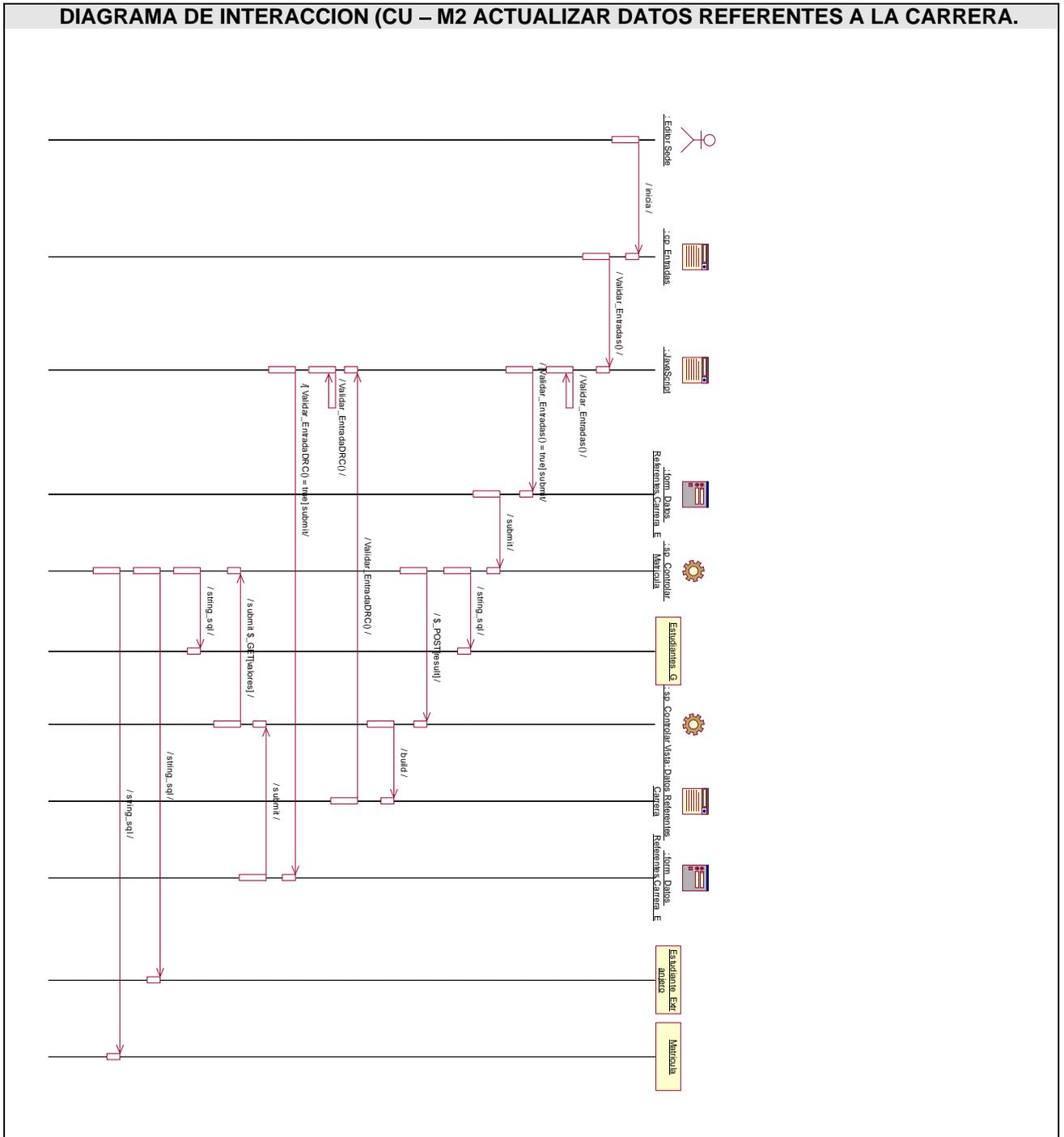


Fig. 10 Diagrama de interacción (CU- M2) Actualizar datos referentes a la carrera.

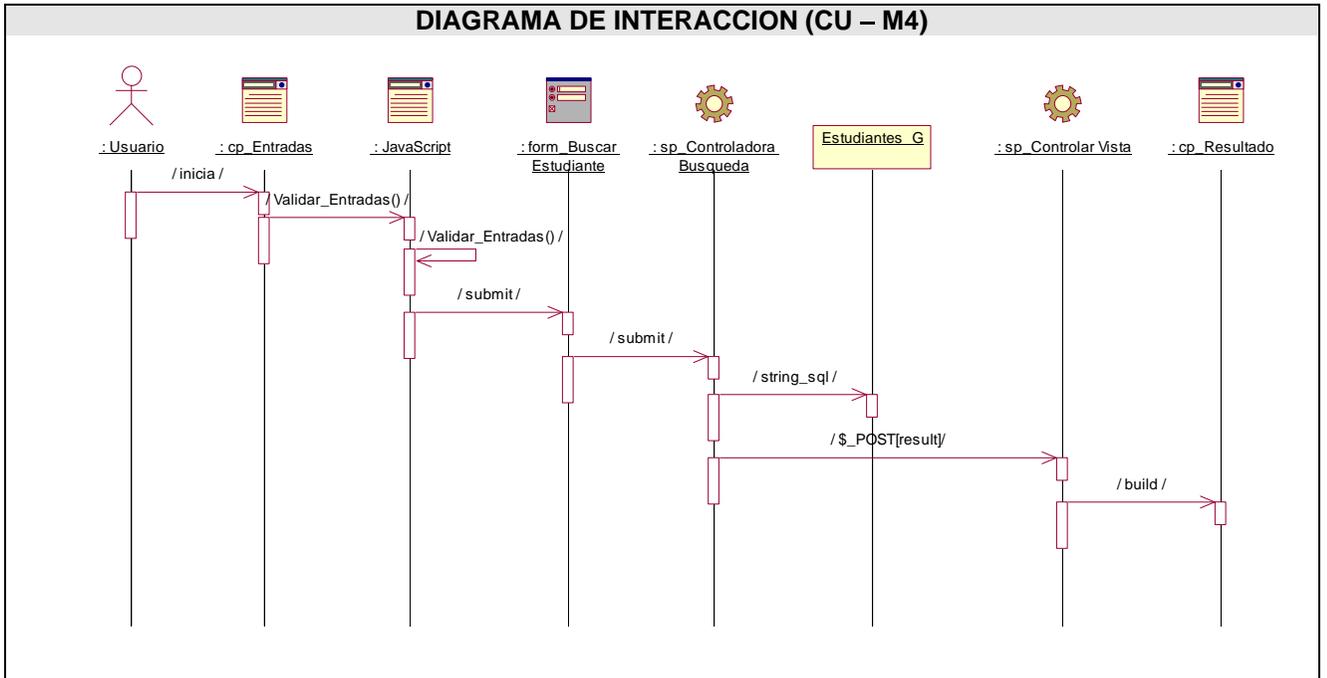


Fig. 12 Diagrama de interacción (CU- M4) Buscar estudiante.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.